



Actas

JENUI 2011

XVII Jornadas de Enseñanza Universitaria de la Informática

Escuela Técnica Superior de Ingeniería Informática
Sevilla, 5 al 8 de julio de 2011



Actas de las XVII Jornadas de Enseñanza Universitaria de la Informática
Sevilla, Julio 2011
jenui2011.us.es

Editores de las Actas:

Antonia Chávez
Francisco Gómez
José Ra. Portillo
Agustín Riscos

ISBN: 978-84-694-5156-4

Depósito Legal:

Prólogo del comité de programa

En 2011, las Jornadas de Enseñanza Universitaria de la Informática (JENUI) alcanzan su decimoséptima edición. Gracias al impulso de AENUI, la Asociación de Enseñantes Universitarios de la Informática, estas jornadas se han convertido en la cita anual a la que muchos profesores de informática preocupados por su tarea docente en la Universidad procuran no faltar. Invitamos a participar en futuras ediciones de estas Jornadas a todos aquellos interesados en aprender y compartir sus experiencias docentes y sus resultados de investigación para la mejora del aprendizaje de nuestros estudiantes.

Este volumen recoge las contribuciones aceptadas para su presentación en la XVII edición de JENUI, celebrada en la Escuela Técnica Superior de Ingeniería Informática de la Universidad de Sevilla del 6 al 8 de julio de 2011.

Algunas de las contribuciones a las Jornadas tratan sobre la docencia de las diferentes materias de informática presentes en estudios universitarios, como por ejemplo la programación, las bases de datos, la seguridad, los fundamentos teóricos de la informática, la arquitectura de computadores o los sistemas operativos. Otras abordan aspectos más generales del proceso de enseñanza-aprendizaje de la informática en la universidad, como la evaluación del alumnado, la calidad y evaluación de la docencia, el desarrollo de competencias profesionales o la innovación pedagógica.

Estamos inmersos en la implantación de las titulaciones de grado adaptadas al EEES. Durante las Jornadas, éste será sin duda uno de los temas estrella.

Sesiones plenarias

La conferencia invitada que abrirá las Jornadas, titulada “Hacia la alineación de Competencias, Metodologías de enseñanza-aprendizaje y Evaluación en los Grados TIC”, será dictada por Davinia Hernández Leo, profesora del departamento de Tecnologías de la Información y las Comunicaciones de la Universitat Pompeu Fabra, directora de la Unidad de Apoyo a la Calidad y a la Innovación Docente (USQUID) de la Escuela Superior Politécnica (ESUP), y coordinadora de la línea de investigación de Technology-Enhanced Learning del Grupo de Tecnologías Interactivas (GTI). En esta charla se presentarán algunos de los resultados obtenidos en el proyecto de Estudios y Análisis AlineaME, en el que el objetivo ha sido trabajar sobre uno de los principales retos que supone la implantación del Espacio Europeo de Educación Superior, quizá el más importante: cómo podemos alinear las metodologías docentes y los sistemas de evaluación con las competencias a desarrollar en las asignaturas. Este proyecto fue coordinado por la Unidad de Soporte a la Calidad y la Innovación Docente de la Escuela Superior Politécnica de la Universidad Pompeu Fabra, e involucraba a docentes y pedagogos de siete universidades españolas. El ámbito de estudio se centraba en asignaturas de Grados relacionados con las Tecnologías de la Información y las Comunicaciones. El trabajo interdisciplinar, entre pedagogos y docentes en Grados TIC, ha dado lugar a la identificación de recomendaciones que permiten avanzar hacia la alineación buscada. Estas recomendaciones se han puesto en marcha en las

diversas asignaturas de los docentes involucrados en el proyecto. Los resultados obtenidos muestran que los profesores se encuentran más satisfechos con los resultados derivados del trabajo por competencias y que los alumnos entienden mejor la razón de ser de las metodologías y actividades de evaluación planteadas en las asignaturas. Asimismo, se observa que las acciones realizadas facilitan la auto-regulación a lo largo de la asignatura y un aumento en el grado de implicación de los estudiantes en las diferentes tareas propuestas. El informe completo del proyecto está públicamente disponible en la dirección <http://www.usquidesup.upf.edu/es/alineame-es>.

Las jornadas también cuentan con una mesa redonda que bajo el título de “Implantación y Calidad de los Nuevos Planes de Estudio de Grado en Ingeniería informática” contará con la presencia de tres ponentes invitados: Saturnino Vicente Díaz, Subdirector de Calidad y Planificación Estratégica de la E.T.S. de Ingeniería Informática; Inmaculada Medina Bulo, coordinadora de los nuevos programas de grado en Ingeniería Informática en la Universidad de Cádiz; así como Javier Segovia, decano de la Facultad de Informática de la Universidad Politécnica de Madrid. Sin duda alguna, se trata de tres ponentes con una gran implicación en la puesta en marcha de los nuevos estudios de Grado en Ingeniería Informática y con una gran experiencia en temas relacionados con la calidad. Esperamos que estos temas atraigan la atención de los asistentes a las jornadas y den lugar a un debate fructífero.

Proceso de revisión

Las contribuciones recibidas han sido examinadas de forma rigurosa y anónima por al menos tres revisores, cuyos comentarios y valoraciones han permitido la selección de los trabajos presentados en las jornadas y la detección de aspectos susceptibles de mejora. Cuando se ha producido una disparidad importante en las valoraciones iniciales, se han solicitado revisiones adicionales para fundamentar mejor la decisión del Comité de Programa.

Se recibieron un total de 102 trabajos: 77 propuestas de ponencia, 9 recursos docentes y 16 pósteres. De los 102 trabajos recibidos han sido aceptados 66, lo que supone una tasa global de aceptación del 64,70%.

Este proceso de revisión y selección no habría sido posible sin la participación casi un centenar de revisores procedentes de la mayoría de universidades españolas. Desde aquí queremos expresar nuestro más sincero agradecimiento a todos ellos.

Somos conscientes de que hemos podido, en algún caso, no tomar la decisión más acertada al rechazar un trabajo, pero estamos seguros de la calidad de las contribuciones recogidas en este libro. Valoramos especialmente el esfuerzo que todos los autores hacen al redactar los trabajos sometidos a evaluación, pues sus contribuciones suponen el núcleo de las Jornadas.

Publicación

Las actas de JENUI 2011 no estarán disponibles en el formato tradicional de libro en papel por primera vez desde que se celebra JENUI. Se repartirán en un pendrive a los asistentes a las jornadas y podrán consultarse permanentemente en la web de JENUI 2011 (<http://jenui2011.us.es>) y en la web de AENUI (<http://www.aenui.net>).

La revista Novática, renovando su compromiso de los últimos años, publicará en su columna sobre enseñanza universitaria de la informática algunos de los trabajos presentados en JENUI.

El acuerdo de colaboración alcanzado con el Capítulo Español de la Sociedad de Educación del IEEE incluye la publicación de algunos trabajos en la revista IEEE-RITA y en TICAI, un libro que anualmente recoge las aportaciones más significativas realizadas en los congresos más importantes de habla española y portuguesa en el ámbito de la Sociedad de la Educación del IEEE.

Además, los editores de ReVisión, la revista de AENUI, invitarán a los autores de alguno de los trabajos a publicar sus contribuciones en la revista.

Estas colaboraciones suponen, sin duda, un estímulo adicional para los participantes en las jornadas, y contribuyen a dar una mayor difusión a los trabajos presentados en JENUI.

Sede de las jornadas

Con unos 3500 alumnos, cerca de 200 profesores, tres titulaciones de grado, cinco de máster, cuatro programas de doctorado y tres titulaciones de Ingeniería Informática, la ETSI Informática es uno de los centros de referencia de la Universidad de Sevilla y del Distrito Universitario Único de la Comunidad Andaluza.

En los últimos cinco años, esta escuela ha estado inmersa en un profundo proceso de adaptación al Espacio Europeo de Educación Superior, que ha supuesto no sólo la creación de nuevos planes de estudio de grado y de máster, sino también multitud de actividades que han tenido como objetivo ofrecer a nuestro profesorado la formación necesaria para acometer también un proceso de reforma pedagógica: cursos sobre competencias, diseño de escenarios pedagógicos, metodologías activas, creación de contenidos e impartición de clases en inglés, charlas invitadas de profesores de otras universidades que destacan por su implicación en estos temas. Entre estas actividades no podía faltar la organización de un evento de repercusión nacional en el que se den encuentro profesionales de la enseñanza universitaria de la Informática de toda España.

Hasta ahora, todas las actividades propuestas en esta escuela para la mejora de la docencia han contado con un gran apoyo y participación por parte del profesorado, por lo que no es de extrañar que en cuanto se propuso la idea de ser sede de esta edición JENUI 2011 fueran más de treinta los profesores que se ofrecieran a colaborar. Por tanto, estas JENUI cuentan con un Comité Organizador bien nutrido de profesores que han trabajado duro con el único

objetivo de que estas jornadas sean todo un éxito para usted. Esperamos que las disfrute y que constituyan de nuevo un punto de encuentro de docentes con ganas de aprender.

Las próximas JENUI

La próxima edición de JENUI tendrá sede en la Escuela Superior de Informática (ESI) de la Universidad de Castilla-La Mancha en el Campus de Ciudad Real. Este campus destaca especialmente por sus comunicaciones, disponibilidad hotelera y ciudad accesible en la que prácticamente es posible desplazarse caminando entre sus puntos de interés y el campus. En particular, destacamos la existencia de estación de tren de alta velocidad (AVE) que permite conectar directamente Ciudad Real con Madrid (50 min.) , Barcelona (3h 50 min.), Zaragoza (2h), Córdoba (1h) o Sevilla (1h 45 min.). Además, la distancia en tiempo desde la estación de AVE a la ESI es de aproximadamente 5 minutos en coche y poco más de 10 caminando. Estas posibilidades en materia de comunicaciones se complementan con la oferta de vuelos del Aeropuerto Central Ciudad Real.

Hay que significar que las fechas en las que está previsto celebrar las Jornadas coinciden con la celebración del Festival Internacional de Teatro Clásico de la ciudad de Almagro. Esto nos abre la posibilidad de disfrutar un programa social y cultural que aproveche al máximo esta coincidencia. Así pues, la organización va a trabajar con el objetivo de programar una visita guiada por la ciudad de Almagro y, especialmente, una obra de teatro para los asistentes a las Jornadas.

Sevilla, 5 de julio de 2011

El Comité de Programa y de Organización

ÍNDICE

Sesión 1A: Evaluación del alumnado I

- Propuesta de evaluación pura por competencias para estudios de Ingeniería Informática** 3
Xavi Canaleta (Universitat Ramon Llull), Agustín Zaballos (Universitat Ramon Llull), David Vernet (Universitat Ramon Llull)
- Evaluadores externos de proyectos de estudiantes: una experiencia en una asignatura de diseño de interfaces** 11
V. Javier Traver (Universitat Jaume I)
- Auto-evaluación discente como estrategia de aprendizaje ¿Cuestión de género?** 19
Carlos Tavares Calafate (Universitat Politècnica de València), Juan Carlos Cano (Universitat Politècnica de València), Pietro Manzoni (Universitat Politècnica de València)
- Práctica de desarrollo de interfaces hardware/software para el manejo de redes de sensores inalámbricos** 27
Ángel Jiménez Fernández (Universidad de Sevilla), Manuel Domínguez Morales (Universidad de Sevilla), Elena Cerezuela Escudero (Universidad de Sevilla), Rafael Paz Vicente (Universidad de Sevilla), José Ignacio Villar de Ossorno (Universidad de Sevilla), Gabriel Jiménez Moreno (Universidad de Sevilla), Alejandro Linares Barranco (Universidad de Sevilla)

Sesión 1B: Ingeniería del software Programación, algoritmos y estructuras de datos

- Aprendiendo a diseñar software usando juegos de mesa de licencia libre como enunciado de prácticas** 37
Pablo Trinidad (Universidad de Sevilla), Manuel Resinas (Universidad de Sevilla), Carlos Müller (Universidad de Sevilla), José A. Parejo (Universidad de Sevilla), Antonio Ruiz-Cortés (Universidad de Sevilla)
- El e-portfolio en la Ingeniería del Software como una herramienta de reflexión en el proceso de aprendizaje** 45
Reyes Grangel Seguer (Universitat Jaume I), Cristina Campos Sancho (Universitat Jaume I), Cristina Rebollo Santamaría (Universitat Jaume I), Inmaculada Remolar Quintana (Universitat Jaume I)
- Uso de metodologías activas en la implantación de IIP en el Grado en Informática de la UPV** 53
Natividad Prieto Sáez (Universitat Politècnica de València), Marisa Llorens Agost (Universitat Politècnica de València), Germán Moltó Martínez (Universitat Politècnica de València), Jon Ander Gómez Adrián (Universitat Politècnica de València), Mabel Galiano Ronda (Universitat Politècnica de València), Carlos Herrero Cucó (Universitat Politècnica de València)
- Corrección automática de ejercicios de estructuras de datos a través de una plataforma de e-learning** 61
Joan Surrell Sauri (Universitat de Girona), Josep Soler Masó (Universitat de Girona), Ferrán Prados Carrasco (Universitat de Girona), Imma Boada Oliveras (Universitat de Girona), Jordi Poch García (Universitat de Girona)

Sesión 2A: Evaluación del alumnado II

- Puntuación entre iguales para la evaluación del trabajo en equipo** 73
Alberto Abelló (Universitat Politècnica de Catalunya), Xavier Burgués (Universitat Politècnica de Catalunya)
- Una recapitulación sobre la autoevaluación de los alumnos en estudios de Informática: formas, utilidad y aplicación** 81
Agustín Cernuda del Río (Universidad de Oviedo), Miguel Riesco Albizu (Universidad de Oviedo)
- La sobre-evaluación** 91
Jesús Serrano Guerrero (Universidad de Castilla-La Mancha), Francisco Pascual Romero Chicharro (Universidad de Castilla-La Mancha), Emilio Fernández Viñas (Universidad de Castilla-La Mancha), José Ángel Olivas Varela (Universidad de Castilla-La Mancha)

Sesión 2B: Métodos pedagógicos innovadores I

Utilización de una herramientas de comunicación online para la mejora docente. Dos casos prácticos	101
Francisco Grimaldo Moreno (Universitat de València), Miguel Arevalillo Herráez (Universitat de València), Emilia López Iñesta (Universitat de València)	
Cómo empezar fácil con PBL	109
Miguel Valero García (Universitat Politècnica de Catalunya), Javier García Zubía (Universidad de Deusto)	
Telecolaboración interuniversitaria en prácticas de bases de datos	117
José Miguel Blanco Arbe (Euskal Herriko Unibertsitatea), César Domínguez Pérez (Universidad de la Rioja), Arturo Jaime Elizondo (Universidad de la Rioja), Ana Sánchez Ortega (Euskal Herriko Unibertsitatea)	

Sesión 3A: Arquitectura de computadores Sistemas operativos Sistemas distribuidos y paralelos

El abogado del diablo como técnica de trabajo cooperativa	127
Pablo Sánchez (Universidad de Cantabria)	
Entrenamiento de la creatividad y la innovación en la Ingeniería de Computadores basándose en la metodología de aprendizaje por proyectos	135
Gabriel Jiménez Moreno (Universidad de Sevilla), José Luis Sevillano Ramos (Universidad de Sevilla), Ángel Jiménez Fernández (Universidad de Sevilla), Rafael Paz Vicente (Universidad de Sevilla), Alejandro Linares Barranco (Universidad de Sevilla), Manuel Domínguez Morales (Universidad de Sevilla), Elena Cerezuela Escudero (Universidad de Sevilla), Lourdes Miro Amarante (Universidad de Sevilla)	
Sistemas Operativos Avanzados: de la clase magistral al entorno colaborativo	143
Joan Navarro (Universitat Ramon Llull), Xavi Canaleta (Universitat Ramon Llull), Andreu Sancho (Universitat Ramon Llull)	
Integración de elementos visuales y animaciones en las prácticas de programación paralela	151
José Miguel Mantas Ruiz (Universidad de Granada), Daniel Guerrero Martínez (Universidad de Granada), Sergio Rodríguez Lumley (Universidad de Granada)	

Sesión 3B: Organización curricular y planes de estudio Informática en otras carreras

Organización y gestión de una titulación del EEES	161
Fermín Sánchez Carracedo (Universitat Politècnica de Catalunya), María-Ribera Sancho Samsó (Universitat Politècnica de Catalunya), Josep Ramon Herrero Zaragoza (Universitat Politècnica de Catalunya)	
Una definición precisa del concepto "Nivel de Dominio de una Competencia" en el marco del Aprendizaje Basado en Competencias	169
Aurelio Bermúdez (Universidad de Castilla-La Mancha), Ismael García-Varea (Universidad de Castilla-La Mancha), María Teresa López (Universidad de Castilla-La Mancha), Francisco Montero (Universidad de Castilla-La Mancha), Luis de la Ossa (Universidad de Castilla-La Mancha), José Miguel Puerta (Universidad de Castilla-La Mancha), Tomás Rojo (Universidad de Castilla-La Mancha), José Luis Sánchez (Universidad de Castilla-La Mancha)	
La promoción de los estudios de Informática mediante la Olimpiada de Informática	177
Francisco José Alfaro Cortés (Universidad de Castilla-La Mancha), Juan José Pardo Mateo (Universidad de Castilla-La Mancha), José Pascual	
Experiencia docente en Informática Aplicada a la Traducción I del Grado de Traducción e Interpretación de la Universidad de Murcia	185
Antonio Ruiz Martínez (Universidad de Murcia), Mercedes Valdés Vela (Universidad de Murcia)	

Sesión 4A: Calidad y evaluación de la docencia Formación para la profesión

¿Cómo aprenden los estudiantes de Informática?	195
José Manuel Badía (Universitat Jaume In), Sergio Barrachina Mir (Universitat Jaume In), M. Asunción Castaño (Universitat Jaume In), Juan Carlos Fernández (Universitat Jaume In)	
Extensión de Moodle para la gestión colaborativa de proyectos	203
Raimon Lapuente (Universitat Politècnica de Catalunya), Jordi Piguillem (Universitat Politècnica de Catalunya), Enric Mayol (Universitat	
Marco para el desarrollo de la competencia transversal "Comunicación Eficaz"	213
David López (Universitat Politècnica de Catalunya)	
Docencia en Desarrollo Global de Software: Una Revisión Sistemática	221
Miguel J. Monasor (Universidad de Castilla-La Mancha), Aurora Vizcaíno (Universidad de Castilla-La Mancha), Mario Piattini (Universidad de Castilla-La Mancha)	

Sesión 4B: Arquitectura de computadores

Proyecto para el diseño, montaje y administración de un Cluster de computadores por parte de los estudiantes	231
Francisco Javier Fernández Baldomero (Universidad de Granada), Mancia Anguita López (Universidad de Granada)	
Simbiosis entre materias troncales y transversales: Análisis de una experiencia	241
Rosalía Peña (Universidad de Alcalá), Antonio Navidad (Universidad de Alcalá), Miguel Garre (Universidad de Alcalá)	
La evaluación personalizada como estrategia de motivación	249
Vicente Arnau Llombart (Universidad de Valencia), Miguel Arevalillo Herráez (Universidad de Valencia), José Claver Iborra (Universidad de Valencia)	
Píldoras formativas audiovisuales para el aprendizaje de Programación Avanzada	257
Luis Bengochea Martínez (Universidad de Alcalá)	

Sesión 5: Métodos pedagógicos innovadores II

Opera: Una herramienta soporte para el Aprendizaje Basado en Proyectos	267
María José Jiménez (Universidad de Sevilla), Pablo Fernández (Universidad de Sevilla), Rocío García (Universidad de Sevilla)	
Experiencia con WebQuest y herramientas Web 2.0 en la evaluación de competencias transversales	275
María Belén Vaquerizo García (Universidad de Burgos)	
Experiencia educativa para fomentar el aprendizaje autónomo a través de preguntas tipo test generadas por los alumnos	283
José Luis Risco Martín (Universidad Complutense de Madrid), Marcos Sánchez-Elez Martín (Universidad Complutense de Madrid), Inmaculada Pardines Lence (Universidad Complutense de Madrid)	

Sesión 6A: Evaluación del alumnado III

Modelo para la Evaluación de Competencias: Una experiencia aplicada a Fundamentos de Programación	295
Jesús Serrano Guerrero (Universidad de Castilla-La Mancha), Francisco Pascual Romero Chicharro (Universidad de Castilla-La Mancha), José Ángel Olivas Varela (Universidad de Castilla-La Mancha), Emilio Fernández Viñas (Universidad de Castilla-La Mancha)	
Autoevaluación y evaluación entre iguales en una asignatura de redes de ordenadores	303
María Cavas Toledo (Universidad de Málaga), Francisco Chicano García (Universidad de Málaga), Francisco Luna Valero (Universidad de Málaga), Luis Molina Tanco (Universidad de Málaga)	
Análisis de distintas metodologías de evaluación en prácticas de laboratorio en asignaturas de Redes de Computadores	311
Jaime Benjumea Mondéjar (Universidad de Sevilla), Ana V. Medina Rodríguez (Universidad de Sevilla), Octavio Rivera Romero (Universidad de Sevilla)	

Sesión 6B: Métodos pedagógicos innovadores III

Tutorizando el aprendizaje proactivo de nuevas tecnologías: Taller de programación Android	321
María José Blesa Aguilera (Universitat Politècnica de Catalunya), Amalia Duch Brown (Universitat Politècnica de Catalunya), Joaquim Gabarró Vallès (Universitat Politècnica de Catalunya), Hugo Hernández Pibernat (Universitat Politècnica de Catalunya), María José Serna Iglesias (Universitat Politècnica de Catalunya)	
Dos casos prácticos del uso de la Wiki en Ingeniería Informática: consideraciones para su uso en el Grado de Informática	329
Marta Elena Zorrilla Pantaleón (Universidad de Cantabria), Inés González Rodríguez (Universidad de Cantabria)	
Investigación práctica en educación: investigación-acción	337
Mercedes Marqués Andrés (Universitat Jaume I), Reina Ferrández Berruero (Universitat Jaume I)	

Sesión 7A: Telemática Multimedia e informática gráfica

Laboratorios virtuales de redes: SÍ, inténtelo en casa	347
Jesús Martínez (Universidad de Málaga), Juan José Ortega (Universidad de Málaga), José Alberto Fernández (Universidad de Málaga)	
Evaluación continua de alumnos mediante el uso de dispositivos móviles basados en iOS	355
Carmen Martínez Cruz (Universidad de Jaén), Antonio Jesús Rueda Ruiz (Universidad de Jaén)	
Introducción a la Informática Gráfica 3D a través de un Taller de Realidad Virtual y Patrimonio Artístico	363
Andrés Bustillo Iglesias (Universidad de Burgos), José Manuel Sáiz Diez (Universidad de Burgos), Laura Martínez García (Universidad de Burgos)	

Sesión 7B: Tecnologías de la información en la gestión empresarial Compromiso social y medioambiental

Docencia de Arquitectura Orientada a Servicios	373
Pablo García Sánchez (Universidad de Granada), Miguel Ángel López Montellano (Fundación I+D del Software Libre), Pedro Ángel Castillo Valdivieso (Universidad de Granada), Jesús González Peñalver (Universidad de Granada), María Isabel García Arenas (Universidad de Granada)	
Informática en la edad de oro	379
Rosa María Gil Iranzo (Universidad de Lleida)	
Diseños para todos. Experiencia en primero de grado	387
Juan José Escribano Otero (Universidad Europea de Madrid), María José Terrón López (Universidad Europea de Madrid)	

Sesión 8A: Calidad y evaluación de la docencia

Análisis empírico del papel de las competencias generales en el marco de los estudios superiores	395
Jessica Díaz Fernández (Universidad Politécnica de Madrid), Jennifer Pérez Benedí (Universidad Politécnica de Madrid), Agustín Yagüe Panadero (Universidad Politécnica de Madrid), Santiago Alonso Villaverde (Universidad Politécnica de Madrid), Javier Gil Rubio (Universidad Politécnica de Madrid)	
En busca del estudiante perdido: caracterización de los 'no presentados'	403
Ismael Sanz Blasco (Universitat Jaume I), María José Aramburu Cabo (Universitat Jaume I), Lledó Museros Cabedo (Universitat Jaume I), María Pérez Catalán (Universitat Jaume I)	
La mentoría como elemento de mejora en el proceso de la enseñanza- aprendizaje	411
José Antonio Gómez Hernández (Universidad de Granada), Eduardo Manuel Eisman Cabezas (Universidad de Granada)	

Sesión 8B: Fundamentos teóricos de la Informática Inteligencia artificial

- Sistema de Generación Personalizada de Ejercicios de Lógica Computacional** 413
José María Alcaraz Calero (Universidad de Murcia), Luis Daniel Hernández Molinero (Universidad de Murcia), Santiago Paredes Moreno (Universidad de Murcia)
- Enseñanza-aprendizaje en Visión por Computador: un enfoque motivador hacia el futuro profesional** 421
María Guijarro Mata-García (Universidad Complutense de Madrid), Gonzalo Pajares Martinsanz (Universidad Complutense de Madrid), P. Javier Herrera Caro (Universidad Complutense de Madrid), Jesús M. de la Cruz (Universidad Complutense de Madrid)

Pósteres y recursos docentes

- COETEST: Corrección óptica de exámenes de test en papel automática, rápida y económica** 431
Darío Álvarez Gutiérrez (Universidad de Oviedo), Díaz Sanzo Lucas (Universidad de Oviedo)
- El placer de las buenas citas** 435
Nieves Pavón Pulido (Universidad de Huelva), Julia González Rodríguez (Universidad de Extremadura), Joe Miró (Universitat de les Illes Balears)
- Un sistema de ayuda a la escritura de textos estructurados en LaTeX para estudiantes con déficit visual** 439
Esther Durá Martínez (Universitat de València), Juan Domingo Esteve (Universitat de València)
- Aplicación de nuevas técnicas docentes en la asignatura Sistemas Cliente-Servidor** 443
Ernesto Jiménez-Ruiz (University of Oxford), Dolores María Lladó Escriba (Universitat Jaume I), Rafael Berlanga Llavori (Universitat Jaume I), Olga Rajadell Rojas (Universitat Jaume I)
- Utilización de tinta digital para favorecer la implantación de una asignatura del grado en Ingeniería Informática** 447
Jorge Más-Estellés (Universitat Politècnica de València), José Vicente Benlloch-Dualde (Universitat Politècnica de València)
- Plan de acogida para los alumnos de nuevo ingreso en el grado de Ingeniería Informática** 451
Diego Cazorla López (Universidad de Castilla-La Mancha), Pedro Cuenca Castillo (Universidad de Castilla-La Mancha), Mere Macià Soler (Universidad de Castilla-La Mancha), José Pascual Molina Massó (Universidad de Castilla-La Mancha), José Miguel Puerta Callejón (Universidad de Castilla-La Mancha)
- Uso de redes sociales para mejorar el rendimiento de los alumnos con diferentes estilos de aprendizaje** 455
Sonia Vázquez (Universidad de Alicante), Irene Garrigós (Universidad de Alicante), José-Norberto Mazón (Universidad de Alicante), Estela Saquete (Universidad de Alicante), Rubén Izquierdo (Universidad de Alicante)
- Práctica de laboratorio sobre implementación Joystick HID-USB de interfaz con una emisora RC** 459
Rafael Paz Vicente (Universidad de Sevilla), Ángel Jiménez Fernández (Universidad de Sevilla), Manuel Domínguez Morales (Universidad de Sevilla), Elena Cerezuela Escudero (Universidad de Sevilla), José Ignacio Villar de Ossorno (Universidad de Sevilla), Miguel Ángel Rodríguez Jódar (Universidad de Sevilla), Alejandro Linares Barranco (Universidad de Sevilla)
- Sistema de co-diseño hardware/software basado en FPGA para la captura de video analógico a través del bus serie USB** 463
Ángel Jiménez Fernández (Universidad de Sevilla), Rafael Paz Vicente (Universidad de Sevilla), José Ignacio Villar de Ossorno (Universidad de Sevilla), Manuel Domínguez Morales (Universidad de Sevilla), Elena Cerezuela Escudero (Universidad de Sevilla), Miguel Ángel Rodríguez Jódar (Universidad de Sevilla)
- Entorno de soporte para el autoaprendizaje en el diseño de circuitos digitales** 471
David Bañeres Besora (Universitat Oberta de Catalunya), Iván Bermejo Izquierdo (Universitat Oberta de Catalunya), Robert Clarisó Viladrosa (Universitat Oberta de Catalunya), Josep Jorba Esteve (Universitat Oberta de Catalunya), Montse Serra Vizern (Universitat Oberta de Catalunya), Francesc Santanach Delisau (Universitat Oberta de Catalunya), Albert Rodríguez Garrido (Universitat Oberta de Catalunya)

Animaciones interactivas para la enseñanza y aprendizaje de los protocolos de coherencia de cachés	479
Alberto Alcón Laguéns (Universidad Jaume I), Sergio Barrachina Mir (Universidad Jaume I), Enrique S. Quintana Ortí (Universidad Jaume I)	
SyCS: una base de conocimiento de soporte a la adquisición de la competencia "Sostenibilidad y Compromiso Social"	487
David López (Universitat Politècnica de Catalunya), Fermín Sánchez (Universitat Politècnica de Catalunya), Jordi Garcia (Universitat Politècnica de Catalunya), Marc Alier (Universitat Politècnica de Catalunya)	
BlueState: un entorno para el aprendizaje de máquinas de estados de UML	495
Alfredo Ortigosa Aguilera (Universidad de Málaga), Carlos Rossi Jiménez (Universidad de Málaga)	
Animación interactiva de algoritmos para cursos de introducción a la programación	503
Lluís Ribas Xirgo (Universitat Autònoma de Barcelona)	
Hiperión: Sistema personalizado para la recomendación de actividades en base a competencias no adquiridas	511
Jesús Serrano Guerrero (Universidad de Castilla-La Mancha), Andrés Cerezo Guillén (Universidad de Castilla-La Mancha), Francisco Pascual Romero Chicharro (Universidad de Castilla-La Mancha), Víctor Hugo Menéndez Domínguez (Universidad de Castilla-La Mancha), José Ángel Olivas Varela (Universidad de Castilla-La Mancha)	
Diseño de material específico docente para el aprendizaje de microcontroladores y sistemas USB	519
Gabriel Jiménez Moreno (Universidad de Sevilla), Ángel Jiménez Fernández (Universidad de Sevilla), Manuel Domínguez Morales (Universidad de Sevilla), Elena Cerezuela Escudero (Universidad de Sevilla), Daniel Cascado Caballero (Universidad de Sevilla), Francisco Gómez Rodríguez (Universidad de Sevilla), Manuel Rivas Pérez (Universidad de Sevilla), Alejandro Linares Barranco (Universidad de Sevilla)	

Sesión 1A: Evaluación del alumnado I

Propuesta de evaluación pura por competencias para estudios de Ingeniería Informática

Xavi Canaleta, David Vernet, Agustín Zaballos

Departamento de Informática
La Salle, Universitat Ramon Llull
C/ Sant Joan de la Salle 42
08022 Barcelona
{xavic,dave,zaballos}@salleurl.edu

Resumen

Uno de los aspectos más relevantes dentro de las directrices de adaptación al EEES es la evaluación por competencias que debe hacerse en los diferentes estudios, ya sean grados o másteres. Si bien es cierto que la mayoría de los nuevos planes de estudio incorporan en sus memorias esta nueva forma de evaluación, la realidad es que o bien se sigue evaluando al alumno de la misma manera o bien los métodos de evaluación por competencias que se usan son una simple traducción de los métodos de evaluación tradicionales. En esta ponencia se pretende dar a conocer una propuesta de evaluación pura por competencias puesta ya en práctica con éxito en el Máster Universitario de Formación del Profesorado y adaptable a otros másteres universitarios, incluidos los de ingeniería en Informática.

Summary

One of the most important aspects in the process of adaptation to the EHEA is the evaluation of the competences of the different study programs of both undergraduate and master degree programmes. Although it is clear that the majority of new study plans have incorporated this new evaluation system into their study programmes, in many cases the student continues to be assessed in the same way as before or the evaluation methods are a simple translation of traditional evaluation methods. This paper presents a clear-cut proposal to evaluate competences which has already been implemented on the Teacher Training Master and which can be adapted to any type of university master degree course, including engineering subjects.

Palabras clave

Evaluación, competencias, EEES.

1. Motivación

Cuando en 1998 los ministros responsables de educación superior de 4 países firmaron, en la Sorbona [4], los principios de lo que tendría que convertirse en un espacio de educación superior común en Europa, seguramente no se esperaban el seguimiento que adquirió en los años posteriores.

Tan solo después de un año, en Boloña [2] se reunieron los ministros responsables de 19 países de la Unión Europea y, actualmente, en la última de las declaraciones ya se habían sumado 46 estados, sentando las bases políticas de lo que será el modelo educativo a seguir.

A lo largo de todos estos años, se ha ido construyendo el Espacio Europeo de Educación Superior (EEES) [3] el cual, aunque debería estar a pleno rendimiento el año 2010 [1], se halla en proceso de expansión y consolidación [11]. La adecuación de la educación superior a unos estándares comunes en toda Europa no es tarea sencilla y necesita del compromiso firme de todos los actores implicados pero, por encima de todo, de gobiernos e instituciones.

Uno de los principales objetivos de este nuevo marco es situar al alumno como protagonista. La educación pasa de estar basada en la enseñanza a estar basada en el aprendizaje. Por este motivo el sistema de créditos estará centrado en el alumno y medirá la carga de trabajo a realizar para conseguir unos objetivos determinados. Para hacer más transparentes y transferibles los títulos obtenidos por los alumnos europeos, los objetivos del estudiante estarán basados en los resultados del aprendizaje y las competencias a adquirir.

Las competencias son ítems de compleja evaluación. Es por este motivo que la implantación de las mismas a las asignaturas que conforman una titulación está resultando un proceso lento y en muchos casos difícil de llevar a cabo.

Pero la motivación de esta ponencia no es tratar de la evaluación de competencias dentro de una asignatura sino de explorar la posibilidad de una evaluación por competencias de todo un curso académico de forma transversal, centrada única y exclusivamente en dichas competencias. De este modo las asignaturas se convierten en un instrumento mediante el cual el alumno adquiere ciertas destrezas y habilidades a través de actividades diseñadas a tal efecto. Y éstas son evaluadas.

Las asignaturas pierden cierta relevancia para centrar todo el interés en las competencias a adquirir. Sólo así se potencia el espíritu de convertir el espacio europeo de educación superior en un entorno basado en el aprendizaje y no en la enseñanza. De aquí la propuesta de denominar a este tipo de evaluación “evaluación pura por competencias”.

Así pues, el objetivo de esta ponencia es presentar una propuesta de evaluación pura por competencias a nivel macro, para los estudios de Grado o Máster en Ingeniería Informática.

2. Antecedentes

La propuesta que se presenta para los estudios de Grado y Máster en Ingeniería Informática está basada en el diseño del plan de estudios [6] [7] [10] que se realizó para el Máster Universitario de Formación del Profesorado en los años 2008 y 2009. Y fue durante el curso académico 2009-2010 cuando dicho modelo se llevó a la práctica, impartiendo este tipo de estudios y más concretamente la especialidad de Tecnología.

Es por ello que en esta parte introductoria parece necesario la descripción de este Máster en la especialidad de Tecnología y su propuesta de evaluación pura por competencias, así de los resultados que se obtuvieron en su implantación, para poder analizar la adecuación y la viabilidad de este método a los estudios de Ingeniería Informática.

Como se observará la descripción detallada de la propuesta se hace a través de las explicaciones

de este Máster ya que, como posteriormente se verá, estas son aplicables a cualquier tipo de estudios universitarios de máster y en particular a los de Informática.

2.1. Máster de Formación del Profesorado

El Máster Universitario en Formación del Profesorado de Educación Secundaria Obligatoria y Bachillerato, Formación Profesional y Enseñanzas de Idiomas tiene una duración de 60 ECTS, de acuerdo con el artículo 5 de la Orden ECI/3858/2007 [5]. El diseño del plan de estudios es presencial en más de un 80% de los créditos totales del Máster, incluido el Practicum. Se estructura en dos semestres y debe cursarse en un curso académico a tiempo completo.

De acuerdo a la Orden ECI/3858/2007, la estructura general de máster puede resumirse como sigue:

- Un módulo básico de 15 créditos ECTS transversal a las diferentes especialidades del máster.
- Un módulo específico de 25 créditos ECTS para cada una de las especialidades
- Un periodo de prácticas de 14 créditos ECTS y un trabajo final de máster de 6 créditos ECTS, específicos de cada especialidad.

En la Tabla 1 se puede observar la distribución por módulos y las materias que componen cada módulo en el caso de la especialidad de Tecnología.

Como puede observarse, dentro del módulo específico se ha optado por descomponer las materias 4 y 5, cada una de ellas de 10 créditos ECTS, en dos submaterias con una carga de 5 ECTS cada una. De esta manera, se obtiene un modelo uniforme de 8 materias con la misma carga de trabajo (5 ECTS) contando los módulos genérico y específico.

Aparte queda el módulo del Practicum y Trabajo Final de Máster que, por sus características propias, tiene mucho más seguimiento y menos estructuración en formato de clases lectivas. Este acompañamiento se realiza mediante seminarios en pequeño grupo (10-12 alumnos) y tutorías individuales. En el caso del Practicum dichas tutorías se realizan tanto por parte de los tutores de la universidad como por los tutores de los centros de prácticas, existiendo una

coordinación importante entre ellos también a nivel de evaluación.

MÁSTER EN FORMACIÓN DEL PROFESORADO
MÓDULO GENÉRICO (15 ECTS) DESARROLLO DE LA PERSONA, EDUCACIÓN Y SOCIEDAD
MATERIA 1 (5 ECTS) Aprendizaje y desarrollo de la personalidad
MATERIA 2 (5 ECTS) Procesos y contextos educativos
MATERIA 3 (5 ECTS) Sociedad, familia y educación
MÓDULO ESPECÍFICO (25 ECTS) FORMACIÓN AVANZADA PARA LA ENSEÑANZA DE LA TECNOLOGÍA
MATERIA 4 (10 ECTS) Complementos para la formación disciplinar: Tecnología
4.1: Objetivos y ubicación de la Tecnología (5 ECTS)
4.2: Contenidos curriculares complementarios y avanzados de la Tecnología (5 ECTS)
MATERIA 5 (10 ECTS) Aprendizaje y enseñanza de la tecnología
5.1: La tecnología en el contexto social (5 ECTS)
5.2: Metodologías de aprendizaje y métodos de evaluación (5 ECTS)
MATERIA 6 (5 ECTS) Innovación docente e iniciación a la investigación educativa en Tecnología
PRACTICUM Y TRABAJO FIN DE MÁSTER (20 ECTS)
MATERIA 7 (14 ECTS) Prácticas en un centro educativo
MATERIA 8 (6 ECTS) Elaboración y defensa de un Trabajo Final

Tabla 1. Estructura del Máster.

2.2. Evaluación en el Máster

Lo más relevante del Máster Universitario en Formación del Profesorado para esta ponencia es su método de evaluación pura por competencias.

Tal y como queda definido en el plan de estudios, el Máster debe de ser evaluado en base a 3 calificaciones numéricas, una para cada uno de los módulos que lo componen. Finalmente, se obtendrá una calificación global del Máster

realizando la media ponderada a partir de dichas notas.

$$MUFP = 15/60 \cdot MG + 25/60 \cdot ME + 20/60 \cdot MPTFM$$

donde *MUFP* es la nota final del Máster, *MG* es la nota del módulo genérico, *ME* la nota obtenida del módulo específico y *MPTFM* la nota del módulo de Practicum y Trabajo Final de Máster.

Pero la parte más novedosa de la evaluación del Máster aparece cuando se especifica el método de evaluación para cada uno de los tres módulos del Máster. Cada módulo tiene asignadas unas competencias que el estudiante debe adquirir. Así pues, para un módulo determinado se hace un análisis para ver qué materias pueden desarrollar cada competencia. Cada responsable de una materia hace su propuesta al coordinador del Máster, que será el responsable de garantizar que todas las competencias se trabajen como mínimo en una materia y, a ser posible, en más de una.

La Tabla 2 muestra como ejemplo la distribución de las competencias específicas de la especialidad de Tecnología dentro del Módulo Específico del Máster.

Comp.	M4.1	M4.2	M5.1	M5.2	M6	
E18	X				X	2
E46	X					1
E47	X					1
E48	X					1
E49		X				1
E50	X	X		X	X	4
E51	X	X	X	X	X	5
E52	X	X			X	3
E53		X	X		X	3
E54		X	X	X	X	4
E55		X	X	X	X	4
E56		X	X		X	3

Tabla 2. Distribución de competencias (Comp: Competencia; M: Materia)

A partir de esta distribución ya es posible saber dónde serán evaluadas cada una de las

competencias y por quién. Así pues, al final del máster cada responsable de la materia se encarga de introducir las calificaciones de las competencias que evalúa en un sistema de información como el que muestra de ejemplo la Tabla 3. Se han escogido las competencias específicas E46, E50 y E51 que aparecían en la anterior relación de la Tabla 2.

Se puede observar que debido a que la competencia E46 sólo es evaluada por la Materia 4.1 la nota final de la competencia es directamente la calificación facilitada por el responsable de dicha materia. Mientras que, por ejemplo, para la competencia E51, que es evaluada en las cinco materias del módulo específico, su nota final se calcula a partir de las calificaciones facilitadas por los responsables de todas las materias del módulo específico del Máster.

Finalmente, para obtener la calificación final de un módulo, se realizará la media de todas las competencias evaluadas en ese módulo. En el ejemplo del módulo específico de la especialidad de Tecnología del Máster la nota final del módulo se obtiene realizando la media aritmética de las 12 competencias que el alumno debía adquirir en dicho módulo.

De esta manera, se puede deducir fácilmente que las materias no tienen una calificación propia. No se suspenden o se aprueban, sino que se aprueban o suspenden sus competencias asociadas. La materia sólo es un medio que facilita la adquisición de competencias dentro del módulo, por ello no tiene sentido dar una calificación a la misma. De aquí el término de “evaluación pura por competencias” como se ha denominado a esta evaluación.

2.3. Evaluando competencias en las materias

Como anteriormente ya se ha comentado, aunque no es el objeto de la presente ponencia detallar los métodos de evaluación de competencias dentro de una materia, de manera breve se describen las líneas de actuación que se han seguido en este Máster para la evaluación de competencias dentro de cada materia:

- Se recomienda el uso de técnicas de *learning by doing*, *problem based learning* y *project based learning* con el objetivo de favorecer la evaluación continua de competencias.
- Se deben diseñar actividades orientadas a la adquisición de competencias y su evaluación.
- Las actividades han de favorecer la evaluación formativa; es decir, han de permitir al alumno ver sus puntos fuertes y puntos débiles durante su proceso de aprendizaje para poder hacer potenciar su esfuerzo en las competencias que tenga más dificultad en adquirir.

2.4. Otras consideraciones para la evaluación

Para acabar la breve descripción del método de evaluación es necesario hacer mención a los criterios de promoción y recuperación relacionados con este tipo de evaluación.

En el sistema tradicional de evaluación un alumno tiene como base las calificaciones de las asignaturas o materias. Estas determinan si es apto y puede promocionar a un curso académico posterior, o bien si no tienen el nivel suficiente y, por lo tanto, debe repetir una o más materias.

Con el sistema de evaluación pura por competencias estos criterios no pueden aplicarse dado que no existen las calificaciones finales por materias o asignaturas.

Nombre	E46	E50					E51					
	M41	M41	M42	M52	M6	F	M41	M42	M51	M52	M6	F
AI 1	8,75	8,33	5,67	5,20	4,00	5,80	8,92	6,00	8,94	8,70	6,00	7,71
AI 2	6,00	6,00	8,67	6,80	7,50	7,24	6,00	7,67	8,10	4,80	9,00	7,11
AI 3	8,08	8,17	7,33	5,40	8,50	7,35	7,75	8,33	8,42	5,40	9,00	7,78
AI 4	6,08	6,00	8,00	5,20	3,00	5,55	5,33	7,00	8,94	8,90	6,00	7,24
AI 5	6,92	6,50	5,67	7,00	10,00	7,29	7,08	6,33	8,20	9,00	5,00	7,12

Tabla 3. Evaluación de competencias.

El primer criterio importante a destacar es que un alumno dentro de un módulo debe de superar todas las competencias que este tiene asociadas. La opinión se fundamenta en que todas las competencias son importantes pero ninguna es imprescindible y que todas las competencias son imprescindibles pero ninguna es importante.

Así pues, el hecho de no superar con éxito una competencia dentro de un módulo genera un camino de ingeniería inversa para su posible recuperación. Si un alumno no obtiene una calificación suficiente para poder considerar que ha adquirido una determinada competencia, por el propio diseño del Máster, el coordinador o profesor puede determinar en qué materias se ha evaluado dicha competencia. Y, de la misma manera, qué actividades asociadas a dicha competencia han hecho que el alumno obtenga una calificación insuficiente. Así pues, quedan fijadas las actividades a superar o repetir, para poder considerar adquirida dicha competencia.

Con todo hay una excepción a este criterio: no es posible dentro del mismo curso académico poder recuperar las competencias asociadas al Practicum, ya que esto obligaría a la repetición de las prácticas en un centro educativo, hecho inviable dentro del mismo año académico.

2.5. Resultados obtenidos y conclusiones

La aplicación del sistema de evaluación pura por competencias dentro del Máster de Formación del Profesorado (MUFP) durante el curso académico 2009-2010 ha presentado unos resultados altamente satisfactorios, tanto desde el punto de vista del estudiante como desde el punto de vista del profesorado implicado.

Desde el punto de vista del alumnado se resumen las siguientes conclusiones facilitadas a través de las encuestas de satisfacción:

- El alumno considera el método de evaluación altamente objetivo ya que una calificación no depende sólo de un único profesor o materia.
- La evaluación formativa durante el transcurso del curso académico ayuda claramente a orientar al alumno en sus carencias.
- La evaluación final a través del informe de competencias es de más utilidad que el certificado de calificaciones numéricas clásico por materias o asignaturas.

Por lo que se refiere al profesorado implicado y Dirección, cabe decir que éste fue seleccionado teniendo en cuenta su alta vocación de cara a la

docencia y su experiencia profesional en el ámbito que nos ocupa (secundaria, bachillerato o formación profesional) y su interés en la innovación docente. Merecen especial atención los siguientes comentarios:

- Aplicación de la propuesta en el diseño de cada materia para poder seguir el método de evaluación pura por competencias. Esto ha sido más viable debido a tener que diseñar desde cero las materias.
- En caso de tener que hacer una adecuación o adaptación de contenidos esto habría dificultado mucho el diseño. Es más costoso adaptar que diseñar desde cero en este caso.
- Buena aceptación de la pérdida de peso o relevancia de la materia para potenciar las competencias. No ha sido traumático para este profesorado, aunque cabe destacar su alta predisposición para esta experiencia docente.
- Sugerencia para tener en cuenta el peso de cada materia dentro de la evaluación de una competencia. Actualmente se calcula mediante una media aritmética sin ponderar en ningún caso el peso de cada materia en la misma.

Finalmente, se hace necesario destacar también la alta motivación y predisposición del grupo de estudiantes que cursaron el Máster durante el curso 2009-2010, lo cual sin duda ha facilitado la implantación de esta propuesta.

3. Adaptación del método a los estudios de Informática

Una vez implantado con éxito el modelo de evaluación pura por competencias y ya dentro del segundo año de experiencia en el mismo, se aborda ahora desde Dirección la posibilidad de adaptación de este método para estudios de Ingeniería y más concretamente para los correspondientes a la disciplina de Informática.

La adecuación de este método se debe estudiar para las dos titulaciones existentes, el Grado en Ingeniería Informática (GII) y el Máster en Ingeniería Informática (MII). A priori todo parece indicar que el Máster será, por analogías, más sencillo de adaptar que el Grado. En primer lugar el MII tiene una duración parecida a la del MUFP y también está partiendo de un diseño completamente nuevo y no de una adecuación de los estudios anteriores de Ingeniería Técnica o Superior.

Así pues, la ponencia que aquí se presenta detallará más la viabilidad del método para los estudios de Máster y, aunque citando algunas de las dificultades que se observan para los estudios de Grado, se propondrá como líneas futuras un análisis más exhaustivo para esta titulación.

3.1. Máster en Ingeniería Informática

El Máster en Ingeniería Informática de la universidad es una titulación que actualmente está en proceso de trámite para su aprobación.

La estructura del MII se estructura en 3 módulos y sus correspondientes materias, como se puede observar en la Tabla 4.

Las competencias generales y específicas que los estudiantes deben adquirir durante sus estudios y que serán exigibles para otorgar el título se han definido teniendo en cuenta la legislación vigente para titulaciones de máster [7], la Resolución 12977 de la Secretaría General de Universidades [9] por la que se establecen las recomendaciones para la propuesta de memorias y el Real Decreto 861/2010 [8] por la que se modifica el RD 1393/2007 que establece la ordenación de las enseñanzas universitarias oficiales.

Módulos	Materias	ECTS
Dirección y Gestión	Dirección y Gestión	12
Tecnologías informáticas	Sistemas De Información	20
	Sistemas Informáticos	20
	Redes y Tecnologías de Internet	8
Trabajo Final de Máster	Seminarios TIC	20
	Trabajo Final de Máster	10

Tabla 4. Estructura del Máster.

En la memoria también quedan descritas las competencias asociadas al MII. Concretamente, se detallan 10 competencias generales y 16 competencias específicas. También se especifica

cómo estas quedan distribuidas dentro de cada uno de los módulos. Como se observa en la Tabla 5, las competencias quedan distribuidas en cantidad proporcionalmente al número de créditos de cada módulo.

Módulos	Generales	Específicas
Dirección y Gestión	7	3
Tecnologías informáticas	6	12
Trabajo Final de Máster	6	3

Tabla 5. Estructura del Máster.

Así pues, el sistema de evaluación pura por competencias diseñado y aplicado para el Máster de Formación del Profesorado es completamente aplicable al Máster en Ingeniería Informática. La estructura general del MII y sus especificaciones coinciden plenamente con las del MUFP. Si bien es cierto que las materias del MII tienen un volumen en créditos muy superior a las del MUFP, esto no debería dificultar el diseño basado en competencias. Además, el diseño del MII posibilita que en su implantación real las materias más voluminosas en número de créditos puedan ser desglosadas en submaterias de menor carga.

Adicionalmente, tal y como se especifica en la memoria presentada para su evaluación, predomina el método del aprendizaje basado en proyectos, también muy usado en el MUFP y que ha facilitado la puesta en marcha de dicho sistema de evaluación.

Con todo, uno de los escollos importantes que puede existir es la aceptación del sistema de evaluación pura por competencias por parte del profesorado implicado. Las reticencias, vistas como una pérdida de poder académico por cierta parte del profesorado, son un punto a tener en cuenta. Es clave la implicación de la Dirección y su apoyo para garantizar el éxito del sistema a implantar. Se precisa de una buena motivación, explicación y formación del profesorado del máster para facilitar su adaptación a los cambios propuestos.

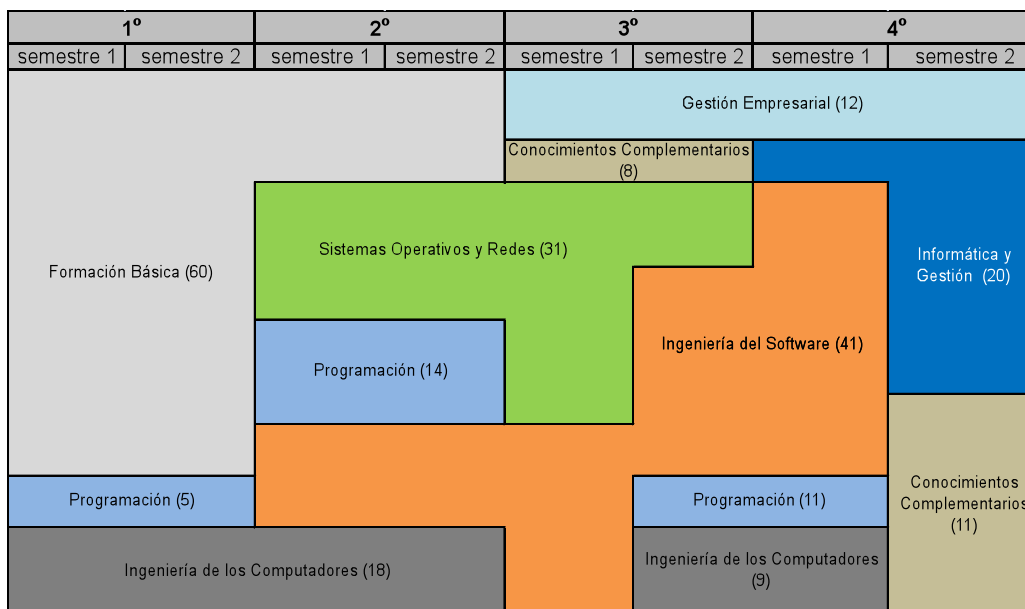


Figura 1. Estructura del Grado en Ingeniería Informática por módulos.

3.2. Grado en Ingeniería Informática

El Grado de Ingeniería Informática es una titulación con unas especificaciones muy distintas a las descritas en los casos anteriores.

Primeramente la titulación tiene 240 créditos ECTS distribuidos en 4 cursos académicos. La Figura 1 muestra el detalle del plan de estudios de grado descompuesto en los diferentes módulos que lo forman y su distribución temporal semestral.

A partir de estas especificaciones ya se pueden detectar una serie de problemas que dificultarán la adecuación del método de evaluación pura por competencias:

- Los módulos están distribuidos dentro de cursos académicos diferentes, incluso en cursos académicos no consecutivos. Esto imposibilitaría la evaluación pura por competencias dentro de cada módulo dentro de un mismo curso académico.
- A los criterios especificados para la recuperación de competencias no adquiridas, deben añadirse ahora los criterios de promoción al siguiente curso académico.
- El cálculo de la calificación final para una competencia dentro de un módulo debe replantearse, ya que una competencia puede

evaluarse en cursos diferentes y es probable que la calificación de una competencia, por ejemplo, evaluada en un tercer curso deba actualizar (o no) la calificación de la misma hecha en un curso anterior.

En un primer análisis superficial se puede intuir que algunos de estos problemas son lo suficientemente críticos para concluir que la adaptación de esta propuesta para estudios de grado no será sencilla ni inmediata ni tan siquiera se puede afirmar si será viable.

4. Conclusiones

Se ha presentado un método de evaluación pura por competencias que se puede considerar novedoso y que rompe con el esquema tradicional de evaluación por asignaturas de los estudios universitarios actuales. Este método ha sido puesto en práctica con éxito en el Máster de Formación del Profesorado y se considera que es claramente aplicable a los estudios de Máster de Ingeniería Informática.

El sistema ha sido bien aceptado y favorece la focalización de la acción en el aprendizaje de competencias del alumno pero presenta muchas dificultades para su aplicación a los estudios de grado.

Referencias

- [1] BUDAPEST-VIENNA DECLARATION ON THE EUROPEAN HIGHER EDUCATION AREA [en línea], Budapest-Viena, 12 de marzo de 2010, disponible en red: http://www.ond.vlaanderen.be/hogeronderwijs/bologna/2010_conference/documents/Budapest-Vienna_Declaration.pdf
- [2] DECLARACIÓ CONJUNTA DELS MINISTRES EUROPEUS D'EDUCACIÓ: L'Espai Europeu d'Ensenyament Superior [en línea], Bologna, 19 de junio de 1999, disponible en red: http://www.gencat.cat/diue/doc/doc_1575914_9_1.pdf.
- [3] DECLARACIÓ CONJUNTA DELS MINISTRES EUROPEUS D'EDUCACIÓ: Vers l'Espai Europeu d'Educació Superior [en línea], Praga, 19 de mayo de 2001, disponible en red: http://www.gencat.cat/diue/doc/doc_4031596_0_1.pdf.
- [4] DECLARACIÓN DE LA SORBONA: Declaració conjunta per l'harmonització del disseny del Sistema d'Educació Superior Europeu [en línea], París, 25 de mayo de 1998, disponible en red: http://www.gencat.cat/diue/doc/doc_1785244_9_1.pdf
- [5] ORDEN ECI/3858/2007 [en línea], de 27 de diciembre, *por la que se establecen los requisitos para la verificación de los títulos universitarios oficiales que habiliten para el ejercicio de las profesiones de Profesor de Educación Secundaria Obligatoria y Bachillerato, Formación Profesional y Enseñanzas de Idiomas*, disponible en red: <http://www.boe.es/boe/dias/2007/12/29/pdfs/A53751-53753.pdf>
- [6] REAL DECRETO 1834/2008 [en línea], de 8 de noviembre, *por el que se definen las condiciones de formación para el ejercicio de la docencia en la educación secundaria obligatoria, el bachillerato, la formación profesional y las enseñanzas de régimen especial y se establecen las especialidades de los cuerpos docentes de enseñanza secundaria*, disponible en red: <http://www.boe.es/boe/dias/2008/11/28/pdfs/A47586-47591.pdf>
- [7] REAL DECRETO 1393/2007 [en línea], de 29 de octubre, *por el que se establece la ordenación de las enseñanzas universitarias oficiales*, disponible en red: <http://www.boe.es/boe/dias/2007/10/30/pdfs/A44037-44048.pdf>
- [8] REAL DECRETO 861/2010 [en línea], de 2 de julio, *por el que se modifica el Real Decreto 1393/2007, de 29 de octubre, por el que se establece la ordenación de las enseñanzas universitarias oficiales*, disponible en red: <http://www.boe.es/boe/dias/2010/07/03/pdfs/BOE-A-2010-10542.pdf>
- [9] RESOLUCIÓN de 8 de junio de 2009 [en línea], de la Secretaría General de Universidades, *por la que se da publicidad al Acuerdo del Consejo de Universidades, por el que se establecen recomendaciones para la propuesta por las universidades de memorias de solicitud de títulos oficiales en los ámbitos de la Ingeniería Informática, Ingeniería Técnica Informática e Ingeniería Química*, disponible en red: <http://www.boe.es/boe/dias/2009/08/04/pdfs/BOE-A-2009-12977.pdf>
- [10] RESOLUCIÓN de 17 de diciembre de 2007 [en línea], de la Secretaría de Estado de Universidades e Investigación, *por la que se publica el Acuerdo de Consejo de Ministros de 14 de diciembre de 2007, por el que se establecen las condiciones a las que deberán adecuarse los planes de estudios conducentes a la obtención de títulos que habiliten para el ejercicio de las profesiones reguladas de Profesor de Educación Secundaria Obligatoria y Bachillerato, Formación Profesional y Enseñanzas de Idiomas*, disponible en red: <http://www.boe.es/boe/dias/2007/12/21/pdfs/A52851-52852.pdf>
- [11] SURSOCK, A., SMIDT, H., Trends 2010: A decade of change in European Higher Education [en línea]. EUA, 2010, disponible en red: <http://www.eua.be/eua-work-and-policy-area/building-the-european-highereducation-area/trends-in-european-higher-education/trends-vi.aspx>

Evaluadores externos de proyectos de estudiantes: una experiencia en una asignatura de diseño de interfaces

V. Javier Traver, vtraver@uji.es

Dep. Lenguajes y Sistemas Informáticos, Universitat Jaume I
Av. Vicent Sos Baynat, s/n, E12071 Castellón (España)

Resumen

El aprendizaje por proyectos es una interesante metodología de enseñanza-aprendizaje. Sin embargo, una dificultad práctica para su efectividad es lograr que los estudiantes se motiven y se impliquen lo suficiente. Ofrecer un premio al mejor proyecto de los estudiantes es, además de una forma de incentivar la calidad de dichos proyectos, una excelente coyuntura para explorar la participación de expertos, externos a la asignatura. Dicha participación aporta, además de ayuda para fallar el premio, realimentación a estudiantes y profesorado, así como mayores garantías de calidad e imparcialidad en la evaluación. Por último, la valoración externa ofrece un mecanismo para indagar una idea poco considerada habitualmente: la conveniencia de separar dos roles (académico y social) asociados al profesorado. El artículo detalla el proceso seguido y los resultados obtenidos, y reflexiona sobre las posibilidades y dificultades que esta experiencia alberga, aspectos que pueden ser de utilidad para iniciativas similares.

Palabras clave

Aprendizaje basado en proyectos; aprendizaje cooperativo; evaluación anónima por expertos externos; premio al mejor trabajo.

1. Introducción

Desde el curso 2003–04 hemos estado impartiendo una asignatura de diseño de interfaces de usuario en tercero de Ingeniería Informática en la Universitat Jaume I (UJI). Se trata de una asignatura obligatoria, cuatrimestral, de 4,5 créditos. En ella concedemos mucha atención al tema del diseño (y no tanto a la programación), así como al aprendizaje cooperativo y basado en proyectos. Creemos que estas metodologías, que pueden ser beneficiosas en un amplio conjunto de asignaturas, lo son de modo especial en esta asignatura, por las características del desarrollo

profesional de la disciplina “human-computer interaction” (HCI), donde la componente de creatividad y trabajo en equipo cobran especial relevancia. Cada equipo de 3–4 estudiantes desarrolla un diseño a partir de un tema propuesto por los profesores. Dada la importancia que concedemos al proyecto, su peso en la nota final es del 70%. El principal producto del proyecto es un informe de una extensión máxima de 12 páginas, siguiendo una plantilla en \LaTeX y unas directrices proporcionadas por los profesores. Incidimos mucho en la importancia de la escritura científico-técnica que, como se les insiste, es *también* una interfaz (entre el autor y su audiencia). Opcionalmente pueden aportar un código correspondiente a la implementación (parcial) de un prototipo de su interfaz.

Una de las acciones del plan estratégico de nuestro departamento persigue promocionar y premiar las mejores prácticas de los estudiantes en asignaturas. En el curso 2009–10 se obtuvo una pequeña financiación (400 euros) del plan estratégico para la asignatura de diseño de interfaces. La idea era, claro está, motivar a los estudiantes por esforzarse en realizar un trabajo de especial calidad.

Al reflexionar sobre la forma de decidir estos premios, surgió la idea de recurrir a expertos en el tema de la interacción, ya sea del contexto académico o profesional, como un apoyo en el fallo del premio. Esta idea resultaba realmente atractiva por numerosos motivos. Por un lado, además de su utilidad para la decisión del premio, el hecho de contar con un panel de expertos ofrecía, de cara a los estudiantes, mayores garantías de seriedad e imparcialidad. Por otro lado, al margen del tema de los premios, contar con otras opiniones sobre los trabajos de los estudiantes puede constituir una valiosa información tanto para el profesor (por contar con otras opiniones para la evaluación en su vertiente sumativa), como para los estudiantes (como realimentación en su evaluación formativa). Por último, pero no menos importante, contar con expertos externos suponía una muy buena oportunidad de poner a prueba una idea con-

siderada muchas veces desde hacía varios años, pero nunca llevada a cabo. Nos referimos a la posible conveniencia de la separación de roles del profesor: enseñar y evaluar (sumativamente).

Para evitar la confusión que supone el término «evaluación», por sus dos posibles connotaciones (formativa y sumativa), nos referiremos a la evaluación sumativa como el papel «social» del profesor, puesto que esa es la utilidad y la necesidad última de la evaluación sumativa: la validación de conocimientos y competencias de cara a la sociedad y que se termina concretando en un título oficial. Por otro lado, al genuino papel de enseñar y de ayudar a aprender lo denominamos formativo, educativo o académico.

Las recientes tendencias en educación abogan por hacer que el docente pase de ser el sabio o el controlador (“ruler”) a ser «guía», «entrenador», o incluso «compañero» (“*from the sage on the stage to the guide on the side*”) [5]. Este cambio, de hecho, se plasma en el papel que el profesor debe jugar en técnicas como el aprendizaje cooperativo, por proyectos o basado en problemas. Sin embargo, estas interesantes ideas pueden resultar difícilmente aplicables o efectivas si los estudiantes perciben una dualidad en el profesor: en el lado positivo, los estudiantes pueden ver, en el mejor de los casos, que el profesor pone cierto empeño en enseñarles; por otro lado, el negativo, los estudiantes perciben en el profesor el «juez» del que, en última instancia, va a depender que ellos terminen aprobando o suspendiendo o qué nota obtengan. No es difícil reconocer que esta dualidad de roles no resulta, en general, fácilmente compatible, al menos desde la perspectiva del estudiante, y puede que también del profesor.

Dado que conciliar ambos roles (formativo y social) en una misma persona (el profesor) no resulta una tarea trivial, cabe plantearse la separación de tales roles. Usando la analogía deportiva “atleta es a entrenador como estudiante es a profesor”, es fácil darse cuenta de que resulta discutible que tenga que ser la misma persona quien asuma ambos roles: el entrenador hace todo lo que está en sus manos en que el deportista mejore sus condiciones físicas, pero luego serán los jueces de una competición quienes valoren la actuación del deportista, no el propio entrenador.

Creemos que la percepción que los estudiantes tienen del profesorado mejora sustancialmente si se

desvinculan las dos facetas o responsabilidades que, quizás poco afortunadamente, ahora recaen en la misma persona. Un indicio que aporta cierta evidencia al respecto es nuestra experiencia personal: hemos comprobado que hemos obtenido mejores resultados en las encuestas de los estudiantes en aquellas asignaturas donde no hemos sido sus responsables y hemos impartido docencia sólo en grupos de laboratorio. En estas condiciones, los estudiantes nos han visto más próximos y solidarios que los profesores que, desde su responsabilidad en la asignatura, imponen las «reglas del juego» y ejercen su rol de «juez». Por tanto, y en relación con el tema principal de este artículo, la participación de expertos externos en la valoración de trabajos de los estudiantes es un posible mecanismo concreto de explorar, a pequeña escala, la viabilidad, interés, alcance y limitaciones de la separación de ambos roles.

2. Planteamiento y desarrollo

A continuación se discuten los principales aspectos relacionados con el planteamiento y desarrollo de la evaluación de trabajos mediante revisores externos. Algo que se tuvo claro desde el principio fue el que la revisión fuese doblemente anónima: ni los revisores conocerían la identidad de los estudiantes ni éstos la de los revisores (y así se informó a ambas partes). Otra consideración fue la de evitar implicar a demasiados revisores ni cargar con demasiado trabajo a un revisor concreto. Pero, al mismo tiempo, era importante conseguir que cada trabajo fuese evaluado por, al menos, dos revisores distintos, para contrastar opiniones.

Respecto a alguna experiencia similar [3], en nuestro caso no se plantea ningún «congreso», ni actas; los revisores son expertos externos, no miembros del propio departamento; y lo que se somete a revisión son los propios proyectos de la asignatura, no actividades voluntarias adicionales.

Para buscar revisores se recurrió, en parte, a personas españolas que los autores de este artículo conocían por algún motivo (e.g. de algún congreso). La búsqueda de revisores se prolongó desde finales de marzo hasta finales de mayo. Al final, siete personas (5 mujeres y 2 hombres) consideraron razonable el volumen de trabajo que se esperaba de ellos (2–3 informes de 12 páginas cada uno), así como el tiempo que se les daría para preparar la evaluación

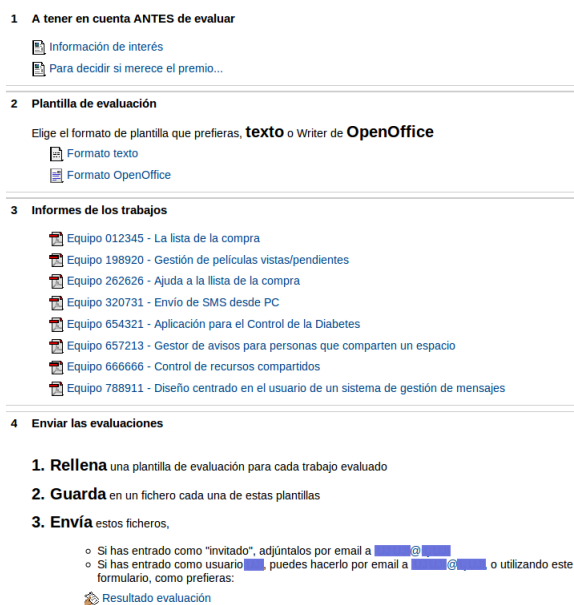


Figura 1: Página web para la revisión

(unas tres semanas) y confirmaron su participación. Todas estas personas están muy relacionadas (sólo uno de ellos lo está en menor medida) con HCI por impartir docencia, por investigar o por trabajar profesionalmente en esta disciplina, aspecto éste muy importante para asegurar una revisión con garantías de calidad.

Para gestionar la comunicación con los revisores se preparó una página web de evaluación (distinta a la de la asignatura y no accesible para los estudiantes), utilizando la plataforma Moodle (<http://moodle.org>), como se muestra en la Fig. 1. En ella se colgaron los PDFs correspondientes a los informes (anónimos) de los equipos de estudiantes (Punto 3, Fig. 1). Para facilitar la labor a los revisores y, al mismo tiempo, para que sus informes de evaluación fuesen de la mayor utilidad a los profesores, se diseñó una plantilla de evaluación, tanto en formato de texto llano como en formato OpenOffice, de modo que los revisores pudieran elegir lo que más les interesara (Punto 2, Fig. 1). Sin embargo, tan importante o más que esta plantilla era ponerles en situación de los aspectos en los que se incidía en la asignatura y, en definitiva, dejar tan claro como fuera posible los criterios con los que debían reali-

Conviene saber algunas cosas sobre la asignatura «Entornos de usuario» y cómo se plantea el trabajo para entender mejor la naturaleza de los mismos y poderlos valorar con la perspectiva adecuada.

Si no tienes mucho tiempo, lee al menos este resumen. . .

- Equipos de 3 estudiantes para hacer un trabajo de diseño que se les asigna (no lo eligen ellos)
- Se concede mucha importancia a un buen análisis de tareas
- Pueden programar poco o nada: se concede mucha más importancia al diseño y a la usabilidad que a la programación
- Se valoran múltiples evaluaciones de usabilidad, con y sin usuarios
- Se insiste en la importancia de escribir bien al preparar el informe
- El trabajo/proyecto representa un 70 % de la nota para el estudiante.

Figura 2: El resumen que los revisores debían leer antes de proceder a la lectura y evaluación de los informes. La descripción extensa se omite por limitaciones de espacio.

zar la revisión (Punto 1, Fig. 1). Esta información se resumió de la mejor forma posible en la página web de evaluación, teniendo en cuenta además aspectos de usabilidad. Para aligerar la tarea a los revisores más ocupados, se preparó un resumen aún más corto, buscando que la información más esencial no dejase de tenerse en cuenta (Fig. 2).¹

La plantilla para evaluar los trabajos se articulaba en cuatro preguntas: impresión general, usabilidad, HCI e informe. De cada pregunta debía elegirse una opción entre las disponibles (Cuadro 1) y justificar la elección mediante texto libre. A continuación se hacían tres preguntas abiertas acerca de aspectos positivos y negativos del trabajo, así como otras recomendaciones y observaciones para los estudiantes. La siguiente pregunta hacía referencia a si consideraban al trabajo merecedor del premio, con dos opciones: «Definitivamente sí» y «Claramente no». Se les pedía no dejar sin contestar esta pregunta (da-

¹Una idea para llevar un paso más allá la idea de separación de roles del profesor, sugerida por un revisor de este artículo, sería que los propios revisores acordasen los criterios de evaluación, quizás de forma parecida a lo que se hace en [4].

do que era la pregunta clave que motivó la puesta en marcha de la experiencia), y que justificaran su decisión. La última parte de la plantilla era de carácter confidencial, como información útil al profesor pero que no se quería hacer llegar a los estudiantes, y que se refería a: (1) cómo consideraban que había sido la dedicación/esfuerzo del equipo en relación con el tiempo disponible para realizarlo (unos 4 meses); (2) qué calificación pondrían (desde «Suspendo» hasta «Matrícula de Honor») si tuviesen que ponerles nota; y (3) otras observaciones o comentarios que juzgasen oportunos.

Al subir su trabajo en la web de la asignatura (también basada en Moodle), cada equipo de estudiantes que se considerara merecedor del premio al mejor proyecto debía justificarlo brevemente. De los 8 equipos que presentaron el trabajo, todos menos uno presentaron su candidatura. En general, pensamos que sus argumentos resultan poco convincentes, pues se basan en afirmar que habían dedicado mucho esfuerzo, sin mostrar evidencias cuantitativas o cualitativas de tal dedicación, y tienden a magnificar su labor cuando en realidad se trata del trabajo mínimo que cabe esperar de un trabajo de estas características que supone el 70% de la nota y se desarrolla en equipo. Por otro lado, suelen recurrir a usar algunas frases estereotipadas en las que enfatizan su vocación por la usabilidad, sin que apoyen tales afirmaciones en resultados de su trabajo, tangibles y contrastables. De hecho, se ha visto que la valoración subjetiva (e incluso objetiva) que el estudiante hace de su esfuerzo es (bastante) menor de lo que cabe esperarse de ellos [6]. Uno de los mejores ejemplos de candidaturas (y aún así, débil), rezaba así:

«En nuestro trabajo hemos aplicado de forma práctica los diferentes conceptos tratados en la asignatura, con el objetivo final de conseguir una interfaz usable y cercana al usuario. Cabe decir que no nos hemos limitado únicamente a seguir los pasos indicados en la guía de elaboración del proyecto, si no que hemos afrontado el trabajo con una actitud activa que nos ha llevado a observar y entrevistar a usuarios para comprender sus necesidades y a hacer pruebas con ellos, cosa que creemos que ha influido positivamente en nuestro resultado final. Debido a esto y al esfuerzo que nos ha supuesto, reflejado en la cantidad de horas que le hemos dedicado, creemos que nuestro trabajo merece optar al premio.»

Transcurrida la fecha tope de entrega de los trabajos (6 de junio), los PDFs de una versión anónima de los informes se publicaron en la web de evaluación, sólo visibles para los revisores mediante contraseña. La asignación de proyectos a revisores buscó que cada trabajo tuviera tres revisores. Además, para evitar que algún trabajo terminase contando sólo con una revisión, la relación de proyectos que se envió a cada revisor estaba ordenada (priorizada), de modo que si por falta de tiempo habían de dejar de evaluar algún trabajo, que éste fuese el menos prioritario.

En todo momento se intentó simplificar la labor de los revisores (Punto 4, Fig. 1), enviándoles mensajes de correo electrónico en los que se les recuerda la URL de la web de evaluación, y la contraseña de acceso, recordándoles el plazo en que se esperaba su respuesta con una semana de antelación, etc. En general, los revisores cumplieron con este plazo, si bien algunos necesitaron algo más de insistencia por nuestra parte o se demoraron un poco más. Todos evaluaron los tres trabajos asignados, excepto uno que sólo pudo evaluar uno.

Los profesores de la asignatura revisaron los trabajos de los estudiantes *antes* de recibir o ver los informes de los revisores, para evitar que sus opiniones les influyeran. La valoración que los profesores hicieron de los informes de los expertos fue tanto cualitativa, a partir de los comentarios de los informes de evaluación, como cuantitativa, asignando valores numéricos a las preguntas con respuestas cerradas entre las que los revisores tenían que elegir (Cuadro 1). Para basar la decisión en resultados lo más objetivos posible, se tomaron los datos numéricos de los revisores y de los profesores, haciendo la media entre los 4 valores (3 de los revisores y 1 del profesor) para cada uno de 5 criterios.

A final de julio se envió a cada equipo los 2–3 informes de evaluación de los revisores, excluyendo la parte confidencial. Seguidamente se plantearon tres sencillas consultas (cuyos enunciados y respuestas se muestran en el Apartado 3.2) en la página web de la asignatura (también basada en Moodle), para conocer la opinión de los estudiantes.

Cuadro 1: Criterios de evaluación y posibles valoraciones (sus claves se usan en el Cuadro 2)

Impresión general	Usabilidad/HCI/Informe	Nota cualitativa	Clave
–	Excelente	Matrícula de honor	6
–	Muy bueno/a	Excelente	5
Tiene una gran calidad	Bueno/a	Notable alto	4
Está bastante bien	Regular	Notable ajustado	3
Es un trabajo algo ajustado	Malo/a	Aprobado	2
Está algo flojo	Muy malo/a	Aprobado ajustado	1
Es un trabajo bastante pobre	Pésimo/a	Suspense	0
–	–	No puedo valorarlo	NS/NC

3. Resultados y valoración

Las puntuaciones de los revisores y del profesor se recogen en el Cuadro 2. El Trabajo 7 fue el que no presentó candidatura y por eso no aparece. Uno de los aspectos que más nos interesaba al realizar esta experiencia, y que se relaciona con la idea de explorar la separación de roles, era conocer el grado de acuerdo entre los revisores entre sí y, especialmente, entre éstos y el profesor, en la valoración de los trabajos. Para cuantificar esto último de una forma compacta y global, se propone una medida sencilla, $D = \frac{1}{n} \sum_{c=1}^n (M_c - P_c)$, donde $n = 5$ es el número de criterios, M_c es la media de las puntuaciones de los revisores para el criterio c , y P_c es la puntuación del profesor para dicho criterio. De este modo, $D \approx 0$ refleja coincidencia entre opiniones, mientras que $D > 0$ ($D < 0$) indica puntuaciones mayores (menores) de los revisores que del profesor. Este índice D se muestra en la última columna del Cuadro 2. Como puede comprobarse, en general los revisores valoran mejor los trabajos que el profesor, con diferencias de hasta 2,3 puntos (Trabajo 1). En dos de los 7 trabajos (el 4 y el 8), la valoración sí es similar ($|D| < 1$), y en uno de ellos (el 4) la opinión del profesor es ligeramente más positiva ($D < 0$).

Al reflexionar sobre por qué se dan estas diferencias entre la valoraciones de profesores y expertos externos, un motivo bastante plausible es que los profesores tienen una información más detallada sobre la asignatura y unas expectativas mucho más concretas acerca de lo que les gustaría que hicieran los estudiantes en sus trabajos, y se desaniman si estos objetivos, sobre los que se insiste en la asignatura, no se reflejan en los trabajos de los estudiantes.

Los revisores, en cambio, a pesar de que se les informa para ponerles en la mejor situación posible, no parten de ningún “proyecto ideal” en mente, y pueden quedar más fácilmente sorprendidos por la novedad de unos trabajos que no han diseñado ni conocen en detalle. Otros dos factores importantes son, por un lado, el inevitable sesgo que el profesor tiene por conocer a los estudiantes y la supervisión realizada de sus proyectos y, por otro lado, la inercia de estar valorando trabajos de muy similares características durante varios años. Todas estas reflexiones son interesantes y particularmente relevantes en el contexto de la separación de roles. Entre otras consideraciones, se hace patente la necesidad de compartir y aplicar unos criterios de evaluación bien definidos.

Es posible analizar otros aspectos a partir del Cuadro 2. Por ejemplo, la mayor nota global G la obtiene el Trabajo 2, que fue, de hecho, al que se le concedió el premio. Para tal decisión se consideraron, además, los comentarios cualitativos de los revisores, y la diferencia de votos a favor y en contra de recibir el premio que cada trabajo había recibido de los revisores.

3.1. Opiniones de los revisores

En general, los revisores mostraron una actitud positiva durante todo el proceso. Algunos incluso se ofrecieron a participar en tareas similares en un futuro. Otros mostraron de modo especial su agrado e interés por esta iniciativa y varios se interesaron por conocer su desarrollo y los resultados. He aquí un par de comentarios personales de los revisores que ilustran estas opiniones e inquietudes.

« [...] ha sido todo un placer participar en la corrección de estos trabajos [...] y puedes contar conmigo para rea-

Cuadro 2: Valoración de cada aspecto de cada trabajo por el profesor (P) y tres revisores distintos (R_A, R_B, R_C). La nota global G es la media de todos los valores e incluye otro criterio (la nota cualitativa asignada), que no consta aquí. La diferencia entre la valoración de los revisores y el profesor se refleja en la medida D (ver texto)

Trabajo (Equipo)	Criterio																G	D
	Impresión general				Usabilidad				HCI				Informe					
	P	R_A	R_B	R_C	P	R_A	R_B	R_C	P	R_A	R_B	R_C	P	R_A	R_B	R_C		
1	1	3	4	3	3	5	5	4	3	4	5	5	2	4	5	5	3,8	2,3
2	3	4	4	–	4	6	6	–	4	5	6	–	5	6	5	–	4,9	1,3
3	2	2	2	1	3	4	4	3	2	4	3	3	3	2	4	4	2,8	1,0
4	3	2	2	4	4	4	3	5	4	4	4	5	4	3	4	5	3,7	–0,1
5	1	3	3	–	3	5	4	–	3	5	3	–	2	5	3	–	3,3	1,7
6	3	3	3	4	3	5	4	5	3	4	5	6	3	3	4	5	3,9	1,4
8	3	4	2	3	3	5	3	4	4	4	3	4	3	5	3	5	3,5	0,4

lizar cualquier otro tipo de tarea similar.» (Un revisor)

«Sí me gustaría saber cómo fue todo el experimento, qué equipo quedó ganador y si mis valoraciones encajaron con las de otros revisores o fueron valoradas positivamente por los alumnos y el equipo docente. Estaría muy bien si tuvierais una memoria de todo ello pues creo que es una muy buena iniciativa a implementar en otras enseñanzas.» (Otro revisor)

Éstas y otras observaciones y comentarios positivos fueron, en parte, una importante motivación para preparar este artículo, con el fin de dar a conocer la iniciativa a una audiencia mayor y potencialmente interesada.

3.2. Opiniones de los estudiantes

Como se comentó en la Sección 2, después de haber enviado a cada equipo los informes de evaluación de todos los revisores, se les plantearon tres sencillas preguntas como consultas en Moodle. Si se ha leído los informes de los revisores, contestar a las consultas podía suponer un insignificante esfuerzo de, pongamos, 10 minutos como mucho. A pesar de esta simplicidad, de avisarles que sus respuestas se trataran con absoluta discreción, y de insistirles varias veces, antes y después del verano, de lo importante que resulta que participen en este tipo de consultas, sólo 6 estudiantes (de 24) respondieron.

Los enunciados de las preguntas y las opiniones entre las que había que elegir se muestran en la Fig. 3. Las respuestas de los estudiantes se resumen en el Cuadro 3. Aunque condicionadas por la esca-

Cuadro 3: Cantidad de estudiantes que eligieron cada respuesta para las tres preguntas sobre «acuerdo», «interés» y «utilidad» (ver Fig. 3)

Respuesta	Pregunta sobre...		
	Acuerdo	Interés	Utilidad
a	2	6	2
b	2	0	3
c	2	0	1
d	0	–	–

sa participación, cabe realizar algunas observaciones. Por un lado, se constata una clara unanimidad en cuanto al interés (segunda consulta) de las opiniones de los revisores. Por otro lado, hay diversidad de opiniones en cuanto al acuerdo (primera pregunta) entre la opinión que el estudiante tiene sobre su trabajo y la de los revisores. Resulta muy interesante comprobar que 2 de los 6 estudiantes piensen que la valoración de los revisores está *por encima* de lo que ellos mismos consideran que su trabajo merece. Análogamente, *ningún* estudiante piensa que los revisores han infravalorado su trabajo. En definitiva, la revisión de los expertos les parece razonable o incluso optimista, lo que de algún modo coincide con la diferente valoración de revisores y profesores, constatada cuantitativamente en el índice D discutido anteriormente (Cuadro 2). Por último, también hay bastante dispersión en la utilidad (tercera pregunta) que ven a la realimentación recibida de los expertos: unos la encuentran provechosa y otros no.

Consulta sobre ACUERDO:

Señala la respuesta que más se acerque a tu opinión respecto al grado de ACUERDO con la valoración de los revisores. Recuerda que puedes matizar o ampliar tu respuesta, escribiendo al profesor o en el foro.

Respuestas:

- La opinión de los revisores coincide bastante con la que yo mismo tengo del trabajo de mi equipo
- Los revisores están acertados en algunas de sus apreciaciones, pero no en otras
- Los revisores han valorado el trabajo algo/mucho MEJOR de lo que desde mi punto de vista merece objetivamente
- Los revisores han valorado el trabajo algo/mucho PEOR de lo que desde mi punto de vista merece objetivamente

Consulta sobre INTERES:

Señala la respuesta que más se acerque a tu opinión respecto al INTERÉS de contar con la valoración de revisores externos. Recuerda que [...]

Respuestas:

- Me parece muy interesante y oportuno poder contar con la opinión de revisores externos, independientemente o además de la valoración del profesor
- La valoración de los revisores externos puede estar bien, pero no aporta mucho respecto a lo ya sé o la realimentación que recibimos del profesor
- No tiene ningún interés contar con las valoraciones de otros expertos; creo que se podrían evitar ese esfuerzo

Consulta sobre UTILIDAD:

Señala la respuesta que más se acerque a tu opinión respecto al grado de UTILIDAD de los comentarios de los revisores. Recuerda que [...]

Respuestas:

- He encontrado opiniones, sugerencias, recomendaciones o críticas muy apropiadas y que me parecen bastante útiles o provechosas en algún sentido
- Hay algunos comentarios pertinentes, pero en general no me parece que la valoración me sea de gran provecho. Quizás echo en falta detalles más concretos o mejor justificación en algunos apartados.
- Encuentro poco relevante o inútil todo o casi todo lo que dicen los revisores sobre nuestro trabajo.

3.3. Valoración de los profesores

Respecto a la opinión de los profesores de las opiniones de los estudiantes, ya se ha indicado, por un lado, la debilidad de las propias candidaturas al premio y, por otro, la decepción por la escasa participación en las consultas posteriores al envío de los informes de los estudiantes.

En cuanto a los informes de los revisores, se observaron ligeras discrepancias en las valoraciones del mismo trabajo tanto entre los diferentes revisores como entre los revisores y los profesores. En general, se ha apreciado una actitud bastante «benevolente» en los revisores. En lo que respecta a la valoración de si un trabajo merece el premio al mejor trabajo, también se detectaron ciertas discrepancias o dificultades. Así, revisores diferentes consideran el mismo trabajo como que *sí* y como que *no* es merecedor de tal distinción. Parte de la dificultad, en este caso, estriba en que cada revisor sólo evalúa en detalle 2–3 trabajos, por lo que no tienen la visión global necesaria para poder emitir esta opinión con mayor fundamento. Esta dificultad ya se anticipó en la fase informativa y, con el fin de disminuir sus efectos, se les sugirió a los revisores que también ojeasen los informes de los demás trabajos, para obtener esa conveniente visión más global.

Se aprecia una desigual exigencia de un mismo revisor respecto de diferentes trabajos revisados, aún siendo éstos de calidades similares. Parecen ser muy críticos en unos trabajos y poco en otros, lo que podría deberse a que los revisaron en momentos o incluso días diferentes (puede que bajo diferentes estados emocionales). Además se comprobó que algún revisor propone más de un trabajo como candidato al premio, lo cual resulta algo contradictorio con la existencia de un único premio. Otras muestras de inconsistencias o dificultades en la valoración de propuestas al mejor trabajo son: (1) un trabajo propuesto para premio tiene, en otros criterios, unas notas similares a trabajos no propuestos; (2) la justificación de por qué un trabajo merece el premio es débil.

Como curiosidad, todos los revisores se muestran en condiciones de valorar con una nota (Aprobado, Notable,...) los trabajos, incluso aquellos que no se consideran capacitados para valorar otros aspectos tales como si merece el premio, o si el esfuerzo/tiempo dedicado es adecuado, etc. Otro aspecto de interés es que incluso a estos expertos parece pa-

Figura 3: Tres consultas para conocer la opinión de los estudiantes sobre la experiencia (ver respuestas en el Cuadro 3)

sárseles por alto errores conceptuales o metodológicos relacionados con HCI. Por ejemplo, en un trabajo los estudiantes hablan del recorrido cognitivo como una técnica de evaluación *con* usuarios, y a un revisor no sólo no le resulta extraño, sino que (se) pregunta cuántos usuarios participaron en la evaluación de la interfaz. Otro ejemplo: a los revisores les parece bien que los estudiantes propongan una priorización de requisitos funcionales —algo a lo que se presta bastante atención durante la asignatura—, pero no cuestionan para qué lo hacen o si dicha priorización es consistente con el resto del trabajo, aspecto que los estudiantes suelen descuidar. La muestra de revisores y los datos no son suficientes para analizar si el perfil del revisor (docente o empresarial) influye en su forma de revisar.

En cuanto a la experiencia en general, ésta resultó bastante laboriosa, pero también interesante. Aunque nos gustaría concluir que la combinación de la competición *inter*-equipos y la colaboración *intra*-equipo resultó beneficiosa, no pareció que fuese así. De hecho, respecto a cursos anteriores, no se ha percibido una mayor calidad en los trabajos, lo que puede deberse, en parte, a que los estudiantes vieron el premio más como algo a lo que podían optar que como un concurso en el que les motivara competir por ganar. El modesto importe del premio (400 euros) podría explicar, en parte, esta baja motivación. De hecho, los profesores insistimos más en la importancia del certificado asociado al premio y de su valor en sus currícula. También es cierto que no se enfatizó mucho el tema del premio a lo largo del semestre, posiblemente porque, como otros profesores [1], cuestionamos la conveniencia de la motivación *extrínseca*. El hecho de tratarse de una asignatura obligatoria y, al parecer, poco atractiva a los estudiantes, podría explicar que, ni con incentivos y premios, les interese mucho. Además, los premios y otro tipo de iniciativas parecen funcionar mejor en asignaturas optativas [3].

4. Conclusiones

El aprendizaje por proyectos es una herramienta educativa con gran potencial, pero no exenta de dificultades. En nuestra experiencia, un importante reto es lograr que los estudiantes apuesten por un esfuerzo intenso y un trabajo de calidad. Una posibilidad

es ofrecer un premio al mejor trabajo y, en este contexto, la participación de expertos, externos a la asignatura, resulta muy útil, no sólo para ayudar a fallar el premio, sino por la realimentación que ofrece a estudiantes y profesores. Además, la experiencia permite explorar la separación de dos roles del profesor hasta cierto punto antagónicos, y que puede aportar beneficios educativos al ejercer con mayor naturalidad su papel de guía más que el de sabio.

Quedan pendientes interesantes preguntas acerca de cómo mejorar la realimentación de los revisores, o cómo lograr mayor implicación y motivación de los estudiantes. Algunas ideas a explorar serían liberar del examen al equipo ganador, concederle la máxima puntuación, o hacer partícipes a los propios estudiantes en la valoración de los trabajos y, por tanto, en la decisión del premio [2]. Un reto mayor es diseñar mecanismos para potenciar la motivación intrínseca y disminuir o eliminar la extrínseca.

Agradecimientos. A los estudiantes y los revisores de sus trabajos. A los revisores de este artículo. A la financiación del departamento, de la UJI y de la Unitat de Suport Educatiu (mediante sus proyectos de mejora educativa).

Referencias

- [1] K. Bain. *What the best college teachers do*. Harvard University Press, 2004.
- [2] A. C. Bovik. What you see is what you learn. *Signal Processing Magazine, IEEE*, 27(5):117–123, Sept. 2010.
- [3] J. Ibáñez, J. Gutiérrez, J. A. Elorriaga, and A. Goñi. El congreso de alumnos como recurso didáctico. In J. M. Julià, editor, *JENUI 2001*, pages 126–131, July 2001.
- [4] D. M. Oh, J. M. Kim, R. E. Garcia, and B. L. Krilowicz. Valid and reliable authentic assessment of culminating student performance in the biomedical sciences. *Adv. Physiol. Educ.*, 29:83–93, 2005.
- [5] M. R. Prensky. *Teaching Digital Natives: Partnering for Real Learning*. Corwin Press, 2010.
- [6] V. J. Traver and J. M. Pérez. Dedicación de los estudiantes en un contexto de aprendizaje cooperativo basado en proyectos: medición, análisis e implicaciones. *RITA*, 4(2):117–128, May 2009.

Auto-evaluación discente como estrategia de aprendizaje ¿Cuestión de género?

Carlos T. Calafate, Juan Carlos Cano, Pietro Manzoni

Dpto. de Informática de Sistemas y Computadores (DISCA)
Escuela Técnica Superior de Ingeniería Informática (ETSINF)
Universitat Politècnica de València (UPV)
E-mail: {calafate, jucano, pmanzoni}@disca.upv.

Resumen

La adopción de las directrices definidas para el Espacio Europeo de Educación Superior ha representado un gran cambio en el paradigma enseñanza/aprendizaje. En particular, los estudiantes deben adaptarse a las actividades y técnicas de estudio basadas en grupo. Además, se espera de ellos que gradualmente sean protagonistas de su propio aprendizaje. En el escenario actual, donde el acceso al conocimiento es un fenómeno generalizado, los profesionales deben ser capaces de mantener sus conocimientos actualizados a medida que se producen continuos avances en sus campos profesionales. En este escenario el auto-aprendizaje, más que deseable, se está convirtiendo en un requisito indispensable. Formar estudiantes con capacidades de auto-aprendizaje es un objetivo transversal en los actuales *curricula* de cualquier grado universitario que desea formar profesionales con éxito profesional. Para conseguir dicha competencia es fundamental que los estudiantes adquieran hábitos de auto-evaluación de su propio proceso formativo. En este trabajo se analiza si el género (masculino, femenino) constituye un factor clave que pueda afectar a las habilidades de auto-evaluación de nuestros estudiantes. También se analiza si las preferencias sociales relacionadas con el aprendizaje dependen del género de los estudiantes, y como éstas pueden afectar al proceso de la auto-evaluación.

1. Introducción

La utilización de la evaluación como herramienta metodológica para el aprendizaje [1] constituye un área que encuentra aplicación en diferentes ámbitos del escenario

universitario, incluyendo la evaluación de planes de estudio, los procesos de evaluación del personal, así como la evaluación de los estudiantes. La evaluación formativa [2] es un concepto relacionado, y se refiere al proceso mediante el cual el instructor mejora el proceso de aprendizaje de los alumnos en base a la información obtenida de los resultados de la evaluación.

La evaluación formativa puede centrarse en el estudiante, en el instructor, o en ambos. Si se hace hincapié en el estudiante, el objetivo principal estriba en permitirle mejorar su proceso de aprendizaje a partir de la retroalimentación de la evaluación. Si se hace hincapié en que el instructor, ésta le debería permitir determinar qué cambios podrían mejorar el proceso de enseñanza, por ejemplo la organización del curso, los contenidos del mismo, o los ejercicios y ejemplos propuestos. Por último, una tercera alternativa es centrarse tanto en el estudiante como en el instructor.

La auto-evaluación [3] está íntimamente relacionada con el aprendizaje autónomo. De hecho, para lograr un auto-aprendizaje eficiente y eficaz se requiere de una buena capacidad de auto-evaluación. Varios estudios indican, que los procesos de aprendizaje y evaluación presentan características diferentes cuando se analizan atendiendo al género de los estudiantes. Willingham y Cole [4] fueron de los primeros autores en estudiar la relación entre el proceso de evaluación y el género de los estudiantes. Autores como Halpern [5] afirman que hay indicios claros que indican que los hombres superan a las mujeres en las pruebas de capacidad cuantitativa y habilidades matemáticas. Otros autores, como Spelke [6] y Hyde [7], también han advertido de estas diferencias en función del género cuando se trata de titulaciones con marcado carácter tecnológico

y matemático, indicando, además, que la investigación sobre las capacidades cognitivas de hombres y mujeres, desde el nacimiento hasta la madurez, no es compatible con tales afirmaciones. Para demostrarlo, Hyde *et al.* han realizado un análisis estadístico más detallado que muestra que las diferencias de puntuación entre hombres y mujeres en términos de habilidades matemáticas son insignificantes.

Aunque los estudios anteriormente citados arrojan algo de luz en el desarrollo de las competencias científicas basadas en el género, la relación particular entre auto-evaluación y su dependencia del género es un tema que requiere un examen más detallado.

En los años noventa, Beyer [8] estudió la relación entre auto-evaluación y capacidades, concluyendo que existen diferencias significativas en función del género. En concreto, las mujeres tenían tendencia a subestimar significativamente su rendimiento y percibían una sensación de fracaso mucho más acentuada que los hombres. Recientemente, Pallier [9] ha realizado dos encuestas diferentes, con una población significativa de adultos, que una vez más confirma que, en general, los hombres expresan niveles más altos de confianza que las mujeres en el trabajo desarrollado y en sus capacidades.

En el presente trabajo, se analizan las habilidades de auto-evaluación de un grupo de estudiantes de Ingeniería Informática cursando un curso de Redes de Computadores.

A partir de una encuesta realizada a los estudiantes durante el curso 2009/2010, se obtuvieron resultados de una población de 70 estudiantes, que fueron clasificados y analizados en función del género. Además, también se estudió si el género afecta a las preferencias sociales de aprendizaje asociados con la evaluación formativa en el aula.

Los resultados obtenidos muestran que, a diferencia de trabajos anteriores, “**los estudiantes**” son las que expresan más altos niveles de confianza en su trabajo. Esta discrepancia nos permite formular la hipótesis de que tal vez las características de los estudiantes de Ingeniería Informática difieren de los comportamientos generales y las pautas de auto-evaluación detectados por otros investigadores en poblaciones diferentes.

El resto del trabajo se organiza como sigue. En la Sección 2 se revisan algunos trabajos relacionados con la evaluación formativa. La Sección 3 describe los detalles sobre la encuesta y la población utilizada. Los resultados de la encuesta se presentan y discuten en la Sección 5. Por último, la Sección 6 presenta las conclusiones del trabajo.

2. Trabajo Relacionado

La evaluación formativa es un proceso auto-reflexivo que persigue promover y mejorar los resultados de los estudiantes [10]. Cowie y Bell [11] lo definen como el proceso bidireccional entre el profesor y el estudiante para mejorar, reconocer y responder al aprendizaje. Black y William [12] consideran que la evaluación es “formativa” cuando la retroalimentación de las actividades de aprendizaje se utilizan efectivamente para adaptar la enseñanza a las necesidades del alumno. Nicol y Macfarlane [13] interpretan de forma diferente los conceptos de evaluación formativa y retroalimentación, mostrando como estos procesos pueden ayudar a que los estudiantes tomen las riendas de su propio aprendizaje (aprendizaje auto-regulado).

La auto-evaluación en un entorno educativo induce a los estudiantes a hacer juicios sobre su propio trabajo. Uno de los primeros experimentos en este ámbito [3] fue realizado por Sedikides (1993), que estudió los diferentes roles de la evaluación prestando especial atención a los aspectos relacionados con la auto-evaluación y la auto-verificación. Trope y Ben-Yair [14] también han estudiado la auto-evaluación, concluyendo que los individuos prefieren adquirir las competencias que les permitan medir sus habilidades con mayor precisión, aunque serán más persistentes evaluando aquellas tareas que requieran una menor precisión en dicha evaluación.

A diferencia de trabajos previos, nuestro trabajo se centra en la auto-evaluación como herramienta que potencia la evaluación formativa en el contexto de los estudios universitarios, utilizando una aproximación por género.

3. Detalles de la encuesta y población del estudio

Nuestro análisis se centra en los estudiantes matriculados en la asignatura Redes de Computadoras, asignatura obligatoria de 4º curso (séptimo semestre) de la titulación de Ingeniería Informática de la Universidad Politécnica de Valencia.

Un total de 70 estudiantes han participado en nuestro estudio (alrededor del 95% de los estudiantes que asisten regularmente a clase), de los cuales 55 son hombres y los 15 restantes son mujeres.

La encuesta constaba de dos partes: (i) una vez finalizado el examen final de la asignatura, y antes de entregar el examen, se solicitó a los alumnos que indicasen por escrito en el propio examen la nota esperada en cada una de las cuestiones, indicándoles para cada pregunta su peso sobre la nota final. (ii) A la semana siguiente del examen, y antes de publicar las notas de los mismos, se les pidió rellenar una encuesta con una amplia colección de preguntas relacionadas con la auto-evaluación y las preferencias sociales relacionadas con la auto-evaluación y el aprendizaje. Con toda esta información se procedió a realizar un análisis discriminando los resultados en función del género (masculino o femenino). A continuación se presentan y analizan los resultados obtenidos.

4. Análisis de resultados sobre las competencias de auto-evaluación

En esta sección se presentan los resultados más representativos de nuestro estudio. En primer lugar, se presentan resultados que hacen referencia a las competencias de auto-evaluación de los estudiantes, mostrando su nivel de conciencia sobre dicho proceso, es decir, la diferencia entre las puntuaciones estimadas y las que realmente han obtenido.

Se han comparado tres datos estadísticos: los resultados reales obtenidos en el examen (S_r), los valores estimados por los alumnos (S_e), y las puntuaciones que los estudiantes creen que merecen (S_d); éste último dato obtenido refleja las habilidades que los alumnos perciben que han adquirido en el contexto de los diferentes conceptos de la asignatura.

En cuanto a los resultados del aprendizaje, la Figura 1 muestra un diagrama de caja y bigotes para la distribución de resultados asociados a ambos géneros. Podemos observar que la puntuación media obtenida por los hombres es un 26% superior a la puntuación obtenida por los estudiantes de género femenino. También encontramos diferencias significativas entre las puntuaciones máximas y mínimas y, aunque el rango es similar para ambos géneros (alrededor del 63%) las puntuaciones obtenidas por las alumnas está ligeramente desplazada hacia la izquierda.

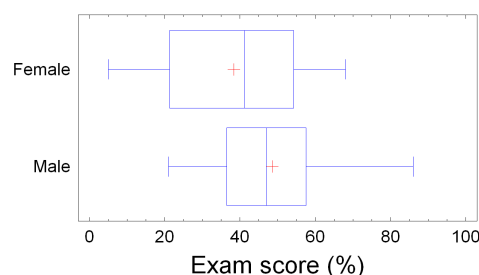


Figura 1. Diagrama de caja y bigotes para las puntuaciones de examen en función del género.

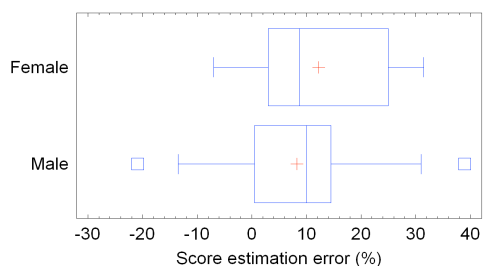


Figura 2. Diagrama de caja y bigotes de los errores en la estimación de la puntuación obtenida, en función del género.

Centrándonos ahora en los errores de estimación, la Figura 2 muestra el diagrama de caja y bigotes para la distribución de los errores de estimación asociada. Podemos observar que tanto chicos como chicas tienden a sobreestimar las puntuaciones obtenidas.

Al comparar la distribución por género, nos encontramos con que un mayor porcentaje de alumnas tienden a sobrestimar los resultados: 81% (mujeres) frente a 76% (hombres). Además, encontramos que alrededor del 25% de los hombres son pesimistas en dicha estimación, con errores de estimación de hasta un -22%. Las chicas, por su parte, son mucho menos pesimistas, con un error de estimación negativa de un -8% en el peor de los casos. En general, encontramos que las estudiantes no estiman con precisión su puntuación en un 12% como media, mientras que para los chicos este valor medio es ligeramente inferior (8%). Además, la distribución que corresponde a la población masculina tiene una gama mucho más amplia en comparación con la población femenina (61% vs 40%), mientras que también tiene una *curtosis* superior y un desplazamiento ligeramente negativo. Los resultados obtenidos presentan discrepancias con respecto a los estudios realizados por Beyer y Pallier [8,9], que indican que las mujeres son más propensas a subestimar su trabajo. Esto puede ser debido a las características de aprendizaje de los estudiantes de ingeniería.

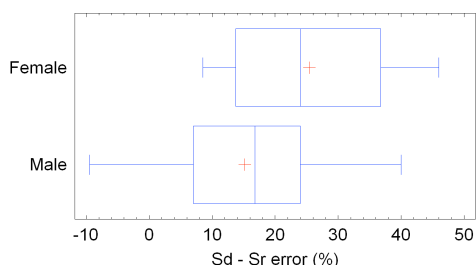


Figura 3. Diagrama de caja y bigotes de los errores en la puntuación que los alumnos creen merecer, en función del género.

Para concluir nuestro análisis de las habilidades de auto-evaluación, analizamos ahora el error entre la nota finalmente obtenida y la que los estudiantes creen que merecen en función de su percepción de conocimiento adquirido a lo largo del curso. Los resultados obtenidos se muestran en la Figura. 3. Las diferencias observadas son bastante acentuadas en función del género. El valor medio del error de estimación es del 24%

para la población femenina y del 15% para la masculina. En general, observamos que ninguna chica es pesimista respecto a la puntuación que cree merecer, siendo la mayoría bastante optimistas (error superior al 10%) e incluso algunas (15%) son extremadamente optimistas, con un error superior al 30%.

Con respecto a la población masculina, nos encontramos con que, aunque también hay un alto porcentaje optimista, ahora también encontramos alrededor de un 30% de estudiantes que podemos considerar realistas (error en el intervalo [-10%, 10%]).

En este punto se hacen evidentes los diferentes puntos de vista entre el profesor y el estudiante con respecto a la evaluación de competencias. Así, los resultados parecen indicar que los estudiantes no identifican el examen realizado como una herramienta de evaluación de los conocimientos adquiridos. Sin embargo, si analizamos las respuestas de los estudiantes a las preguntas relativas a dicho aspecto (ver Figura 4) verificamos que sólo el 10% de los estudiantes piensan que el examen no evalúa sus conocimientos adecuadamente, mientras que una amplia mayoría sí que está de acuerdo sobre la adecuación del examen. No hubo diferencias significativas entre ambos géneros sobre esta cuestión. Así pues, todo parece indicar que los resultados se deben por una parte a la falta de competencias de los estudiantes con respecto a la auto-evaluación de sus propios conocimientos y, por otra, quizás a no haber sabido transmitir a los alumnos desde un principio los objetivos y competencias que se espera que éstos adquieran en la asignatura. Nos planteamos el reto de cómo transmitir a los estudiantes lo que se espera de ellos mejorando, al mismo tiempo, sus competencias de auto-evaluación.

Según diferentes autores [15,16], la utilización de rúbricas pueden ser de gran utilidad para lograr este objetivo, ya que los estudiantes se pueden beneficiar de un conjunto de normas claramente definidas desde el principio, que son de utilidad para centrar su atención en los aspectos realmente importantes, mejorando al simultáneamente sus capacidades de auto-evaluación. Como trabajo futuro para el próximo curso académico tenemos previsto implantar un sistema de evaluación basado en rúbricas de calificación con información detallada sobre qué

aspectos son los más representativos en cada una de las unidades, y cómo estos aspectos serán evaluados.

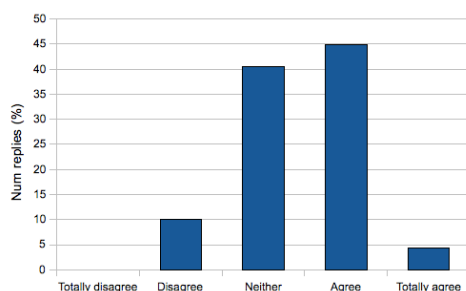


Figura 4. Histograma de la opinión sobre el examen realizado, y si éste ha sido representativo del conocimiento adquirido a lo largo del curso.

5. Análisis de los resultados de las preferencias sociales del aprendizaje

En la sección anterior hemos demostrado que, en general, las habilidades de auto-evaluación del estudiante son bastante pobres. En estas circunstancias, la retroalimentación del instructor es un requisito esencial para una evaluación formativa más precisa, que permita además que los estudiantes sean más conscientes del proceso de adquisición de conocimientos.

La retroalimentación por parte del instructor acerca de la evaluación puede tener lugar en cualquier momento del proceso de aprendizaje. Así pues, en la segunda parte de nuestro estudio, se analizarán las preferencias sociales de los estudiantes con respecto a dicha retroalimentación en los procesos de aprendizaje. Nos hemos centrado en las siguientes cuestiones:

1. Describir el grado de incomodidad al recibir comentarios de evaluación en clase, es decir, cuando se evalúa frente a otros estudiantes.
2. Describir el grado de incomodidad cuando la retroalimentación se produce en horas de tutorías, es decir, cuando se evalúa a solas con el profesor.

3. Opinión con respecto a la siguiente afirmación: "la retroalimentación por parte del instructor mejora mi capacidad de auto-evaluación".

La Figura 5 presenta los resultados de la primera pregunta en función del género. Podemos observar que la mayoría de las estudiantes (63%) se sienten cómodas cuando son evaluadas en clase, mientras que la mayoría de los estudiantes (67%) se sienten moderadamente (44%) o muy incómodos (23%) al ser evaluados frente a otros estudiantes.

Cuando los alumnos asisten a horarios de tutorías del instructor para solucionar dudas, las respuestas son muy diferentes. Ahora nos encontramos con que la inmensa mayoría de estudiantes de género femenino (74%) se sienten moderadamente incómodas, mientras que la mayoría de los estudiantes se sienten más cómodos en este escenario (58%). Al analizar estos resultados podemos intuir que los resultados son, en parte, debidos a que todos los profesores de la asignatura son hombres. Otra cuestión digna de mencionar es que, particularmente en la población masculina, parece que un número significativo de estudiantes no está satisfecho con cualquiera que sea la estrategia de retroalimentación utilizada por el profesor.

Para completar nuestro análisis, hemos analizado la opinión de los estudiantes sobre la importancia de la retroalimentación del instructor como un mecanismo para mejorar la evaluación del aprendizaje. El propósito es confirmar si el punto de vista donde el instructor juega un papel central en la mejora de habilidades de aprendizaje de evaluación, que se adopta en este y otros trabajos, es una opinión compartida por los estudiantes o si, por lo contrario, no están de acuerdo con dicha opinión.

Los resultados mostrados en el histograma de la Figura 7 confirman que la mayoría de los estudiantes tienen una opinión positiva acerca de la retroalimentación del instructor, considerándola beneficiosa para su aprendizaje. Sin embargo, alrededor de un 11,6% de los estudiantes no están de acuerdo con esta opinión, lo que indica que no asocian la tarea del instructor con la tarea de promover la adquisición de conocimientos de estudiante,

guiándolo durante todo el proceso a través de diferentes mecanismos de evaluación.

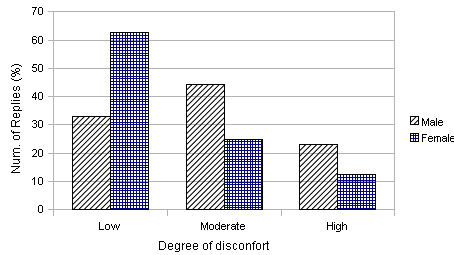


Figura 5. Histograma que describe el grado de incomodidad del estudiante (en función del género) al ser evaluados en clase.

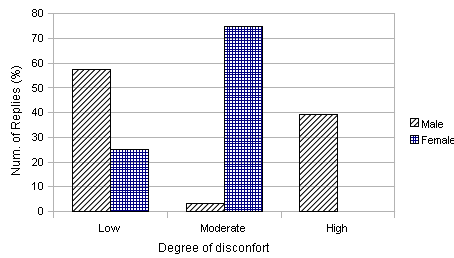


Figura 6. Histograma que describe el grado de incomodidad del estudiante (en función del género) al ser evaluados durante las horas de tutoría.

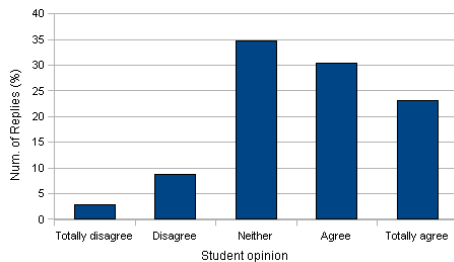


Figura 7. Histograma acerca de la importancia de la retroalimentación del instructor como un mecanismo para mejorar sus capacidades de auto.

6. Conclusiones

En este trabajo hemos analizado las capacidades de auto-evaluación de los alumnos como herramienta de aprendizaje, en el ámbito de un

curso de Redes de Computadores de una titulación de Ingeniería Informática. Nos hemos centrado en las habilidades de auto-evaluación de nuestros estudiantes, así como de sus preferencias sociales relacionadas con la retroalimentación obtenida por parte del instructor. A partir de los datos recibidos de una población de 70 estudiantes de Ingeniería Informática, los resultados muestran que, aunque todos los estudiantes tienden a sobrestimar su calificación del examen, las estudiantes son mucho más optimistas (una diferencia de puntuación de 10% más que los chicos en comparación con el resultado real obtenido). Además, algunos estudiantes (masculinos) creen merecer una nota que está por debajo del resultado real obtenido (hasta un 10% menos). Entre las estudiantes de género femenino no encontramos casos pesimistas en cuanto a la nota que creen merecer.

En general, el estudio realizado revela que las mujeres son más propensas a sobrestimar la calidad de su trabajo. Aunque este resultado está en clara contradicción con trabajos anteriores, consideramos que puede ser debido a que la población de estudiantes de Ingeniería Informática tiene características especiales que la hacen diferente respecto a patrones encontrados en muestras de otras poblaciones.

En cuanto a las preferencias sociales, encontramos que las estudiantes se sienten más cómodas que los varones cuando se está evaluando en clase (63% vs 33%). Sin embargo, hay un grupo significativo de estudiantes que, independientemente de su sexo, se sienten muy incómodo siendo evaluados frente a otros estudiantes (23% hombres y 12% de estudiantes mujeres). Cuando se analizan los resultados de la evaluación realizada en horarios de tutorías, la tendencia es la opuesta: el 57% de los estudiantes masculinos se sienten cómodos, mientras que este porcentaje se reduce al 25% en el caso de las alumnas. Además, alrededor de un 11,6% de los estudiantes tienen una opinión negativa acerca de la retroalimentación del instructor, aspecto que está en clara contradicción con la interacción natural estudiante-profesor de base. Este factor merece un estudio más detallado.

Como trabajo futuro, planeamos realizar estudios similares en próximos cursos que

permitan validar estos primeros resultados. Además, tenemos la intención de evaluar el impacto de la utilización de rúbricas en el proceso de enseñanza-aprendizaje que permita a los estudiantes tener una visión más objetiva de aquellos aspectos representativos, ofreciéndoles además una guía detallada de cómo dichos aspectos serán posteriormente evaluados.

Agradecimientos

Este trabajo ha sido parcialmente financiado por la Escuela Técnica Superior de Ingeniería Informática (ETSINF - UPV), y el Departamento de Informática de Sistemas y Computadores (DISCA) de la Universitat Politècnica de València, España.

Referencias

- [1] D. Carless, "Prospects for the implementation of assessment for learning". In *Assessment in Education: Principles, Policy and Practice*. Routledge: Taylor & Francis Group, 2005, Vol. 12(1), pp. 39-54.
- [2] P. Black and D. William, "Inside the Black Box: raising standards through classroom assessment". London: King's College, Reprinted in *Phi Delta Kappan*, 1998, Vol. 80(2), pp. 139-48.
- [3] C. Sedikides, "Assessment, enhancement, and verification determinants of the self-evaluation process". *Journal of Personality and Social Psychology*, 1993, Vol. 65(2), pp. 317-338.
- [4] W.W. Willingham, N.S. Cole, *Gender and fair assessment*, Mahwah, NJ: Erlbaum, 1997.
- [5] D. Halpern, *Sex Differences in Cognitive Abilities*, Lawrence Erlbaum Associates, 2000.
- [6] E. Spelke, *Sex Differences in Intrinsic Aptitude for Mathematics and Science?: A Critical Review*, *American Psychologist*, Vol. 60(9), pp. 950-958, 2005.
- [7] J.S. Hyde, E. Fennema, S.J. Lamon, *Gender differences in mathematics performance: a meta-analysis*. *Psychol Bull.*, Vol. 107(2), pp. 139-55, 1990.
- [8] Sylvia Beyer, *Self-Consistency and Gender Differences in the Accuracy of Self-Evaluations*, 100th Annual Convention of the American Psychological Association, Washington, DC, 14-18 August, 1992.
- [9] Gerry Pallier, *Gender Differences in the Self-Assessment of Accuracy on Cognitive Tasks, Sex Roles*, Vol. 48(5/6), pp. 265-276, 2003.
- [10] T. Crooks, *The Validity of Formative Assessments*, Paper presented to the British Educational Research Association Annual Conference, University of Leeds, 13-15 September 2001.
- [11] B. Cowie, B. Bell, *A model of formative assessment in science education*, *Assessment in Education*, 1999, Vol. 6, pp. 101-116.
- [12] P. Black, D. William, *Inside the black box: Raising standards through classroom assessment*. *Phi Delta Kappan*, 1998, Vol. 80(2), pp. 139-149.
- [13] D.J. Nicol, D. Macfarlane-Dick, *Formative assessment and self-regulated learning: A model and seven principles of good feedback practice*. *Studies in Higher Education*, 2006, Vol. 31(2), pp.199-218.
- [14] Y. Trope, E. Ben-Yair, *Task construction and persistence as means for self-assessment of abilities*. *Journal of Personality and Social Psychology*, 1982, Vol. 42, pp. 637-645.
- [15] H. Goodrich, "Understanding Rubrics". *Educational Leadership*, 1996, Vol. 54(4), pp. 14-18.
- [16] H. Andrade, *Self-assessment through rubrics*. *Educational Leadership*, 2007, Vol. 65(4), pp. 60-63.
- [17] E. Wenger, "Artificial intelligence and tutoring systems: computational and cognitive approaches to the communication of knowledge". Morgan Kaufmann Publishers, 1987.

Práctica de desarrollo de interfaces hardware/software para el manejo de redes de sensores inalámbricos

Ángel Jiménez Fernández, Manuel Domínguez Morales,
Elena Cerezuela Escudero, Rafael Paz Vicente, Gabriel
Jiménez Moreno, Alejandro Linares Barranco

Dpto. de Arquitectura y Tecnología de Computadores

Universidad de Sevilla

Escuela Técnica Superior de Ingeniería Informática – Avda. Reina Mercedes s/n

41012 Sevilla

ajimenez@atc.us.es

José Ignacio Villar
de Ossorno

Dpto. de Tecnología
Electrónica

Resumen

En este artículo se describe una práctica de laboratorio en la que se ilustran los mecanismos necesarios para implementar redes de sensores inalámbricos y la comunicación hardware/software en un PC. En los últimos años se han popularizado las redes de sensores inalámbricos. En la actualidad podemos encontrarlos realizando una elevada diversidad de tareas, desde aplicaciones domóticas, médicas, o de monitorización del medio ambiente. Proponemos una práctica en la que cada alumno usará un dispositivo basado en la tecnología XBEE, y desarrollará un software en el PC que implementará una red inalámbrica con una topología de maya en el laboratorio. Ilustrando, de esta manera, los principios necesarios para el desarrollo de interfaces hardware/software para la comunicación con el modem, así como proporcionar al alumno los conocimientos necesarios el manejo y gestión del propio modem.

Summary

In this paper is described a lab session where is illustrated the mechanisms for implementing wireless sensor networks and hardware/software communication using a computer. In the last year wireless sensor networks have become very popular. Nowadays, we can find them performing a high diversity of tasks, from domotic applications, medical, or environment monitoring. We propose a lab session where each student will use a XBee-technology-based device and will develop a software in order to implement a wireless sensor network with a mesh topology in the laboratory. Thus, we can illustrate the principles needed to develop hardware/software

interfaces for modem communication, and also give students enough knowledge for driving and managing the modem.

Palabras clave

Aprendizaje basado en problemas, trabajo en equipo, aprendizaje colaborativo, redes de sensores inalámbricos, programación orientada a objetos.

1. Motivación

La adaptación del Espacio Europeo de Educación Superior y, en particular, la adopción del sistema de crédito europeo (ECTS) están determinando un cambio de modelo educativo. Evolucionamos desde un modelo centrado en la “enseñanza del profesor” a uno basado fundamentalmente en el “aprendizaje del alumno” [1]. Esto ha determinado que prácticamente todas las universidades hayan promovido planes de innovación docentes que faciliten la implantación de nuevas metodologías por parte del profesorado. En la Universidad de Sevilla, en el curso 2007, se puso en marcha el Plan de Renovación de las Metodologías Docentes 2007 [2]. Este plan estaba dividido en varias líneas promoviendo, entre otras cuestiones, la formación del profesorado en nuevas las metodologías docentes, implantación de dichas metodologías en las asignaturas, dotación de espacios docentes, etc.

Con la implantación de los nuevos grados, los profesores están ultimando los contenidos, metodologías y planificaciones de las nuevas asignaturas. En muchos casos nos hemos encontrado con la novedad de la carga docente no presencial por parte del alumno, esto nos ha

supuesto plantearnos nuevas cuestiones acerca de cómo implementar dicha carga. Estas cuestiones giran en torno a cómo elaborar actividades y tareas que ocupen esta carga por parte del alumno, así como su seguimiento y evaluación por parte del profesor. En definitiva, las universidades han invertido una elevada cantidad de recursos en formar al profesorado en nuevas metodologías didácticas y de aprendizaje, sin embargo, poco se ha invertido en la formación en nuevos contenidos. Esta deficiencia se agudiza en el ámbito de la ingeniería informática, y en particular en el ámbito de la arquitectura de computadores, sistemas empotrados, dispositivos móviles, etc... Ya que esta es una tecnología en continuo cambio y evolución, y además resulta crucial para el desarrollo tecnológico que los ingenieros informáticos posean conocimientos de la tecnología más actual posible, estando esta responsabilidad casi exclusivamente sobre el profesor. No se debe de perder de vista, que en las asignaturas impartimos contenidos para que los alumnos adquieran una serie de competencias, así que los proyectos del Aprendizaje Basado en Proyectos (ABP) han de estar muy enfocados a la adquisición de las competencias propias de la materia donde se imparte.

En el ámbito de las ingenierías, las cuales son ciencias aplicadas, opinamos que el uso de las metodologías del ABP resulta muy adecuado para su uso en las partes no presenciales de las asignaturas. Está tratado ampliamente en numerosos trabajos [3,4,5] y a pesar de que es una metodología relativamente nueva, ha demostrado ser eficaz en la enseñanza de las más diversas ramas de conocimiento [5]. Sin embargo, puede que no todo sea ventajoso, el uso del ABP en la carga no presencial de las asignaturas demanda un elevado nivel de motivación por parte del alumno, ya que es el propio alumno el que tiene que “obligarse” a participar en el ABP. Es decir, mientras que en las sesiones presenciales de prácticas el alumno está realizándola y el profesor controlándola, en el ABP, el alumno es libre de realizar las tareas que le correspondan donde y cuando quiera, de tal forma que la falta de motivación en una asignatura le llevará a invertir más tiempo en cuestiones ajenas a la asignatura. Para tratar de conseguir esa motivación extra, pensamos que se puede conseguir proponiendo

proyectos que les sean muy atractivos y colaborativos, de manera que se vean obligados a colaborar, que no trabajar, con otros alumnos en busca de un sistema o resultado común.

En este artículo exponemos uno de los proyectos que será propuesto a los alumnos de los nuevos grados en ingenierías informáticas y electrónicas, en diversas asignaturas. En este proyecto los alumnos tendrán que diseñar e implementar mayados de redes de sensores inalámbricos en el laboratorio. La novedad y la simpleza de estos sistemas los hace muy atractivos a los alumnos, y además es una actividad netamente colectiva, como la mayoría de las actividades relacionadas con las redes de información. La idea es proporcionar a cada alumno un modem inalámbrico, que exponemos más adelante, de manera que cada alumno lo conectará a un PC e implementará el software necesario para su manejo. Finalmente, mediante el material proporcionado por el profesor, desarrollarán un sistema de gestión del modem inalámbrico, controlado mediante una interfaz gráfica que les permite comunicarse con otros alumnos, así como poder analizar en tiempo real diversos parámetros de la red implementada.

2. El problema del material y los laboratorios

El problema surge cuando se pretende dar una docencia práctica que precisa cierto material específico: la solución menos costosa, y bastante eficaz, es usar simuladores. Pero si lo que queremos es que de verdad el alumno se instruya con un sistema real podemos proponer dos tipos de soluciones:

- Proporcionar al alumno el material necesario para que desarrolle los proyectos donde quiera.
- Dejar laboratorios habilitados como si fueran “salas de estudio” donde el alumno disponga libremente del material [6].

Normalmente la opción a elegir dependerá del coste final de cada una de ellas. En la mayoría de los casos se opta por la primera opción preferiblemente, porque si se elige la segunda, se

restringe en gran medida las posibilidades de los alumnos.

Actualmente, los sistemas inalámbricos han llegado a un nivel en su evolución que permite la adquisición de equipos de desarrollo y entrenamiento a un coste bastante reducido. En este sentido no parece descabellado pensar que el alumno pudiera asumir dicho coste, como se ha hecho siempre, por ejemplo, con el material de dibujo. Sin embargo, nuestro entorno no permite tal extremo, ya que los alumnos en informática están muy acostumbrados al acceso gratuito del material. Es por ello que hemos optado por la adquisición de un sistema que pueda ser prestado al alumno, de forma que su coste no repercuta directamente en el mismo y, además, pueda ser reutilizado y, por tanto, amortizado en varios cursos por parte de la Universidad.

Como en esta práctica se propone que los alumnos creen redes de sensores con una topología en maya, es necesario que cada alumno sea un "nodo" de la maya. De forma que están obligados a colaborar con otros estudiantes para elaborar la maya. En consecuencia, se necesita que varios alumnos, cada uno con su equipo de desarrollo, estén en el radio de acción de un modem para implementar la maya. Es decir, para el alumno el hecho de tener un equipo de desarrollo no es suficiente para el desarrollo de la práctica, sin embargo, sí que lo es para su desarrollo parcial.

La solución a esta problemática fue la creación de un espacio docente adecuado para el correcto desarrollo de la práctica, siendo éste un punto de encuentro de alumnos con prácticas similares. Gracias al I Plan Propio de Docencia de la Universidad de Sevilla en el curso 2008-2009 (línea 13: infraestructuras y equipamiento de espacios docentes) se adaptó un laboratorio para el desarrollo hardware para su libre uso por parte del alumnado. Este laboratorio se equipó, entre otras medidas de seguridad, con un control de acceso controlado por la tarjeta identificativa de los estudiantes, de manera que los alumnos pertenecientes a las asignaturas participantes de experiencias ABP están autorizados a su uso, pudiendo sólo acceder los alumnos adecuados. Además, este sistema nos permite hacer un seguimiento acerca del uso individual de cada

alumno del laboratorio. A diferencia de las prácticas simuladas y eminentemente software, en las que todo el material necesario es un ordenador, para el desarrollo de prácticas hardware suele ser necesario el uso de un material mucho más específico y costoso, como son fuentes de alimentación, multímetros, osciloscopios, generadores de señales, analizadores lógicos o de espectro. El alumno puede encontrar disponible todo este equipamiento y puede hacer un uso libre del mismo, acercando y familiarizando al alumno al equipamiento de laboratorio necesario para el desarrollo hardware que se encontrará en su vida laboral. Es muy importante destacar que el alumno no sólo desarrolla la práctica propuesta en el laboratorio, sino que en su propio desarrollo se instruye en el uso del equipamiento de test y medida existente. El desarrollo de sistemas hardware no sólo implica los conocimientos teóricos o fundamentales subyacentes, sino que implica un fuerte conocimiento de los elementos necesarios para la fabricación y desarrollo, no siendo esta implicación tan fuerte en los desarrollos software. Los alumnos están comprometidos a hacer un uso responsable del laboratorio, siendo advertidos, que un mal uso del equipamiento repercutirá en una calificación, ya que demuestran así su falta de conocimiento en la materia.

Finalmente, como se puede extraer de todo lo anteriormente expuesto, optamos por una solución híbrida. Por un lado los alumnos tienen un equipo de desarrollo a su disposición, que les permite el desarrollo de una gran parte de la práctica (comunicación software/hardware, gestión del modem,...); y, además, disponen de acceso a nuestro laboratorio, donde se encontrarán con alumnos en su misma situación y podrán colaborar en el desarrollo completo de la práctica (comunicación entre nodos, implementación de mayas,...).

3. Material proporcionado para la elaboración de la práctica

A continuación se expone el material proporcionado al alumno y las herramientas de la que dispone para la correcta realización de la práctica.

3.1. Modem inalámbrico: XBEE

El módulo XBee proporciona una interfaz serie con un enlace de radiofrecuencia. Dicho módulo puede convertir datos serie a señales de radiofrecuencia para, posteriormente, ser enviados a cualquier dispositivo en una red de RF. El módulo XBee-PRO 868[7], que es el utilizado en desarrollo de esta actividad, fue diseñado para cubrir necesidades de bajo coste, baja potencia y, como se ha nombrado con anterioridad, la creación de redes de sensores inalámbricos. Los módulos requieren poca potencia y proporcionan un flujo fiable de datos entre dispositivos remotos. Funcionan en la banda ISM de 868 MHz de frecuencia. Se puede apreciar un módulo XBee en la figura 1.

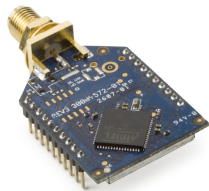


Figura 1. Fotografía de un modem XBEE.

Se trata de un módulo integrado e inteligente, mediante el cual es posible operar en redes 802.15.4 (base de Zigbee). Estos módulos se presentan como una interesante y atractiva alternativa, acortando el tiempo de desarrollo, dado que ya poseen una pila de protocolos configurada y operativa, funcionando como modems configurables mediante el set de comandos AT y pudiendo operar en un modo transparente, lo que permite (por ejemplo) reemplazar un cable.

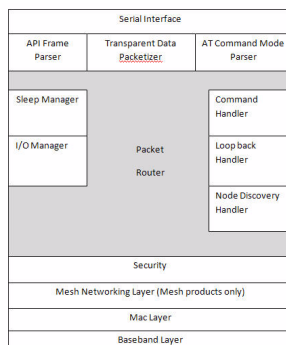


Figura 2. Capas de protocolo implementadas en el interior de un modem XBEE.

Presentan una potencia de salida de 315mW (25dBm), y la sensibilidad del receptor es de -112 dBm. Esto le permite operar a más de 10km en espacios abiertos, y hasta 550m en espacios urbanos. La corriente de operación de estos dispositivos ronda los 200mA, tanto para transmisión (500mA a máxima potencia para el XBee-PRO 802.15.4) como para recepción (65mA), mientras que en el modo de bajo consumo se reduce a tan sólo 10uA. La tensión de operación es 3 a 3,6V. Las opciones de antena son: antena chip integrada en el módulo, antena de cuarto de onda en el módulo, conector UFL o RPSMA para antena externa al módulo, aunque solo se comercializan las dos primeras opciones.

Los módulos permiten configurar muchos de los pines, destinados por defecto como I/O, para funcionar como entradas analógicas. Existen además dos salidas PWM, una de las cuales normalmente muestra la intensidad de señal recibida. El funcionamiento estándar permite operar fundamentalmente en uno de estos dos modos: punto a punto (P2P) o con coordinador. En el modo P2P, cada módulo se comunica con cualesquiera otros módulos, emitiendo broadcasts o direccionando un módulo remoto. Esto requiere que todos los módulos tengan su receptor constantemente encendido, dado que cualquiera puede recibir un mensaje en cualquier momento, pero permite mantener mensajes entre todos los módulos.

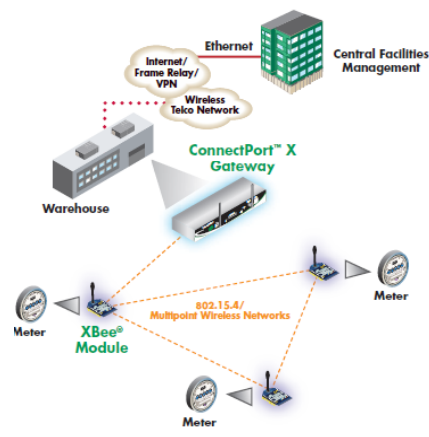


Figura 3. Arquitectura de una red de sensores basada en XBEE.

En el modo con coordinador, uno de los módulos se configura para el rol de coordinador y está siempre alerta, pudiendo los nodos remotos permanecer en modo bajo consumo por un tiempo determinado, el que se calcula para minimizar el consumo. Todas las comunicaciones de los nodos remotos son hacia el coordinador. Éste, puede almacenar hasta dos mensajes para un nodo remoto, hasta que este último, al retornar a funcionamiento normal, interroga al coordinador si tiene algún mensaje para él.

La configuración de los módulos se realiza mediante comandos AT, pudiendo operar fundamentalmente en dos modos: transparente o API. En el modo transparente, el módulo envía al remoto configurado como destinatario los mensajes que recibe por su puerto serie, y presenta en éste los mensajes que recibe del módulo remoto. Los cambios de configuración se realizan escapando a modo comando e ingresando comandos AT.



Figura 4. Ejemplo comando AT.

En el modo API, especialmente recomendado para el módulo coordinador, tanto comandos como mensajes y respuestas viajan dentro de un framing documentado, permitiendo mandar y recibir mensajes hacia y de múltiples nodos remotos de forma sencilla, en una única interfaz serie.

	Frame Fields	Offset	Example	Description
A P I	Start Delimiter	0	0x7E	
	Length	MSB 1	0x00	Number of bytes between the length and the checksum
LSB 2		0x02		
P a c k e t	Frame-specific Data	Frame Type	3	0x8A
		Status	4	0x00
	Checksum	5	0x75	0xFF - the 8 bit sum of bytes from offset 3 to this byte.

Figura 5. Ejemplo de paquete en el modo API.

En una primera fase de control es útil que los alumnos controlen el enlace P2P para comprobar el correcto funcionamiento del software desarrollado para el dispositivo. De esta forma solo son necesarios dos alumnos para esta primera fase. Una vez controlado el modo P2P, se puede dar paso al modo con coordinación, que ya abarca una red RF con múltiples nodos, y un protocolo más complejo.

3.2. Comunicación con el PC

Con el fin de facilitar al alumnado el trabajo con los módulos Xbee se les proporciona una interfaz de comunicación con la que poder conectar fácilmente el módulo al PC y poder controlarlo. En la siguiente figura se muestra la PCB en la que se aloja el módulo para su uso. Cabe destacar que la comunicación serie se realiza a través del puerto USB con el módulo XBee para el control de las comunicaciones y configuración del dispositivo.

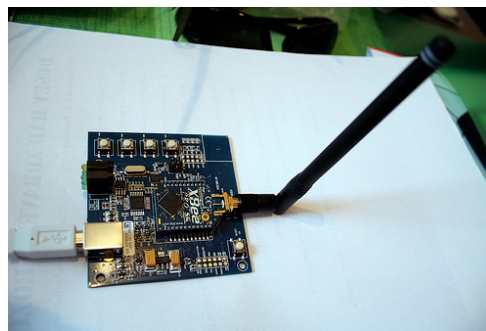


Figura 6. Módulo de conexión del XBee con PC.

Una vez expuesta la interfaz física de conexión con el módulo XBee, hace falta especificar las herramientas software de las que el alumno dispondrá para configurar el módulo. Se utilizará el entorno Visual Studio 2008 para realizar el software de control y configuración del dispositivo. En cuanto al lenguaje utilizado, se ha optado por C# [8].

C# es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET, que después fue aprobado como un estándar por la ECMA e ISO. Su sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de la plataforma .NET, similar al de Java, aunque incluye mejoras derivadas de otros lenguajes. Con el C# se pretendió que incorporase las ventajas o mejoras que tiene el lenguaje JAVA. Así se consiguió que tuviese las ventajas del C, del C++, pero además la productividad que posee el lenguaje JAVA y se le denominó C#.

Algunas de las características del lenguaje de programación C# son: Su código se puede tratar

íntegramente como un objeto. Su sintaxis es muy similar a la del JAVA. Es un lenguaje orientado a objetos y a componentes. Armoniza la productividad del Visual Basic con el poder y la flexibilidad del C++. Ahorramos tiempo en la programación ya que tiene una librería de clases muy completa y bien diseñada.

En cuanto al problema que nos ataña, el lenguaje C# permite la comunicación con dispositivos USB de forma bastante automática. El módulo XBee, al comunicarse utilizando el puerto USB de su PCB, se instalan como un puerto COM virtual dentro de los dispositivos del sistema. El lenguaje C# nos permite la creación de instancias de un puerto COM virtual y comunicarnos con él vía USB. Para ello se necesitan dos cosas: el número del puerto al que se ha conectado el dispositivo, y el protocolo de comunicación del módulo XBee. Al alumno se le proporciona el manual del dispositivo para que sea capaz de conocer el protocolo de comunicación; en cuanto al puerto COM, se puede obtener desde el administrados de dispositivos.

4. Metodología de elaboración y seguimiento

La práctica de laboratorio expuesta adopta la siguiente organización:

a. Clase teórica (2 horas): Se orientan totalmente a la realización de la práctica. En estas 2 horas se realiza una introducción al protocolo XBee, así como a los módulos utilizados en la práctica. De esta forma, el alumno comenzará la práctica con unos conocimientos básicos ya adquiridos, que le facilitarán la tarea a llevar a cabo en la misma.

b. Clases de laboratorio (2x4 horas): La práctica se divide en dos sesiones de cuatro horas cada una. En la primera sesión se hace entrega al alumnado del material a usar en la práctica, así como las instrucciones a seguir para elaborar correctamente la misma. A partir de ahí, teniendo ya un conocimiento previo del protocolo XBee, de redes inalámbricas y del lenguaje y entorno de programación a usar, el alumno comenzará a realizar la práctica con la ayuda de una guía aportada por el profesor. De esta forma, los primeros pasos, que son los más costosos, se realizan con la ayuda de un documento y con el

profesor como soporte. Es, a partir de ahí, cuando el alumno ejerce su autonomía y comienza a profundizar en la realización de la práctica. Tras esta primera sesión, se les permite a los alumnos llevarse el material para poder realizar la práctica con mayor comodidad fuera del laboratorio. Es en la segunda sesión de prácticas cuando el alumno ya ha conseguido dominar al protocolo de comunicación P2P y, el profesor, aporta la documentación, indicaciones y conocimientos necesarios para efectuar la última parte de la práctica: elaboración de la red inalámbrica por RF utilizando un nodo como coordinador. Las dos sesiones se realizarán con una distancia temporal de 3 o 4 semanas entre cada una, para permitir que los alumnos maduren los conocimientos y consigan alcanzar los objetivos de la práctica.

c. Trabajo fuera de horario (10 horas): el alumno pondrá en práctica los conocimientos adquiridos en las prácticas de laboratorio para elaborar los objetivos puestos en conocimiento por el profesor en la primera sesión de prácticas. Este trabajo se llevará a cabo en un horario libre, a disposición por el alumno. También es posible realizarlo en el aula habilitada por el profesor a tal efecto.

d. Para la evaluación se tiene en cuenta tanto el resultado obtenido en el proyecto, como el seguimiento del mismo; así como una presentación por parte del alumno del trabajo realizado (evaluación individual).

Cabe destacar que, aunque la evaluación de la práctica es individual, es necesaria la interacción de un conjunto de alumnos para poder realizar la práctica de forma satisfactoria; con lo cual un buen resultado global se verá reflejado como buenos resultados individuales de la mayoría de los alumnos. Así se fomenta el trabajo en grupo pero con un esfuerzo individual por parte de cada miembro de la clase. De igual forma, el buen resultado individual solo se obtendrá con un satisfactorio trabajo colectivo.

Aparte, como mecanismo tanto de ayuda como de control del trabajo individual de cada alumno, se dispone de unas tutorías especiales sólo para dicha práctica, en las cuales los alumnos expondrán el trabajo que llevan realizados hasta el momento, los problemas con los que se han encontrado y las decisiones tomadas para

solventarlos. Para que estas tutorías sean efectivas y, como se ha expuesto que las prácticas tienen una distancia temporal de 3 o 4 semanas entre ellas, se aporta a los alumnos un horario específico situado entre las dos sesiones de prácticas y en el que se obligará a la asistencia obligatoria de, al menos, una sesión.

5. Conclusiones

Se ha expuesto una práctica de laboratorio novedosa, que incorpora tanto las tecnologías más actuales, como los nuevos mecanismos de evaluación del alumnado. Se propone una evaluación híbrida en dos frentes: por un lado el trabajo se realizará tanto en aulas de laboratorio como en horario libre, dejando incluso un aula disponible fuera de horario para el uso responsable por parte de los alumnos.

Por otro lado, se propone una metodología de evaluación en la que interviene el trabajo individual de cada alumno, pero tiene gran peso el trabajo colectivo, puesto que el resultado final deberá ser una red inalámbrica en la que cada alumno controla un nodo.

La práctica expuesta y su mecanismo de evaluación permiten un aprendizaje escalonado, trabajo individual por parte del alumno y con unos fines colectivos; aportando un gran conocimiento de cara al mundo empresarial para el futuro próximo de los alumnos.

Agradecimientos

Este trabajo ha sido posible en parte gracias al proyecto VULCANO (TEC2009-10639-C04-02).

Referencias

- [1] M. Valero-García, J. Navarro, *FAQ sobre la docencia centrada en el aprendizaje*. <http://epsc.upc.edu/~miguel%20valero/continut3.html>.
- [2] Universidad de Sevilla. *Plan de Renovación de Metodologías Docentes 2007*. http://www.institucional.us.es/plan_renovacion.
- [3] T. Markham. *Project Based Learning, a guide to Standard-focused Project based learning for middle and high school teachers*. Buck Institute for Education, (2003).
- [4] G. Solomon. *Project Based Learning: a Primer, Technology and Learning*, vol. 23, num. 6 (Jan 2003).
- [5] H. Dolmans, et al. *Problem-based learning: future challenges for educational practice and research*. Blackwell Publishing Ltd 2005. *Medical Education* (2005); 39.
- [6] M. Hedley, S. Barrie. *An undergraduate microcontroller systems laboratory*. *IEEE Transactions on Education*, Volume: 41, Issue: 4, (Nov.1998).
- [7] Digi International Inc, *XBee-PRO® 868 OEM RF Modules*. <http://www.digi.com>, 2008.
- [8] Microsoft MSDN. *Introducción a Visual C#*. <http://msdn.microsoft.com/es-es/vcsharp>.

**Sesión 1B: Ingeniería del software /
Programación, algoritmos
y estructuras de datos**

Aprendiendo a diseñar software usando juegos de mesa de licencia libre como enunciado de prácticas

Pablo Trinidad, Manuel Resinas, Carlos Müller, José Antonio Parejo, Antonio Ruiz-Cortés

Departamento de Lenguajes y Sistemas Informáticos

Universidad de Sevilla

E.T.S.Ingeniería Informática, Av. Reina Mercedes s/n

41012 Sevilla

{ptrinidad, resinas, cmuller, japorejo, aruiz}@us.es

Resumen

El planteamiento de trabajos prácticos para el aprendizaje del diseño de software es una tarea compleja pues resulta necesario encontrar problemas de cierta envergadura, con puntos de variación y comportamientos complejos que eviten diseños triviales y que, además resulten motivadores para el alumno. En este artículo describimos nuestra experiencia de 3 años en el diseño, por parte del alumno, de software que implementa juegos de mesa. Los juegos de mesa cumplen los requisitos anteriores. Además, sus exhaustivas reglas son fácilmente usables como documentos de requisitos y la existencia de una amplia base de datos de juegos de licencia libre permite disponer con facilidad de multitud de problemas similares. Con este enfoque hemos mejorado significativamente el porcentaje de éxito de la asignatura.

Palabras clave

Ingeniería del Software, aprendizaje cooperativo, juegos de mesa, diseño de software, orientación a objetos

1. Motivación

El diseño de software es una labor que, a menudo, resulta complicada de abordar por el estudiante que se está iniciando en la materia. Aunque existe una gran variedad de razones por la que esto ocurre, en este artículo queremos destacar dos de ellas. En primer lugar, el diseño de software es una labor en la que hay que encontrar el equilibrio entre distintos aspectos contrapuestos. Por tanto, en muchos casos no es posible dar soluciones aplicables a cualquier situación, sino que sólo se puede transmitir un conjunto de principios básicos. Es el estudiante quien debe evaluar ventajas e inconvenientes de las distin-

tas alternativas para decidir la solución más adecuada. En segundo lugar, aunque los alumnos tienen ya una cierta experiencia en la implementación de código, les cuesta trabajo relacionar los conocimientos adquiridos con el nivel de abstracción utilizado en el diseño de software [5]. La consecuencia de esto es que, en general, no son capaces de comprender las implicaciones del diseño de software sobre el código y viceversa.

Una forma de abordar estos dos problemas es dotar a la docencia de un marcado carácter práctico, por ejemplo, mediante el desarrollo de una práctica de curso. De este modo, el estudiante se tiene que enfrentar en diversas ocasiones a la necesidad de elegir la solución más adecuada entre varias alternativas posibles. Además, en base a nuestra experiencia, es fundamental exigir también una implementación del diseño planteado por el estudiante, pues ayuda mucho a paliar el problema de la separación entre los mundos del diseño y la implementación.

Desafortunadamente, diseñar una práctica que sirva para desarrollar las capacidades de diseño de software de los estudiantes plantea varios retos: (1) Los problemas deben tener cierta envergadura; (2) deben ser abordables en un único cuatrimestre; (3) tienen que plantear un cierto reto desde el punto de vista del diseño. Es decir, deben requerir diseños que no sean excesivamente repetitivos ni triviales, e incluir comportamientos complejos con distintos puntos de variación. De este modo, los estudiantes podrán hacer uso de técnicas más avanzadas de diseño como patrones de diseño o refactorizaciones; (4) los requisitos deben estar bien definidos de manera que los estudiantes puedan centrarse en los aspectos más relacionados con el diseño; (5) finalmente, tienen que ser atractivos y motivadores tanto para el alumno como para el profesorado.

En este artículo, abordamos esta problemática describiendo nuestra experiencia de 3 años en el uso

de juegos de mesa como enunciados de prácticas de diseño de software. Como detallamos a lo largo de este trabajo, los juegos de mesa cumplen con los requisitos anteriormente mencionados, por lo que, en base a nuestra experiencia, constituyen una excelente fuente de problemas para dichas prácticas. Una ventaja adicional es la existencia de una amplia base de datos de juegos de mesa de licencia libre que permite disponer con facilidad de multitud de problemas similares, lo que resulta muy conveniente para eliminar las posibilidades de copia.

Por último, en este artículo también describimos los distintos cambios que hemos ido introduciendo en el desarrollo de la práctica a lo largo de estos años para resolver las deficiencias que hemos ido encontrando y que nos ha llevado a mejorar los resultados obtenidos en la asignatura. Por tanto, creemos que puede ser una contribución valiosa para los docentes que quieran abordar este tipo de prácticas.

El artículo está estructurado de la siguiente forma: en primer lugar, describimos el contexto académico en el que desarrollamos nuestra experiencia. En la sección 3 justificamos por qué los juegos de mesa son adecuados como enunciados de prácticas de diseño. La metodología de las prácticas se presenta en la sección 4. En la sección 5 detallamos la evolución de la práctica a lo largo de estos 3 años. En la sección 6 mostramos los resultados obtenidos en la evaluación de la asignatura. En la sección 7 enumeramos algunos trabajos relacionados. Finalmente, presentamos las conclusiones en la sección 8.

2. Contexto de la Asignatura

La experiencia descrita en este artículo se ha llevado a cabo en la asignatura Ingeniería del Software de Gestión II (ISG2), que se imparte en el primer cuatrimestre del tercer curso de la titulación de Ingeniería Técnica en Informática de Gestión. El lugar que ocupa la asignatura en el contexto de toda la titulación es el siguiente: en las asignaturas de cursos inferiores como Introducción a la Programación I y II, Análisis y Diseño de Algoritmos y Estructuras de Datos y Algoritmos, los alumnos aprenden los fundamentos de la programación, algoritmia y estructuras de datos. En cuanto a las asignaturas dedicadas a la ingeniería del software, en Ingeniería del Software de Gestión I, los alumnos aprenden técnicas de elicitación y análisis de requisitos. Es decir, adquieren la

capacidad de crear modelos de análisis de un dominio del problema. Finalmente, Ingeniería del Software de Gestión III, que se imparte en el segundo cuatrimestre del tercer curso, se centra en el diseño arquitectónico de aplicaciones de gestión de tamaño medio y se abordan problemas relacionados con la persistencia y la interfaz de usuario.

En este contexto, el objetivo de la asignatura Ingeniería del Software de Gestión II es dotar al alumno de los conocimientos y herramientas necesarias para abordar con éxito el proceso de diseño microarquitectónico de aplicaciones a partir de un conjunto de requisitos definidos informalmente y centrándose fundamentalmente en aspectos de mantenibilidad del código. Es necesario destacar que, aunque los alumnos aprenden los conceptos de orientación a objetos en las asignaturas previas, éstas se encuentran orientadas desde un punto de vista más algorítmico. Por tanto, los alumnos no han adquirido conocimientos en relación con el diseño orientado a objetos y conceptos básicos del diseño de software como la cohesión, el acoplamiento y la protección frente al cambio. Esto lleva a que en la asignatura se presentan métodos para diseñar desde cero como los principios generales de diseño, principios de asignación de responsabilidades (GRASP) [11] y patrones de diseño [9], y las refactorizaciones [7] para mejorar diseños existentes. También se hace especial hincapié en el desarrollo de pruebas unitarias junto al diseño e implementación de la aplicación. Otro objetivo es presentar el desarrollo de software como una actividad a realizar en equipo. Para ello se propone el uso de metodologías ágiles y de tecnologías para la gestión de equipos como son los sistemas de control de versiones.

3. ¿Por qué juegos de mesa?

Los objetivos anteriormente descritos establecen unos requisitos sobre la metodología y práctica docente que suponen un reto importante. En primer lugar, la propia problemática del diseño orientado a objetos exige plantear problemas de cierta envergadura, donde la evaluación de alternativas y aplicación de patrones de diseño tengan sentido. Para ello resulta beneficioso que las sesiones no se planteen de manera independiente, sino que se vertebren en torno a un problema común.

Además, el diseño e implementación orientado a

objetos es una tarea que se realiza en equipo. Esto es fundamental para adquirir las competencias de exposición y análisis de alternativas de diseño, planificación de las tareas asociadas a la implementación de los diseños y su seguimiento. Dado que el tamaño de los grupos de clase resulta excesivo para conseguir que se establezca esta dinámica, se hace necesario distribuir a los estudiantes en equipos y, a ser posible, plantear problemas distintos para cada equipo.

Dado que la inestabilidad y falta de claridad en los requisitos es una de las características típicas de los proyectos fallidos [2], los problemas a desarrollar necesitan establecer un conjunto de requisitos con las siguientes características:

- Claridad, el problema debe estar claramente especificado pero sin usar una aproximación formal, de manera que en el fases de análisis y diseño se planteen alternativas y problemas de interés.
- Tener un alto nivel de detalle, puesto que se pretende realizar diseños micro-arquitectónicos e implementación de los mismos.
- Estabilidad, de manera que las descisiones de diseño que se adopten no se vean comprometidas y los diseños puedan ser implementados.
- Incluir reglas de negocio que guíen las decisiones de diseño.

Se plantea pues la necesidad de buscar una fuente de problemas con estas características. En 2008 se optó por usar juegos de mesa clásicos (Stratego, Monopoly, Poker, Risk, etc.) como fuente de requisitos para los problemas. Posteriormente, dada la necesidad de plantear problemas distintos para cada grupo y año académico, se optó por usar juegos de mesa de libre distribución publicados como *Print & Play* (P&P). Los juegos P&P son de licencia libre y están preparados para imprimir todo el material necesario para jugar. Esto permite que los alumnos tengan acceso a los juegos de mesa a muy bajo coste y puedan familiarizarse con las reglas de los mismos jugando.

Además de satisfacer los requisitos anteriormente enumerados, el uso de juegos de mesa P&P presenta las siguiente ventajas:

1. **Motivación:** para los alumnos la implementación de un juego resulta mucho más motivadora que una aplicación de gestión.
2. **Dinamismo:** los comportamientos a modelar suelen ser muy dinámicos y cambiar según la ronda del juego en la que nos encontremos. Esto permite que se puedan aplicar patrones de diseño y soluciones más complejas que en aplicaciones de gestión al uso.
3. **Variabilidad:** la mayoría de los juegos tiene variantes y reglas especiales. Estos puntos de variación permiten plantear problemas de diseño en cuanto a la protección frente al cambio y evaluar la flexibilidad de los diseños. Además, muchos de los juegos se encuentran en fase de desarrollo y pueden aparecer nuevas versiones que cambian las reglas durante el desarrollo de la asignatura y que deberán considerarse. Esto simula los requisitos cambiantes de un entorno real.
4. **Variación:** Aún cuando es posible encontrar ciertas similitudes en los problemas a resolver, las diferencias entre las reglas y mecánica de los juegos suelen ser suficientemente grandes como para plantear problemas diferentes y evaluar la originalidad de las soluciones. La posterior exposición en clase de dichos problemas y sus soluciones dota a los alumnos de distintos puntos de vista para resolver problemas similares. Esto permite desarrollar la competencia de evaluación de ventajas e inconvenientes.
5. **Extensión:** BGG[1] es una base de datos de juegos que actualmente supera los 50.000 juegos registrados en 10 años. BGG es el mayor repositorio de juegos de mesa existente en la red, tanto comerciales como libres. A modo de referencia, en 2010 se publicaron 320 nuevos juegos P&P.

4. Metodología

El enfoque de nuestra asignatura exige la elaboración de una lista de juegos previa al comienzo del curso académico. Los profesores realizan las siguientes actividades preparatorias:

1. Búsqueda de juegos de mesa: se consulta la base de datos BGG en busca de nuevos juegos publicados en el último año bajo licencia P&P. Se elabora una lista preliminar de unos 40 juegos aproximadamente.
2. Descarte de juegos: el objetivo de esta fase es

descartar aquellos juegos que no sean realizables como software. Ejemplos de éstos son juegos en los que los jugadores deben narrar una historia o realizar gestos. Se supervisa que el material sea completo. Para agilizar esta fase y evitar una lectura exhaustiva de las reglas, se hace uso de blogs especializados para leer reseñas de los juegos seleccionados.

3. Valoración de la complejidad: se realiza una lectura rápida de las reglas y se analiza su complejidad. Se definen 4 niveles en los que el profesor valora la dimensión del problema, las interacciones entre los elementos del juego, los casos especiales en las reglas, las variantes y el nivel de madurez de las reglas. La calificación máxima a la que puede aspirar un grupo va en consonancia con su complejidad.
4. Elaboración de la lista definitiva: Puesto que anualmente el número de alumnos matriculados se encuentra en torno a los 130 y los grupos suelen ser de 5 a 6 personas, por lo que se debe obtener una lista de al menos 25 juegos. Se busca un mayor número de juegos de los niveles de complejidad medios (2 y 3) que son los más demandados por los estudiantes.

Como se puede apreciar, no es recomendable delegar a los grupos de estudiantes la responsabilidad de elegir juegos de mesa, puesto que no disponen del conocimiento necesario para evaluar un juego sin conocer las implicaciones que tienen las reglas en un diseño software. Además el alumno no debe invertir tiempo en una actividad que no contribuye a lograr los objetivos marcados para la asignatura.

En la primera semana del curso académico, se forman grupos de 5 ó 6 estudiantes de la siguiente forma:

1. Solicitudes: Los alumnos se inscriben en grupos mediante solicitudes por escrito. Las solicitudes pueden ser individuales o grupales (2 a 6 estudiantes por inscripción). En la solicitud deben especificar la complejidad deseada mediante un rango que cubra uno o dos niveles de complejidad consecutivos. Así podrán solicitar una complejidad de 4 ó de 2-3 pero no de 1 ó 3.
2. Formación de grupos: Los profesores forman grupos de estudiantes que hayan solicitado un nivel de complejidad lo más similar posible. Se intenta formar grupos de 6 personas.

3. Asignación de juegos a grupos: se realiza la asignación según la complejidad solicitada. Se publica la lista de grupos y trabajos a las 48 horas, ofreciendo los enlaces necesarios para la descarga de los juegos de mesa.

Los estudiantes disponen entonces de una semana para realizar un video de 7 minutos de duración máxima en el que expliquen la mecánica del juego¹. Esto obliga a:

- Jugar al menos una vez al juego y por tanto conocer todos los detalles del mismo.
- Explicar el juego, lo que es una evaluación implícita del conocimiento del mismo.
- El realizar una actividad lúdica ayuda a que los miembros del grupo de trabajo se conozcan entre ellos y a establecer un mejor clima.
- Detectar roles y liderazgo y organizar el grupo en base a los mismos.
- Terceras partes pueden comprender rápidamente las reglas, tanto profesores para la evaluación como alumnos de otros grupos que quieran evaluar el trabajo o tener referencias de otros grupos.

Una vez publicados los videos, los estudiantes realizan evaluaciones cruzadas. Cada alumno evalúa 2 videos de otros grupos rellenando un sencillo formulario elaborado mediante Google Forms. Esto genera aproximadamente 12 revisiones por video, tomándoles únicamente 20 minutos de trabajo por alumno. Los profesores recopilan dicha información y publican los resultados para cada grupo. Así los grupos son conscientes de la calidad relativa de su video.

Desde este momento, los estudiantes deben realizar varias iteraciones, en cada una de las cuales deben elaborar una memoria con el trabajo desarrollado. Como entrega final, deben presentar junto con la memoria una implementación completa del juego asignado. Todos los entregables son públicos para que los alumnos puedan tomar ideas de otros grupos. Para publicar toda esta documentación se ha incorporado para el curso 2010 la plataforma Ópera en la gestión de grupos. El lector puede acceder a ella a través de opera-portal.us.es/isg2.

¹En http://www.youtube.com/results?search_query=isg2 se puede encontrar los videos realizados para esta asignatura

5. Evolución

5.1. 2008

En este año se incorpora por primera vez los juegos de mesa como fuente de problemas de diseño. Los estudiantes deberían realizar un diseño en grupo para un juego de mesa asignado. Al tratarse de una experiencia piloto, decidimos no imponerlo como requisito obligatorio para aprobar la asignatura. Lo pusimos a prueba como un mecanismo de evaluación alternativo a un examen final. Sin embargo, sólo podían optar a realizar esta alternativa aquellos estudiantes que superasen un test de conocimientos teóricos a mitad del cuatrimestre. Este prerrequisito dejó únicamente dos meses para la realización del trabajo. El trabajo era evaluado al final con una nota común para todos los integrantes del grupo independientemente de su esfuerzo. Para evaluar a cada integrante de manera individual se evaluaban otros dos aspectos: (1) cada integrante del grupo debía responder individualmente a preguntas en una defensa pública del trabajo; y (2) el alumnado debía resolver individualmente un problema de diseño.

Prácticamente la totalidad de los estudiantes hizo el test de conocimientos teóricos pero no todos los que aprobaron optaron por diseñar juegos de mesa, sino que prefirieron la prueba escrita. Encontramos tres causas posibles para explicar este fenómeno: (1) utilizaron el test para autoevaluar su nivel de asimilación de los conocimientos teóricos; (2) declinaron hacer el trabajo por factores externos como la carga docente en otras asignaturas; o (3) al disponer de dos mecanismos alternativos, un estudiante se suele decantar por el examen a pesar del riesgo que conlleva una prueba única puesto que tiene la sensación de que un examen exige menos esfuerzo que un trabajo en grupo.

El nivel de acabado de los diseños obtenidos por los estudiantes que decidieron hacer el trabajo en grupo no fue todo lo elevado que se esperaba. Como principales causas podemos citar:

- El trabajo estaba sobredimensionado con respecto al poco tiempo que tuvieron para realizarlo.
- Al alumnado le costaba asimilar la mecánica de los juegos de mesa que se le fueron asignados.
- Algunos juegos de mesa eran de licencia comercial y aun disponiendo de las reglas de los

mismos, no era posible en muchos casos adquirir una edición física del juego para poder evaluarlo y mejorar el entendimiento de sus reglas.

Los profesores de la asignatura también se vieron afectados por el nuevo mecanismo de evaluación, apareciendo dificultades tales como:

- Dificultad en detectar las horas invertidas y la productividad de cada integrante del grupo.
- Sobrecarga en las múltiples revisiones que el profesorado tuvo que hacer de cada una de las entregas en las que se dividió el trabajo.
- Sobredemanda de tutorías por parte del alumno, al no poder realizar un seguimiento frecuente en clase.

5.2. 2009

Comprobada la efectividad de la experiencia piloto de 2008 en algunos grupos comprometidos con el nuevo sistema de evaluación, decidimos establecer la obligatoriedad de realizar el diseño de un juego de mesa. Así se evitaba que existieran alumnos que pudieran superar la asignatura sin haber realizado un diseño de envergadura en grupo.

Además de este cambio nuclear, se pretenden corregir otros defectos detectados durante el año anterior. Puesto que la asignatura queda vertebrada por el trabajo en grupo, se decide adelantar el comienzo del mismo al principio del cuatrimestre. Por esta razón se elimina el prerrequisito de aprobar un test de conocimientos. Para evitar la dificultad de acceder al material de juegos comerciales, se hace uso únicamente de juegos con licencia P&P. Para evitar la carga de trabajo que produce el seguimiento constante de los trabajos por parte del profesor y la sobredemanda de tutorías, se implementa un sistema de evaluaciones cruzadas online en la que cada grupo debe revisar el trabajo de al menos otros dos grupos cada vez que realiza una entrega. Para poder reflejar los esfuerzos desiguales de cada uno de los miembros de un grupo durante la realización del trabajo, se permite a cada grupo establecer diferentes ponderaciones de la calificación del trabajo en su entrega final. Este mecanismo ofrece al estudiante la posibilidad de resolver conflictos debidos a diferentes niveles de implicación de sus compañeros.

Al disponer de más tiempo para realizar la práctica se introduce por primera vez la realización de

un video explicativo del juego. Para introducir al alumno en el uso del sistema de evaluaciones cruzadas, se realiza una primera evaluación de los videos.

Aunque la experiencia fue satisfactoria y consideramos que los cambios tuvieron un efecto global positivo, detectamos algunas deficiencias a corregir para el siguiente curso:

- El reparto de calificaciones al final del cuatrimestre no reflejan proporcionalmente el esfuerzo real del alumnado. En muchos casos un esprint final de un estudiante que no ha trabajado lo suficiente durante el cuatrimestre no se refleja correctamente en las ponderaciones. En los casos detectados se entrevistó a los estudiantes por separado, argumentando todos que aunque el mecanismo de ponderación era correcto, los problemas «se enfrían» una vez entregado el trabajo.
- Al formar los grupos al comienzo del curso a lo largo del cuatrimestre surgen alumnos que no se han inscrito aún. Al no establecerse una normativa clara al respecto, se les asignó a grupos «en marcha».

En cuanto a los problemas que afectan directamente al profesorado destacamos los siguientes:

- Baja interacción del profesor con el alumnado. El gran número de revisiones cruzadas hace aparentemente innecesaria la intervención del profesor. No obstante, se observó un gran número de revisiones cruzadas de nulo o escaso valor, por lo que hubo que implementar un buscador automático que ayudase a detectar estos casos.
- Sobrecarga en la gestión de las revisiones. La asignación de revisiones a grupos requiere la intervención explícita del profesor. Al no existir una fecha única para las entregas intermedias, se generaba un flujo constante de trabajos a revisar y peticiones de revisión, lo que obligaba a realizar diariamente una asignación de revisiones.

5.3. 2010

En términos generales tanto profesores como alumnos estaban satisfechos con el resultado obtenido con los cambios realizados en el curso anterior. No obstante, se realizan algunas modificaciones para

paliar los problemas detectados. Para poder reflejar correctamente el esfuerzo del alumno en la calificación final del trabajo en equipo, se decide introducir una evaluación interna de los grupos en cada uno de los entregables intermedios. De esta forma los alumnos deben repartir un número determinado de puntos de la forma que entiendan más justa en cada entrega. Al realizarse entregas aproximadamente cada dos semanas, se permite reflejar con mayor nivel de detalle el grado de implicación de cada alumno. La nota final se asignará al grupo, pero se repartirá de forma proporcional a la suma de los puntos de las entregas intermedias.

Para evitar la inscripción tardía en grupos de algunos estudiantes, se hace uso del mecanismo anteriormente descrito para permitir la inscripción de nuevos miembros en grupos que ya han comenzado a trabajar. Estos alumnos reciben una calificación de cero puntos para todas las entregas que su grupo haya realizado con anterioridad. Con objeto de aumentar la intervención del profesor en el seguimiento de los trabajos, se sustituye un gran número de las prácticas de laboratorio por tutorías presenciales. Con esto se eliminan las revisiones cruzadas obligatorias. No obstante, al estar la documentación de todos los grupos disponibles a través de la plataforma Opera, los grupos pueden consultar los entregables de otros grupos con total libertad.

Por último, hemos establecido que todos los juegos de mesa deben implementarse completamente para ser evaluados. No se admiten pruebas de concepto ni implementaciones parciales.

El resultado percibido en este año ha sido muy satisfactorio. Hemos detectado algunos pequeños problemas que se resolverán en años venideros como por ejemplo:

- La calificación por entregables refleja correctamente el esfuerzo del estudiante y ha conseguido evitar en gran medida el retraso en las inscripciones en los grupos. No obstante existe una desviación entre esfuerzo y productividad que no hemos visto reflejado en algunos grupos.
- Al centrarse en la implementación, los grupos no le dan relevancia a la documentación.
- Al no especificar el contenido que debe presentarse en cada iteración, algunos grupos presentan problemas de organización debido a la falta de liderazgo.

6. Resultados académicos

En el Cuadro 1 se presentan los resultados académicos obtenidos desde el año 2007 hasta el presente curso. Los resultados se presentan para cada una de la convocatorias oficiales (febrero, septiembre y diciembre) y para la evaluación continua.

En todas estas convocatorias, junto a la práctica descrita, se ha realizado también un test de conocimientos teóricos y un problema escrito de diseño con pequeñas variaciones en las ponderaciones en cada año, pero, en todos los casos obligando a aprobar cada una de las partes por separado. Las pruebas superadas se conservan para todas las convocatorias de un mismo curso académico. En el caso de la evaluación continua, se realizan las mismas pruebas pero durante las clases, en lugar de en un examen final. Esto permite al alumnado disponer de dos intentos para aprobar en primera convocatoria.

Se puede observar cómo el número de aprobados entre la evaluación continua y la primera convocatoria es superior cada año respecto a los resultados obtenidos en el año 2007. Un resultado sorprendente es que hemos logrado reducir el número de alumnos no presentados del 48 % en primera convocatoria del 2007 al 18 % en la misma convocatoria del 2010. Si bien entendemos que la extinción de la asignatura para el año 2012 como consecuencia de la aplicación del EEES, puede ser un catalizador de este efecto, creemos que la metodología aplicada ha animado a los estudiantes a presentarse a la asignatura. En este sentido, cabe destacar que de los 22 estudiantes que no se han presentado en el curso 2010, 12 de ellos no lo hicieron en el curso 2009 y de éstos 9 tampoco se presentaron en el curso 2008.

7. Trabajos relacionados

Desde hace varias décadas se viene comentando la capacidad motivadora de los juegos en procesos cognitivos del alumnado desde temprana edad [3, 4, 14, 15].

Más cercanos a nuestro contexto, existen trabajos en los que se utilizan juegos para enseñar técnicas de inteligencia artificial a los estudiantes como, por ejemplo [6, 12]. Nosotros, en cambio, nos centramos en el diseño orientado a objetos y, de hecho, no exigimos ningún tipo de inteligencia artificial implementada en el juego.

Más relacionado con nuestro trabajo encontramos [10], donde los autores utilizan la implementación de un Sudoku como medio para transmitir conceptos de patrones de diseño como el patrón MVC, observador, fábrica abstracta y estrategia. Además les permite poner en práctica conceptos de estructuras de datos y algoritmos vistos en asignaturas anteriores y se les introduce en la tecnología del desarrollo de interfaces de usuario. Las principales diferencias con nuestro trabajo son dos. En primer lugar, nuestras prácticas no son guiadas ni tenemos un diseño previo planteado como objetivo para los alumnos, sino que son ellos mismos los que van haciendo el diseño de forma autónoma, lo que hace que el planteamiento de la práctica sea radicalmente distinto. En segundo lugar, como nos centramos en el diseño de la microarquitectura y no tratamos explícitamente aspectos relacionados con el diseño de interfaces de usuario, lo que prima es la complejidad del modelo de datos y de las reglas del juego. Por tanto, los tipos de juego planteados son significativamente distintos con una complejidad en cuanto a lógica de negocio superior a la de los sudokus, tres en raya o similares.

8. Conclusiones y Evoluciones Futuras

Después de tres años desarrollando y madurando nuestra idea, y en base a los resultados obtenidos podemos llegar a las siguientes conclusiones:

1. Los juegos de mesa son un instrumento adecuado como fuente de problemas de diseño de envergadura.
2. Con los juegos de mesa hemos aumentado el grado de implicación del alumnado en la asignatura.
3. El estudiante adquiere una sensibilidad especial por los problemas que surgen del diseño de sus juegos. Esta sensibilidad ayuda en el aprendizaje de aspectos avanzados del diseño.
4. El profesor tiene mayor certeza de que el alumno ha cumplido con los objetivos de la asignatura.
5. Existe una mayor motivación del profesorado al aumentar la interacción con el alumnado.
6. Ha aumentado el número de estudiantes que realizan el proyecto final de carrera con los profesores de la asignatura.

Convocatoria	2007			2008				2009				2010	
	1C	2C	3C	Cont.	1C	2C	3C	Cont.	1C	2C	3C	Cont.	1C
Aprobados	64	9	3	56	7	12	4	44	39	10	4	35	30
Suspensos	9	8	5	17	13	11	2	33	19	5	1	52	36
No Presentados	69	61	61	61	58	48	53	59	34	38	38	36	22

Cuadro 1: Resultados académicos de los últimos 4 años

En cuanto a los aspectos mejorables que hemos detectado en el último año, hemos decidido introducir los siguientes cambios para el año 2011:

1. Hacer uso de la plataforma Opera no sólo como repositorio de contenido sino también para la creación de grupos.
2. Debido a la complejidad que alcanza nuestro sistema de evaluación, se introducirán contratos de aprendizaje. [13] con objeto de aumentar la garantía del profesor en que el alumno conoce los mecanismos de evaluación.
3. Uso de rúbricas para evaluar las presentaciones y las memorias [8].
4. Uso de formatos de memoria predefinidos a fin de mejorar la calidad de los entregables.

Referencias

- [1] Boardgamegeek. www.boardgamegeek.com.
- [2] Chaos. Technical report, Standish Group International, 1994.
- [3] J. Bruner. Play, thought, and language. *Peabody Journal of Education*, 60:6–69, 1983.
- [4] A. Butler, E. Gotts, and N. Quisenberry. Play as development. *Columbus, OH*, 1978.
- [5] Cristina Cachero, Otoniel López, and María José Durá. Del diseño a la implementación del software: Una metodología de cohesión. In *Actas de las X Jornadas de Enseñanza Universitaria de la Informática (JENUI 2004)*, pages 263–270, 2004.
- [6] O. Colomina, P. Compañ, R. Satorre, F. Aznar, P. Suau, and R. Rizo. Aprendiendo mediante juegos: Experiencia de una competición de juegos inteligentes. In *Actas de las X Jornadas de Enseñanza Universitaria de la Informática (JENUI 2004)*, pages 513–515, 2004.
- [7] Martin Fowler. *Refactoring: Improving the Design of Existing Code*. Addison-Wesley Professional, 1999.
- [8] Isabel Gallego, José Manuel López, Eva Rodríguez, Esther Salamí, Eduard Santamaría, and Miguel Valero. Presentaciones orales a un coste razonable. In *Actas de las XVI Jornadas de Enseñanza Universitaria de la Informática (JENUI 2010)*, pages 25–32, 2010.
- [9] Erich Gamma, Richard Helm, Ralph Johnson, and John M. Vlissides. *Design Patterns. Elements of Reusable Object Oriented Software*. Addison-Wesley, 1994.
- [10] Juan A. Recio García, Baltasar Fernández Manjón, and Marco Antonio Gómez Martín. Aprendizaje de técnicas avanzadas de programación orientada a objetos mediante programación de juegos. In *Actas de las XII Jornadas de Enseñanza Universitaria de la Informática (JENUI 2006)*, pages 387–394, 2006.
- [11] Craig Larman. *UML y Patrones: Una introducción al análisis y diseño orientado a objetos y al proceso unificado*. Prentice-Hall, 2 edition, 2001.
- [12] Juan A. Suárez Romero, Amparo Alonso Betanzos, and Marcos Villares Souto. Entorno de juegos para el desarrollo de prácticas de inteligencia artificial. In *Actas de las XII Jornadas de Enseñanza Universitaria de la Informática (JENUI 2006)*, pages 537–544, 2006.
- [13] T. Franquet Sugañes, D. Marín Consarnau, M. Marqués, and E. Rivas Nieto. El contrato de aprendizaje en la enseñanza universitaria. In *4º Congreso Internacional Docencia Universitaria e Innovación*, pages 1–16, Barcelona, 2006.
- [14] Pauline Davey Zeece and Susan K. Graul. Play and its role in the mental development of the child. *Play-Its role in development and evolution*, pages 537–554, 1976.
- [15] Pauline Davey Zeece and Susan K. Graul. Learning to play: Playing to learn. *Early Childhood Education Journal*, 18:11–15, 1990.

El eportfolio en la Ingeniería del Software como una herramienta de reflexión en el proceso de aprendizaje

Reyes Grangel, Cristina Campos, Cristina Rebollo, Inmaculada Remolar
Departament de Llenguatges i Sistemes Informàtics
Universitat Jaume I
Campus del Riu Sec
E-12071 Castelló de la Plana
{grangel, camposc, rebollo, remolar}@uji.es

Resumen

La aplicación de nuevas metodologías de enseñanza-aprendizaje y sistemas de evaluación, basados principalmente en los contratos de aprendizaje, ha conseguido mejorar visiblemente los resultados de los alumnos en la superación de la asignatura 'Ingeniería del Software'.

Sin embargo, el profesorado de dicha asignatura se plantea, con el ánimo de mejorar, si estos cambios hacen que realmente los alumnos estén aprendiendo más. Para dar respuesta a esta cuestión se ha utilizado el eportfolio como una herramienta de evaluación enfocada a la reflexión sobre el propio proceso de aprendizaje.

En este artículo, se explica esta experiencia docente, detallando la situación de partida, la planificación, ejecución y evaluación de la misma, así como los resultados obtenidos.

1. Introducción

La aplicación de nuevos métodos de enseñanza-aprendizaje y nuevas formas de evaluar lo aprendido es una realidad que se está adoptando progresivamente en el panorama universitario español [7], ligado al llamado proceso de Bolonia [9]. Aunque este proceso no propugna ningún cambio metodológico, sino que solo persigue una adaptación de los estudios universitarios para que sean homologables en un espacio común europeo, el Espacio Europeo de Educación Superior (EEES) [10], ambas mejoras se están realizando al unísono. Es por ello que en este periodo de adaptación, la cuestión que se plantea parte del profesorado inmerso en este proceso y preocupado por adoptar nuevas formas de enseñar y aprender es si realmente los alumnos aprenden más con los nuevos métodos. Puesto que lo que sí se observa es un aumento de las calificaciones.

En este contexto, se presenta este artículo, cuyo objetivo es mostrar una experiencia docente de mejora aplicada a una asignatura de Ingeniería del Software

que progresivamente ya se ha adaptado al EEES, y cuyo punto de mira está enfocado ahora en investigar el grado de aprendizaje de los alumnos con los nuevos métodos.

El artículo se estructura, además de esta breve introducción, de la siguiente forma. En la sección 2, se expone la necesidad y contexto de la experiencia docente. En la sección 3, se analiza el eportfolio como método de enseñanza-aprendizaje y su uso en el proceso de evaluación. En la sección 4, se explica con detalle la experiencia docente llevada a cabo en la asignatura de Ingeniería del Software, analizando la ejecución de la misma, y el resultado final junto con las lecciones aprendidas. Por último, la sección 5 presenta las principales conclusiones.

2. Contexto de la experiencia docente

En esta sección se explica la situación de partida sobre la cual se ha aplicado esta experiencia docente, mostrando en primer lugar las principales características de la asignatura sobre la cual se ha realizado y en segundo cual es el origen de la misma y su necesidad.

2.1. La asignatura: Ingeniería del Software

La Ingeniería del Software es una asignatura troncal y anual que se sitúa en el 4º curso de la titulación de Ingeniería Informática en la Universitat Jaume I. Esta asignatura, de 10,5 créditos, es la primera correspondiente al bloque del mismo nombre que se imparte a los alumnos, si bien éstos ya han cursado diversas asignaturas de programación y bases de datos. Por lo tanto, se puede considerar introductoria, siendo su principal objetivo el proporcionar una visión general de la disciplina. Por otra parte, esa visión debe ser de todo el conjunto de actividades que se llevan a cabo en el proceso de desarrollo de un sistema informático, si bien algunas de ellas como las de implementación no se analizan en profundidad puesto que existen diversas asignaturas de programación en la titulación [6].

Cuadro 1: Planificación de la asignatura Ingeniería del Software.

Ingeniería del Software			
UD	Temas	Nº	
		ST	SP
UD1	Procesos de desarrollo de sistemas de información	1	–
	Reingeniería de procesos	3	2
UD2	Definición de los requisitos del SI	1	2
	Análisis del SI	2	2
	Diseño del SI	2	2
	Implementación del SI	1	2
UD3	Paradigma Orientado a Objetos	1	–
	UML: Diagramas de estructura	1	–
	UML: Diagramas de comportamiento	1	–
	Especificación de requisitos con UML	2	2
	Análisis con UML	5	3
	Patrones de diseño con UML	5	3
	Arquitecturas de implementación con UML	2	2
	Model Driven Architecture (MDA)	3	–
CT	Presentación escrita: LaTeX, ePortfolio	–	1
	Presentación oral: transparencias, póster, Prezi	–	1

El contenido de la asignatura se estructura en tres unidades didácticas: la 1 sobre el estudio de los fundamentos de la Ingeniería del Software (**UD1**), la 2 que presenta la metodología estructurada (**UD2**) para el desarrollo de sistemas informáticos, y la 3 en la cual se analiza la metodología orientada a objetos (**UD3**). Además, se incluye una cuarta sobre competencias transversales (**CT**) en la cual se trabaja la expresión escrita y la presentación oral. En el Cuadro 1 se presenta su planificación indicando para cada una de las unidades didácticas, mostradas en la primera columna, los temas que la componen en la segunda, y el número de sesiones de teoría (Nº ST) y de prácticas (NºSP) ¹ asignadas en las dos últimas columnas.

Esta asignatura ha sido adaptada al EEES en un proceso de mejora continua a lo largo de diversos cursos, siendo los resultados obtenidos apreciables. Tanto en cuanto al número de aprobados sobre los presentados como sobre los matriculados, y por otra parte también en cuanto a otros aspectos como asistencia a clase, trabajo desarrollado por el alumnado, implicación del profesorado, etc. Dicha adaptación, ha supuesto un cambio tanto en la metodología de enseñanza-aprendizaje como en el sistema de evaluación, destacando la implantación satisfactoria de los contratos de aprendizaje [6].

En el sentido más amplio se establece un contrato de aprendizaje cuando un alumno y un profesor intercambian sus opiniones, comentan sus necesidades, comparten proyectos y deciden en colaboración

¹Cada sesión de teoría es de dos horas, al igual que las de prácticas.

la forma de llevarlos a cabo y la evaluación del aprendizaje que han realizado y de los resultados conseguidos [3]. Como método de enseñanza-aprendizaje y evaluación un contrato de aprendizaje supone la implantación de un sistema en el cual profesor y alumno negocian cuales van a ser los objetivos de aprendizaje, cuales son los recursos que se van a usar para conseguirlos, cuales son las evidencias que el alumno va a presentar para demostrar que ha alcanzado esos objetivos, y como el profesor va a evaluar esas evidencias. Toda esta información se puede recoger en un documento que simbólicamente firman tanto alumno como profesor, pero lo verdaderamente importante es el compromiso del profesor a ofrecer un feedback que mejore la autonomía del alumno, y sobre todo del alumno a hacerse responsable de su proceso de aprendizaje.

La implantación del contrato de aprendizaje ha sido adaptada a las características de la asignatura, de manera que se ofrece un modelo de contrato al alumnado el cual se negocia a principio de curso. Progresivamente se han ido modificando los pesos de cada elemento presente en este modelo, así el examen final, que suponía un 70% de la nota, se ha convertido en diferentes pruebas de evaluación continua que suponen un 40%. El resto del peso de la nota está repartido entre las prácticas, un 30%, que se realizan en grupo; y ejercicios y actividades entregables, un 30%, que se entregan con carácter individual pero que en ocasiones se trabajan previamente en grupo [6].

2.2. Planteamiento de la experiencia

La mejora llevada a cabo ahora y que se explica en este artículo tiene un cariz diferente a las anteriores. Con anteriores actuaciones se ha conseguido cambiar la metodología docente adaptándola al EEES y con ello mejorar los resultados de la asignatura [6]. La principal cuestión que en estos momentos se plantea al equipo docente es si realmente los alumnos están aprendiendo más, puesto que con la nueva metodología ha ascendido el número de aprobados.

Pero qué significa aprender más, mayor cantidad de conocimientos, de habilidades, quizás más rápido, mejor. Si se relaciona directamente el aprendizaje con la nota obtenida, se podría concluir que sí, puesto que progresivamente se ha aumentado tanto el número de aprobados en la asignatura como la nota media obtenida. Pero realmente el aprendizaje sobre una determinada materia va más allá de la nota obtenida en ella. Puesto que es posible que dos alumnos que hayan obtenido la misma nota, hayan aprendido uno

más que el otro y este hecho no se vea reflejado en la nota, debido fundamentalmente a lo limitado de la evaluación (número de evidencias de aprendizaje, tiempo, etc). Por lo tanto, queda claro lo difícil que resulta responder a esta pregunta en términos cuantitativos reales, y por ello la experiencia que aquí se presenta trata de dar respuesta desde otra perspectiva más cualitativa.

Para conseguirlo se ha utilizado el eportfolio como herramienta de reflexión durante el proceso de aprendizaje. La experiencia se ha llevado a cabo de forma que el diseño y la implantación del eportfolio no sólo ha dado respuesta a la cuestión planteada sino que también ha hecho posible que el alumnado junto con el profesorado reflexionen sobre este proceso con la finalidad de poder introducir las mejoras necesarias durante y una vez acabado el mismo. Además, el eportfolio ha sido utilizado como método de evaluación de parte de la asignatura.

Por otra parte, con el uso del eportfolio en la asignatura se ha pretendido introducir en la docencia una innovación que permita asumir a los alumnos un nuevo rol, pues en el LCMS (Learning Content Management System) usado [2], el profesorado es quien decide los contenidos, estructura, forma de aprender, etc. Mientras que con el nuevo rol principal que asume el alumnado con el uso del eportfolio se fomenta su autonomía y capacidad para aprender y reflexionar sobre su propio aprendizaje [8], y el profesorado por su parte actúa como apoyo y evaluador de este proceso.

3. El eportfolio como método de aprendizaje

En un contexto de enseñanza-aprendizaje centrado en el alumno, el eportfolio apareció en los años 90 como una estrategia pedagógica que permitía a los alumnos gestionar sus experiencias de aprendizaje, a la vez que permitía al profesor evaluar tanto las evidencias del aprendizaje como el proceso en sí mismo. Su principal potencial pedagógico se encuentra en el hecho de que los alumnos asumen un nuevo rol y son capaces de evaluar su progreso en el proceso de aprendizaje, mientras que el profesorado por su parte puede apoyar dicho proceso a través de medios tecnológicos [7]. Lo cual según [10] es uno de los principales cometidos del profesor para la adaptación al EEES.

El portfolio o carpeta docente sería la versión analógica del eportfolio, la técnica en sí era ya utilizada con fines educativos antes de la aparición de las TIC como método para la recopilación de las evidencias

de aprendizaje. Por lo tanto, el eportfolio seguiría la misma filosofía pero soportada por las TIC. De esta forma, se puede definir el **eportfolio** como un espacio virtual (normalmente basado en web) el cual contiene los productos digitales (artefactos o reflexiones) que demuestran en un determinado ámbito de conocimiento el aprendizaje que ha llevado a cabo el estudiante. Esta demostración puede ser hecha con fines educativos, de colaboración profesional, amistad, social, etc. [7, 8].

Otro de los entornos virtuales que cada día se utilizan más en la docencia son los Learning Content Management Systems (LCMS). La principal diferencia que existe entre un sistema LCMS y un eportfolio es que en el primer caso el control recae en el profesor, mientras que el eportfolio es un sistema en el que el foco de control recae el alumno, con lo cual se favorece el proceso de aprendizaje centrado en el alumno [4].

Los principales objetivos de un sistema basado en el eportfolio se resumen en ofrecer al estudiante [7]:

1. Un espacio web personal que le permita almacenar, clasificar y/o seleccionar sus productos de aprendizaje.
2. Un apoyo en línea por parte del profesorado con la finalidad de guiar el proceso de aprendizaje.
3. La libre administración de sus tareas académicas con el fin de fomentar su autonomía en el aprendizaje.

Con frecuencia el contrato de aprendizaje incluye la elaboración del eportfolio como evidencia del proceso de aprendizaje y como recurso de evaluación del aprendizaje desarrollado [5]. Sin embargo, en relación con su impacto en los estudiantes, la función principal del eportfolio es mejorar el aprendizaje a través de la reflexión. Esta capacidad de reflexión sobre los resultados y el proceso de aprendizaje permite a los estudiantes tomar conciencia de su aprendizaje y hace que éste sea más significativo [7].

El diseño, implementación y uso del eportfolio es una realidad que beneficia el proceso de aprendizaje de los estudiantes universitarios, haciendo que éste se centre en el alumno y favoreciendo su autonomía, siendo su impacto inmediato y positivo cuando se combina con una estrategia de aprendizaje mixta [7]. Como también se apunta en [10], la aplicación de distintos métodos educativos combinados multiplica su efecto en la obtención de los resultados esperados.

4. Experiencia docente: reflexión sobre el proceso de aprendizaje

4.1. Planificación de la mejora

Para llevar a cabo el objetivo propuesto por la mejora, se pensó utilizar el eportfolio como una herramienta de reflexión en el proceso de aprendizaje de la asignatura. Por lo tanto, la finalidad en el diseño e implantación del eportfolio en la asignatura era:

- Por una parte proporcionar al alumnado un espacio en el cual pudiera almacenar, clasificar, seleccionar, reflexionar, etc. sus evidencias de aprendizaje, fomentando su autonomía y creatividad.
- Y por otra ofrecer al alumnado apoyo por parte del profesorado con el fin de guiar el proceso de aprendizaje a través de la evaluación del eportfolio.

Para conseguir estas dos finalidades se planteó la necesidad de llevar a cabo las siguientes acciones, cuya ejecución se detalla en la siguiente sección:

- Diseñar diferentes actividades para dar a conocer el uso del eportfolio como herramienta de aprendizaje.
- Incluir una actividad sobre el desarrollo del eportfolio, individual y grupal, por parte del alumnado dentro del contrato de aprendizaje de la asignatura, para que éste estuviera motivado en su realización.
- Realizar un eportfolio de la asignatura por parte del equipo docente de la asignatura, el cual serviría a la vez de ejemplo para el alumnado y como guía de su aprendizaje.

4.2. Ejecución de la mejora

La **primera tarea** realizada ha sido la modificación de la planificación de la asignatura para incluir una unidad didáctica sobre competencias transversales (ver Cuadro 1), en la cual se ha trabajado la forma de expresarse por escrito y oralmente de forma práctica a lo largo de cuatro horas divididas en dos sesiones. Es en estas sesiones donde se ha enseñado al alumnado a utilizar la plataforma tecnológica para la gestión de eportfolios usada en esta experiencia, el Mahara [1].

La **segunda** ha sido incluir en el contrato de aprendizaje una actividad que se ha de realizar durante todo el curso pero que se evalúa al final del mismo sobre la realización del eportfolio. La actividad consta

de dos partes, **un eportfolio individual** (4% de la nota final), en el cual se trata de presentar el resto de actividades y ejercicios que se realizan en clase de modo que se reflexione sobre la actividad y sobre lo aprendido en ella, o incluso sobre las dudas que plantea. Por ejemplo, en la realización de un diagrama de UML se pueden incluir la versión antes de la corrección y la posterior acompañada de la correspondiente reflexión sobre su utilidad, importancia, comprensión de errores, dudas, etc. Lo fundamental a la hora de confeccionar el eportfolio es mostrar cual ha sido el proceso de aprendizaje y no tanto si el resultado ha sido perfecto, por ello es importante la reflexión que el propio alumno hace sobre sus fallos, el por qué y de cómo ha aprendido. Y **otro eportfolio grupal** (8% de la nota final), en la cual la reflexión que se hace individualmente también hay que coordinarla y mostrarla para la parte práctica de la asignatura que se lleva a cabo en grupo. Para llevar a cabo ambas actividades los alumnos disponen en el LCMS de una guía de los contenidos mínimos que debe incluir el eportfolio (blog de la asignatura con al menos cinco entradas, reflexión sobre la corrección de un diagrama de UML, mapa conceptual, etc.), aunque se hace hincapié en que solo se trata de una ayuda y de que ésta es una actividad en la cual debe predominar su creatividad.

Por último, la **tercera** ha sido la realización del propio eportfolio de la asignatura por parte del equipo docente. En la Figura 1, se presenta un extracto del ejemplo que el equipo docente ha realizado para el presente curso. Este eportfolio tiene una doble finalidad, por una parte los alumnos deben crear su propio eportfolio como una de las actividades del contrato, por lo tanto éste les sirve a modo de ejemplo. Por otra, en él los alumnos pueden encontrar información más dinámica sobre la asignatura, complementaria a la que existe en la plataforma LCMS y reflexionar sobre los temas que propone el profesor.

Por lo tanto, el uso del eportfolio se ha realizado en la asignatura a tres niveles:

- **Del equipo docente:** en el desarrollo de este eportfolio ha participado todo el equipo docente. Puesto que el material de la asignatura ya está disponible en el LCMS, este espacio se ha dedicado a la reflexión de temas novedosos, de la práctica profesional relacionada con la asignatura, métodos avanzados en Ingeniería del Software, etc.
- **De cada grupo de alumnos:** donde los alumnos

ePortfolio II27 curs 2010/2011

Aquest és un ePortfolio per a l'assignatura II27 durant el Curs 2010/2011. En ell podreu trobar informació complementària a la que trobeu habitualment a l'Aula Virtual, on seguireu tenint la planificació de les sessions de teoria i pràctiques, així com tots els materials necessaris. En l'ePortfolio trobareu altres tipus d'informació com poden ser reflexions sobre la marxa de l'assignatura, enllaços Web, discussions sobre el cas pràctic, etc. A més a més, vos servirà d'exemple per tal de realitzar les diverses activitats proposades a l'AV.

Myself

Em diuen Reyes Grangel Seguer, valg nàixer a L'Alcora, i sóc professora del Dept. de Llenguatges i Sistemes Informàtics a l'UJI. Aquest curs 2010/2011 imparteix les assignatures: SIK003, IG22 i II27. També dirigisc projectes final de carrera (IG31 i II31). El meu horari de tutories és dilluns de 12 a 13 h, i dimecres de 12 a 13 h i de 15 a 16 h. També em pots preguntar en la 'Tutoria pública o privada' de l'AV.

• Adreça de correu: grangel@si.uji.es

Diari de classe

En aquest blog recolliré el que anem fent en el dia a dia de la classe, reflexions, impressions, etc. Ens servirà com a diari de la classe, és a dir, per tindre un recull del que hem fet en les classes basant-se en la planificació de les sessions que tinguem a l'AV. Podeu fer qualsevol comentari que cregueu oportú. Vos anima a que participeu, i moltes gràcies per anticipat. Ànim!! Reyes

Etiquetes: Metodologia Requisites An... Disseny Implementació

Com traure un 10 en l'Activitat 4?

La resposta seria obtenint un 'sí' en totes les preguntes de la llista de control. Però, que és la llista de control i on està? Anem per parts, la llista de control serien una sèrie de preguntes per comprovar si els requisits que es demanen en l'Activitat 4 estan inclosos o no en el treball que heu presentat. Aquestes preguntes s'han de respondre amb un sí o amb un no. I he dit serien, perquè en realitat la llista de preguntes no està escrita, està en el meu cap, però no l'he feta explícita. La idea d'una ensenyança de qualitat és que aquesta mena d'avaluació siga feta explícita a l'alumnat anteriorment a la pròpia avaluació i servisca per tant durant el desenvolupament per a fer un treball de més qualitat.

Dit tot açò, el que anem a fer es completar una llista de control per a l'Activitat 4, per a la qual vos proposa tres primeres preguntes a mode d'exemple. Després durant la classe elaborarem una rúbrica (com les que tinguem a la carpeta d'avaluació) que vos servisca de guia per al desenvolupament i avaluació de l'activitat.

Afegiu les preguntes seguint la numeració i contestant a aquest post, recordeu que les preguntes s'han de poder contestar amb un sí o un no.

Grady Booch

Material

Continguts:	
Nom	Descripció
II271011_...ctlc.pdf	
II271011_...acte.pdf	

Procés de desenvolupament de programari ...

Figura 1: Extracto del eportfolio de ejemplo de la asignatura creado por el equipo docente.

que forman los grupos de trabajo de la asignatura (tres en cada grupo) han podido cooperar para llevar a cabo el proyecto práctico de la asignatura y reflexionar de forma conjunta sobre su proyecto (ver Figura 2).

- **De cada alumno:** donde cada alumno ha reflexionado sobre su aprendizaje en la asignatura, como se ha realizado este proceso, como se han llevado a cabo los progresos, las dudas surgidas, etc.

4.3. Evaluación de la mejora

La evaluación y seguimiento de la mejora se ha llevado a cabo a través de dos mecanismos. El **primero** mediante el feedback de los alumnos. Éste se recoge a través de un cuestionario diana muy sencillo (ver Figura 3), en el cual los alumnos sombreen en la diana su valoración sobre distintos aspectos de la asignatura en general, siendo el centro la puntuación más baja y los extremos la más alta. En el caso de que la puntuación sea baja los alumnos tienen espacio adicional en el cuestionario para indicar los puntos débiles y las propuestas de mejora para el aspecto valorado negativamente. Con el pase de tres cuestionarios a lo largo del curso se puede recoger el feedback de los alumnos de forma que es incluso

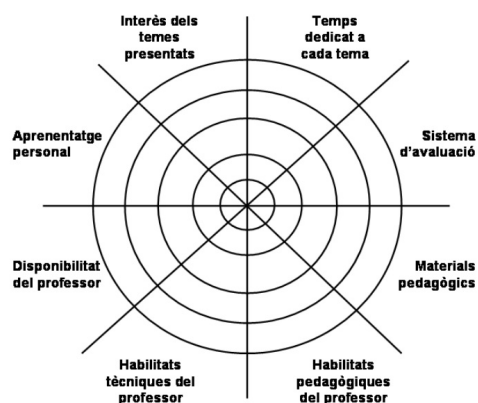


Figura 3: Cuestionario diana usado para recopilar el feedback sobre la asignatura.

posible introducir alguna mejora o renegociar el contrato para el segundo semestre si la valoración lo recomienda.

El **segundo** mecanismo usado para concluir hasta qué punto la técnica del eportfolio ha servido para conseguir el objetivo propuesto, están aprendiendo más

En esta vista se recoge el trabajo realizado por el grupo 6 de Ingeniería del Software I durante el segundo semestre del curso 2010/2011

Breve historia del grupo 6

El grupo 6, formado por Javier Vilar Sánchez, Rubén Martínez Villar y Sandra Catalán Pallarés, surgió el curso 2009/2010 en la asignatura de Algoritmica, mientras uno de los futuros miembros (por aquel entonces) estaba de viaje en Japón. Desde ese momento hemos permanecido como grupo de trabajo en todas aquellas asignaturas que han requerido grupos de 3 personas, lo que nos ha permitido realizar los trabajos eficientemente gracias al conocimiento que hemos adquirido los unos de los otros durante este tiempo.

Proyecto final

A continuación se presenta la memoria del trabajo realizado para el desarrollo del sistema informático de la empresa Estudios y Proyectos Metálicos. El sistema a desarrollar informatiza la gestión de gastos de los viajes realizados por los montadores de la empresa y la planificación de los mismos. En esta memoria se recoge el desarrollo del prototipo utilizando una metodología estructurada.

Tras la memoria, se presentan dos modelos que contienen el trabajo realizado para la definición y análisis del sistema informático de la empresa utilizando el paradigma orientado a objetos (a través de UML). Los modelos que se presentan contienen los requisitos funcionales que el sistema debe cumplir (modelo de casos de uso), así como el comportamiento que debe tener y la estructura de la información del dominio (modelo de análisis).

Informe del proyecto

[informe.pdf](#)
Informe del proyecto "Estudios y Proyectos Metálicos"
852.2KB | Monday, 21 March 2011 | [Details](#)

Modelo de casos de uso

Esta carpeta contiene el diagrama de casos de uso y las plantillas que los describen

Continguts:

Nom	Descripció
CU1	Plantilla del caso de uso "Registrar información de los viajes"

Figura 2: Extracto de un eportfolio creado por un grupo de alumnos para la asignatura.

los alumnos con la aplicación de nuevos métodos pedagógicos, ha sido analizar por parte del profesorado los eportfolios realizados por los alumnos, tanto de forma individual como grupal, para determinar el progreso en su proceso de aprendizaje. Este análisis se realiza mediante la rúbrica general generada en la asignatura para evaluar trabajos escritos, la cual incluye como elementos a evaluar: la estructura y el formato, la corrección, el estilo, el contenido y la ampliación de objetivos.

Para concluir cabe señalar algunas **lecciones aprendidas** sobre el uso del eportfolio como herramienta de reflexión del proceso de aprendizaje que pueden ser vistas como mejoras para próximos cursos:

- El desarrollo del eportfolio debe ser una actividad valorada en la nota final para que los alumnos perciban su utilidad y se sientan motivados a realizarla.
- Se deben marcar entregas parciales del eportfolio que deben ir acompañadas del correspondiente feedback del profesor, de forma que el resultado final sea fruto de un proceso de maduración a lo largo del tiempo.
- No importa tanto la cantidad de evidencias presentadas como su calidad. Los trabajos de la

asignatura se pueden entregar en el LCMS, el eportfolio solo debe incluir aquellas aportaciones que el alumno crea significativa en su proceso de aprendizaje.

- El eportfolio es una herramienta útil no solo para reflexionar sobre el propio aprendizaje sino también sobre la planificación y marcha de la asignatura.

5. Conclusiones

La primera conclusión que se puede extraer de la experiencia es que en ocasiones es positivo desarrollar completamente la misma tarea que se les pide a los alumnos, cosa que se ha podido experimentar con el desarrollo del eportfolio del equipo docente, y que no sucede en el caso de trabajos escritos. De esta forma el profesor es consciente del trabajo que pide a sus alumnos y que realmente la adopción de un nuevo método de enseñanza-aprendizaje no sobrepasa las horas de dedicación del antiguo. Puesto que en cualquier caso el tiempo del alumnado y el del profesorado es limitado.

En referencia al trabajo del alumnado cabe destacar que es interesante ver como alumnos que con las actividades más tradicionales no se muestran cuidadosos en su presentación o trabajo, y poco

implicados en clase, demuestran mayor madurez a la hora de desarrollar sus eportfolios. En cualquier caso también se da el caso contrario, en el cual a otros alumnos con buenas actividades en 'privado' les cuesta más expresarse o reflexionar sobre su aprendizaje de una forma más pública. Esta variedad en la respuesta del alumnado se observa también cuando en un examen hay preguntas tipo test, de respuesta corta, de razonar, etc.s, hay alumnos que tienen más desarrolladas las competencias para responder a un tipo u otro de preguntas. Por lo tanto, es importante diseñar la asignatura para que esas distintas competencias estén equilibradas y se pueda dar al alumnado un feedback sobre aquellas en las que necesita mejorar.

Por último, y para responder a la pregunta planteada al inicio del artículo se ha podido comprobar en la mayoría de casos la evolución del aprendizaje de los alumnos con el seguimiento de sus eportfolios, de forma que existe diferencia entre sus comentarios o reflexiones al inicio y sus últimas aportaciones. En cualquier caso es difícil concluir que han aprendido más, lo que sí se ha podido observar es que aprenden de forma diferente y 'cosas' diferentes.

Referencias

- [1] Mahara website. <http://mahara.org/>, mayo 2011.
- [2] Moodle website. <http://moodle.org/>, mayo 2011.
- [3] F.J. García Bacete and R. Grangel Seguer. Aplicació dels contractes d'aprenentatge en l'assignatura 'Aplicacions per a la gestió de l'ETIG'. In *Formació del professorat davant la convergència europea. Actes de la V Jornada de millora educativa*, 2005.
- [4] H.C. Barret. Differentiating electronic portfolios and online assessment management systems. In *Proceedings of SITE 2004*, 2004.
- [5] M. de Miguel Díaz (coord.). *Metodologías de enseñanza y aprendizaje para el desarrollo de competencias: orientación para el profesorado universitario ante el EEES*. Alianza, cop., 2006.
- [6] R. Grangel, C. Campos, V. Verde, and C. Rebollo. Aprender a aprender estudiando la Ingeniería del Software. In *Actas de las XVI JENUI (2010)*, pages 197–204, 2010.
- [7] O. Lopez-Fernandez and J. L. Rodriguez-Illera. Investigating university students' adaptation to a digital learner course portfolio. *Computers & Education*, 52(3):608–616, 2009.
- [8] E. Meyer, P.C. Abrami, C.A. Wade, O. Aslan, and L. Deault. Improving literacy and metacognition with electronic portfolios: Teaching and learning with ePEARL. *Computers & Education*, 2009.
- [9] Benelux Bologna Secretariat. The official Bologna Process website 2007-2010. <http://www.ond.vlaanderen.be/hogeronderwijs/bologna/>, mayo 2011.
- [10] M. Valero-García and J.J. Navarro Guerrero. Preguntas frecuentes y nuestras respuestas favoritas sobre la pertinencia de los métodos docentes centrados en el estudiante para adaptar una asignatura al EEES. *Novática*, (192):48–50, 2008.

Uso de metodologías activas en la implantación de IIP en el Grado en Informática de la UPV

Nati Prieto, Marisa Llorens, Germán Moltó, Jon Ander Gómez, Mabel Galiano, Carlos Herrero

Departament de Sistemes Informàtics i Computació

Escola Tècnica Superior d'Enginyeria Informàtica

Universitat Politècnica de València

Camí de Vera S/N – 46022 València

{nprieto,mlllorens,gmolto,jon,mgaliano,cherrero}@dsic.upv.es

Resumen

Este artículo describe la experiencia de implantación de la asignatura Introducción a la Informática y la Programación (IIP) de primer curso del Grado en Informática en la Escuela Técnica Superior de Ingeniería Informática (ETSINF) de la Universitat Politècnica de València (UPV), destacando el uso de metodologías activas de enseñanza-aprendizaje que incorporan el trabajo en grupo, el diseño de un método de evaluación acorde con la metodología empleada y la incorporación de herramientas tecnológicas de soporte a la docencia. Adicionalmente, se describen y evidencian los aspectos positivos y negativos de la experiencia tanto desde el punto de vista del profesor como del alumno.

Summary

This paper describes the experience of setting up the IIP subject (Introduction to Computer Science and Programming) to new course degrees of the first course at School of Computer Science (ETSINF) in the Universitat Politècnica de València (UPV), pointing out the usage of active learning methodologies based on work group, the design of an evaluation method considering the applied methodology and the integration of technological tools to support teaching. Additionally, the positive and negative aspects of the experience are discussed both from the point of view of the teacher and the student.

Palabras clave

EEES, grado, programación, metodologías activas, evaluación continua.

1 Introducción

El curso 2010/2011 se pone en marcha el título de Grado en Informática (GI) en la Universitat Politècnica de València (UPV) en el marco normativo que viene definido por el real Decreto 1393/2007 [12], el documento publicado por el Ministerio de Educación y Ciencia en 2006 referente a la renovación de las metodologías educativas [11] y, ya en la UPV, por la nueva normativa de régimen académico y de evaluación del alumnado [10] que, entre otros aspectos, promueve cambios dirigidos a:

- Impulsar la implantación de nuevas metodologías docentes, que hagan al estudiante parte activa del proceso de enseñanza-aprendizaje.
- Definir un nuevo marco que facilite la implantación de un sistema de evaluación continua, que determine el grado de alcance de los objetivos de aprendizaje mediante las actividades docentes en el aula y fuera de ella, restando importancia a los exámenes finales.
- Utilizar las tecnologías de la información en los procesos de evaluación, calificación, tutorización, etc.

El diseño de la asignatura Introducción a la Informática y la Programación (IIP) se ha realizado respetando este contexto y siguiendo algunas de las pautas que se indicaron en el taller “Adaptación de asignaturas al EEES”, impartido por los profesores Miguel Valero y Margarida Espona en nuestra universidad en el año 2010 [8]. Se trataba de definir los objetivos formativos de la asignatura, clasificarlos en tres niveles (conocimiento, comprensión y aplicación) y planificar las actividades a realizar en el aula y también las no presenciales para incorporar al alumno como elemento central del proceso de enseñanza-aprendizaje; además, y no menos importante, se debía definir un método de evaluación continua acorde con la metodología propuesta. Este pro-

ceso de diseño, así como su implementación posterior, ha estado marcado básicamente por dos factores connaturales a la asignatura, una de las cinco básicas de 6 créditos que se cursa en el primer semestre del primer curso de la titulación de GI: la coordinación de un elevado número de profesores y el número de alumnos tutelados por cada uno de ellos. En concreto, los datos de la asignatura son: 532 alumnos matriculados que se organizan en 11 grupos de 50 aproximadamente que, a su vez, se dividen en dos para la realización de las prácticas y 14 profesores utilizando, según el grupo, algunas de estas lenguas: castellano, inglés y valenciano. Además, cabe señalar que la procedencia del alumnado es muy diversa: casi un 45% de bachiller (principalmente de la modalidad científico-técnica, aunque no exclusivamente), un 30% de ciclos formativos (principalmente de los relacionados con la informática, aunque no exclusivamente), titulados y otros en un bajo porcentaje y, finalmente, un 25% son alumnos adaptados de los planes de estudios de Ingeniería Superior y Técnica.

Por tanto, la coordinación del profesorado no ha podido ser realizada más que consensuando los siguientes puntos: la definición de objetivos formativos; la planificación docente, esto es, la estructuración en temas y distribución en sesiones de teoría y laboratorio siguiendo el calendario aprobado por el centro; el material común a desarrollar y distribución de tareas entre los profesores y, no menos importante, la definición de las normas de evaluación comunes que establecen tanto los actos de evaluación como sus pesos: exámenes comunes escritos, pruebas comunes para la evaluación de las prácticas de laboratorio y actividades de seguimiento de las clases a definir por cada profesor.

Nótese que este marco de actuación es perfectamente compatible con las normativas a las que hacemos referencia al comienzo y permite que los profesores se sientan cómodos definiendo las actividades presenciales y no presenciales según sus gustos, intereses, preparación, etc. En concreto, los autores de este artículo prepararon de forma coordinada las diferentes sesiones incorporando técnicas de aprendizaje cooperativo, metodologías de evaluación continua acordes y la tecnología a su alcance tanto para el (auto)aprendizaje del alumno como para su evaluación.

Para explicar estas incorporaciones, en la sección 2 de este artículo se realiza una breve descripción de la asignatura para dar a conocer el marco de actuación. En la sección 3 se describen las técnicas y metodologías empleadas para dar soporte al aprendizaje cooperativo dentro de este marco, así como los mecanismos de autoevaluación y evaluación del alumno utilizados. Finalmente, en las secciones 4 y 5 se presentan, respectivamente, los resultados obtenidos, analizándose en base a las opiniones de los profesores y alumnos involucrados, y se destacan las líneas de actuación en las que se está pensando para mejorar la experiencia el próximo curso.

2 La asignatura IIP: ordenación docente y sistema de evaluación

La asignatura IIP tiene como objetivo iniciar al alumno en el proceso de construcción y estudio de los programas. En concreto, se pretende que el alumno sea capaz de escribir programas correctos de baja complejidad en un lenguaje imperativo de alto nivel, utilizando aspectos básicos de programación orientada a objetos. Los 6 ECTS que tiene asignados están divididos en 4.5 de Teoría de Aula y Seminario (TAS) y 1.5 de Prácticas de Laboratorio (PL). La docencia presencial se organiza en dos sesiones en el aula de 1.5 horas por semana (15 semanas) y 1 sesión en el laboratorio de 1.5 horas por semana (10 semanas). Los contenidos de la asignatura y su distribución en horas de trabajo presencial y no presencial se pueden consultar en la guía docente de la asignatura [9].

Para la calificación final del alumno se tienen en cuenta tres notas: la primera es la nota de parciales ponderada (NPP) y se obtiene de dos pruebas escritas obligatorias con un peso del 60%; la segunda es la nota de las prácticas de laboratorio (NPL), cuyo peso es un 20%, y la tercera es la nota de actividades de seguimiento (NAS) con un valor del 20%. Los detalles del cálculo de la nota final de IIP también se pueden consultar en [9].

Las dos pruebas escritas constan de una serie de cuestiones y problemas de respuesta abierta que son comunes para todos los grupos y que se realizan en fechas planificadas por el centro.

Para la evaluación de las prácticas de laboratorio se han realizado 7 actos de evaluación comunes a

todos los grupos y se han utilizado distintas estrategias: test de corrección automática en PoliformaT [3], uso del portal de autocorrección [4], evaluación y prueba de código por parte del profesor en el laboratorio y realización de una memoria.

El 20% de la nota final correspondiente a la calificación de las actividades de seguimiento se ha dejado a criterio de cada profesor. No obstante, los autores, de forma coordinada y basándose en las pautas dadas en [8], han planificado las actividades presenciales y no presenciales y su evaluación para incorporar al alumno como centro del proceso de enseñanza-aprendizaje, tal y como se describe en la sección 3.

3 Metodologías empleadas para estructurar el aprendizaje cooperativo y su evaluación

De lo dicho hasta ahora resulta claro que la estructura de IIP, respetando el nuevo marco de evaluación continua, no supone de per-se la implantación de nuevas metodologías docentes; también resulta evidente que las actividades de seguimiento y su peso del 20% en la asignatura son el único punto de su diseño donde tales metodologías se pueden incluir. Así pues, teniendo como objetivo básico fomentar el aprendizaje cooperativo, los autores de este artículo decidieron realizar una propuesta conjunta de actividades de seguimiento y de su evaluación basada en la especificación detallada de las actividades que los alumnos debían realizar dentro y fuera de la clase y el modo en que éstas serían evaluadas para contribuir a su NAS.

Nuestro interés por el aprendizaje cooperativo se asienta en dos hechos: por una parte, las técnicas que lo estructuran permiten establecer qué deben hacer los alumnos organizados por grupos dentro y fuera del aula y, por tanto, deben ser contempladas por un programa que, como el presentado, pretende centrarse en el aprendizaje del alumno; por otra, estas técnicas tienen ventajas de sobra conocidas que nos interesaba aprovechar [1, 5, 6]. A saber: reducir el problema del absentismo, fuertemente implantado en nuestro entorno, incrementar la motivación personal del alumno haciendo uso de las explicaciones que da o le dan otros para entender mejor la materia y, finalmente, transmitir de forma efectiva al

alumno cuáles son los conocimientos mínimos que se le van a exigir para superar la asignatura con éxito para que se conciencie a tiempo de cuáles son sus problemas de aprendizaje y de cómo puede resolverlos estableciendo y manteniendo con otros una interdependencia positiva. También se era consciente de que, asociados a estas ventajas, podían aparecer los inconvenientes que todo el mundo destaca. Por un lado la falta de compromiso de algunos estudiantes que, en nuestro caso, se ha visto potenciada por un número elevado de alumnos de nivel homogéneo bajo la tutela de un mismo profesor; por otro, la escasa o nula experiencia en la cooperación en el aula, tanto de los profesores como de los alumnos de “primero de primero”, lo que supone redoblar esfuerzos y construir cada actividad sobre la valoración coordinada del resultado de la anterior hasta volver a establecer una línea metodológica en la que se sigan viendo implicados los profesores y alumnos participantes.

A continuación se describen los elementos que han permitido desarrollar nuestra propuesta de actividades de seguimiento y su evaluación mediante entregables individuales o en grupo, que representan los resultados tangibles del trabajo del alumno y nos permiten observar y cuantificar tanto su evolución individual como la que resulta de su pertenencia a un grupo de trabajo y un grupo de clase:

- Selección de tipos de actividades a realizar y metodologías empleadas en ellas para estructurar el aprendizaje cooperativo.
- Guías didácticas con la planificación detallada de todas las actividades y entregas a realizar en cada sesión con una indicación del tiempo que deben dedicar a cada actividad.
- Herramientas tecnológicas para facilitar la implementación de las actividades de seguimiento y su evaluación, i.e. un portal Web para la autocorrección de programas de ordenador [4] al estilo de un oráculo y la plataforma de teleformación de la UPV PoliformaT [3].

3.1 Actividades para el aprendizaje cooperativo: tipos, metodologías y evaluación

Las actividades asociadas a cada tema son bastante heterogéneas tanto por su tipo como por el momento y lugar en el que se realizan. Básicamente se han planteado dos tipos:

- las que forman parte del trabajo personal semanal que ha de realizar el alumno, que se podrían plantear en cualquier grupo de la asignatura como actividades de seguimiento, tanto si son individuales como en grupo y presenciales o no presenciales (pequeñas pruebas escritas de evaluación, test de autoevaluación, trabajos de búsqueda y elaboración de documentación, ejercicios de programación a evaluar automáticamente por el Portal, etc.); y
- las que se diseñan con el objetivo de favorecer el aprendizaje cooperativo y evaluarlo; éstas pueden ser individuales o en grupo, presenciales o no.

En cualquier caso, el segundo tipo de actividades se pueden considerar variantes más o menos estrictas de un puzle de Aronson [1, 5]. Ello implica, obviamente, dividir cada clase al principio de curso en grupos de aprendizaje cooperativo (o grupos base), cada uno formado por tres alumnos como máximo; en nuestro caso una media de 16 grupos base en cada grupo de la asignatura. A partir de aquí, la realización de cada una de estas actividades supone:

- asignar a cada miembro de un grupo base la resolución de una parte del trabajo bien diferenciada de las restantes;
- bien fuera del horario lectivo o bien durante una sesión de teoría, hacer una puesta en común del trabajo realizado con el resto de compañeros (del grupo base o de todos los grupos);
- elaborar un único entregable por grupo como conjunción de las diferentes soluciones a cada parte del trabajo;
- evaluar en base a dicho entregable conjunto y, en ocasiones, a los entregables individuales que cada miembro de un grupo ha tenido que realizar antes de éste y gracias a los cuales el profesor controla su actividad en el grupo. Por supuesto, esta evaluación debe respetar siempre el criterio fundamental que hace que el aprendizaje cooperativo no sea exactamente trabajo en grupo: los resultados individuales dependen de los del grupo y viceversa y sólo es posible obtener la puntuación máxima si cada miembro del grupo aporta su parte no sólo en beneficio propio sino también en beneficio común.

Sobre estos trabajos-puzle podemos apuntar también que la característica que tal vez los distingue de

otros planteados en experiencias del género [1, 5, 6] es que cada uno de ellos no se prolonga demasiado en el tiempo, sino que se dedica a trabajar de forma intensiva un aspecto fundamental de la tarea de programar. Aunque ello puede restar flexibilidad a la técnica y a la forma de afrontar la diversidad en el aula, creemos que fomenta la asistencia a clase, condición que resulta imprescindible para la evaluación del trabajo, y es una forma efectiva de transmitir al alumno qué conceptos y habilidades tiene que adquirir para poder iniciarse en la construcción y análisis de programas en un lenguaje imperativo de alto nivel.

3.2 Guías Didácticas

Como ya se ha comentado, para cada tema se ha elaborado una Guía Didáctica que supone una referencia tanto para el alumno como para el profesor y contiene los siguientes ítems:

1. Índice de contenidos del tema.
2. Objetivos formativos, clasificados según los tres primeros niveles de la taxonomía de Bloom [2, 7].
3. Bibliografía detallada para cada apartado del tema.
4. Planificación temporizada de cada sesión que explicita el trabajo a realizar por el alumno previo a la sesión de clase, el trabajo del profesor durante la sesión, y las actividades que realizarán los alumnos en clase.
5. Tabla resumen con todas las entregas a realizar en el tema: de qué material se dispone, qué hay que entregar, cómo, cuándo y su puntuación; así mismo, una explicación de cómo se evalúan las entregas a realizar para formar parte de la NAS.

3.3 Portal de autocorrección

En la línea de incorporar herramientas tecnológicas de apoyo al autoaprendizaje y la evaluación de los alumnos, se ha trabajado con un portal Web para la autocorrección de programas de ordenador al estilo de un oráculo [4]. Los enunciados de los problemas están disponibles en el portal agrupados por temas. Tras ser compilado, cada programa debe superar una batería de pruebas y, tanto en caso de éxito como de fallo, el alumno recibe inmediatamente la respuesta a cada intento de solucionar un problema. Todos los

intentos que envían los alumnos se registran en una base de datos para que éstos puedan consultar sus estadísticas particulares.

Este sistema se puede utilizar de manera autónoma y con libertad de horario, lo que ha hecho posible su uso como herramienta de autoaprendizaje por parte de cualquier alumno de la asignatura IIP, como herramienta de corrección automática de algunas prácticas de la asignatura y, finalmente, y sólo para los alumnos de los grupos con aprendizaje cooperativo, como un instrumento para realizar una actividad de seguimiento por tema (20 puntos por dos problemas resueltos en el portal).

La Tabla 1 muestra la media de envíos al portal de autocorrección por alumno y por problema, aceptados y no aceptados, para un total de 458 alumnos y 53 problemas. El número total de envíos aceptados ha sido de 4385 y el de no aceptados de 11006. Nótese que el número medio de intentos para la resolución de un problema es aproximadamente 2.5 por alumno.

Envíos		Media
por alumno	aceptados	9.57
	no aceptados	24.40
por problema	aceptados	82.73
	no aceptados	215.80

Tabla 1: Envíos por alumno y por problema al portal de autocorrección.

3.4 PoliformaT

Una de las herramientas tecnológicas que ha facilitado la gestión de las actividades de seguimiento y la evaluación de alguna de ellas ha sido PoliformaT [3], la plataforma de teleformación de la UPV. En concreto, sus herramientas Exámenes y Tareas han servido respectivamente para dar soporte a actividades consistentes en tests de corrección automática (también para la evaluación de algunas prácticas) y para la gestión y corrección de los entregables asociados a las actividades de seguimiento (los alumnos envían sus entregas a un repositorio centralizado donde el profesor puede ir corrigiendo e informando al alumno o grupo acerca de este proceso). Otras herramientas que han sido útiles para el

funcionamiento general de la asignatura han sido el Correo Interno, para poder intercambiar mensajes a los alumnos; el Foro y el Chat, a modo de canales de comunicación entre los participantes del sitio, donde se presentan y resuelven dudas acerca de algún aspecto que atañe al funcionamiento de la asignatura o sus temas y Recursos, que permite construir una jerarquía de carpetas donde albergar no sólo el material común a toda la asignatura sino también el complementario que cada profesor pudiera considerar apropiado para sus alumnos.

Aunque el potencial de esta herramienta es muy grande, el uso intensivo que se le ha dado ha puesto de manifiesto, también, algunas deficiencias. Quizás las más importantes que se han detectado son: no dispone de una herramienta para la gestión de grupos de trabajo cooperativo y, si el número de alumnos y profesores del sitio es elevado, queda patente que no está bien dimensionada y presenta algunos problemas de fiabilidad y seguridad, lo que ha frustrado muchas de las expectativas que se habían puesto en ella para este curso.

4 Resultados de la experiencia de aprendizaje cooperativo

La experiencia aquí descrita afectó a 314 alumnos distribuidos en 6 grupos, un 59% de los alumnos matriculados en la asignatura (532); estos grupos no fueron seleccionados a priori, sus alumnos no fueron informados de ello en el momento de la matrícula, por lo que el hecho de que fueran todos grupos de mañana es mera coincidencia. Se han utilizado los indicadores de rendimiento de estos alumnos, su opinión a través de las encuestas que se les hicieron y el proceso de reflexión llevado a cabo por sus profesores para extraer los resultados y conclusiones que figuran a continuación.

4.1 Resultados de rendimiento

La Tabla 2 refleja los indicadores de rendimiento de los alumnos de los diferentes grupos. Sus seis primeras filas corresponden a los datos de los grupos donde se aplicaron técnicas de aprendizaje cooperativo, porcentaje de presentados, de aprobados sobre presentados y matriculados y sus notas totales, de actividades de seguimiento (NAS sobre 2 puntos), prácticas de laboratorio (NPL sobre 2) y nota

de parciales ponderada (NPP sobre 6). A efectos de comparación también se muestran en la tabla estos mismos datos a partir de la fila 8 para el resto de grupos de la asignatura; es necesario destacar que entre estos grupos figura el grupo 1E de alto rendimiento (impartido en inglés) y el grupo 1D que han tenido unos resultados extraordinarios, no sólo en IIP sino también en el resto de asignaturas. Como puede verse, las diferencias de cifras entre las columnas “Total AC” y “Todos” es nula o insignificante: 79% de tasas promedio de presentados frente a 77%, 49 y 49% de aprobados sobre matriculados, 63 vs 64% sobre presentados, nota media de actas 5.43 vs 5.56. Sí se observa en cambio que, independientemente de que el grupo esté por encima o por debajo de la fila 6 (“Total AC”) las variaciones de todos los indicadores son muy grandes entre algunos grupos; el grupo 1C y el grupo 1D suelen marcar las mejores cifras mientras que los grupos 1B y 1K tienen las peores. La única columna de cifras que se libra de fuertes variaciones entre grupos es NPL, probablemente debido al uso de criterios comunes en la corrección de las prácticas.

De todas estas cifras deducimos dos cosas:

- el uso de criterios comunes de evaluación en todas las actividades puede tener más importancia sobre las calificaciones que los planteamientos que se sigan a la hora de proponerlas; y
- que el nivel de conocimientos de los alumnos sea tan homogéneo en cada grupo (bueno, malo, medio) impide hacer alguna valoración sobre el resultado de utilizar o no aprendizaje cooperativo.

4.2 Opinión de los alumnos

Durante la última semana del curso, se propuso a los alumnos que rellenaran una encuesta a través de *Google Docs* para valorar su percepción de la asignatura. El diseño de la encuesta fue el propuesto por Valero en [8]. De los 314 alumnos implicados, sólo contestaron 167 que suponen un 53,2%. Cabe destacar por tanto que la participación de los alumnos en esta encuesta no ha estado al mismo nivel que en el resto de actividades propuestas a lo largo del curso y de cara a sucesivas experiencias será necesario replantear las fechas de pase de la encuesta y el énfasis del profesor por concienciar a los alumnos

	Presentados			Aprobados			Acta	NAS	NFL	NPP
	Matr.	Total	%	Total	% Pres.	% Matr.	Media	Media	Media	Media
Gr. 1A	56	43	77%	23	53%	41%	5,19	1,23	1,32	2,70
Gr. 1B	47	35	74%	15	43%	32%	4,30	0,88	1,06	2,41
Gr. 1C	49	45	92%	36	80%	73%	6,72	1,37	1,50	3,92
Gr. 1F	54	41	76%	28	68%	52%	5,52	0,92	1,29	3,31
Gr. 1G	54	39	72%	26	67%	48%	5,08	0,96	1,26	2,72
Gr. 1H	54	45	83%	27	60%	50%	5,10	1,10	1,31	2,67
Total AC	314	248	79%	155	63%	49%	5,43	1,11	1,31	3,02
Gr. 1D	54	45	83%	37	82%	69%	7,00	1,60	1,69	3,64
Gr. 1E	31	26	84%	17	65%	55%	6,58	1,16	1,55	3,91
Gr. 1I	29	26	90%	16	62%	55%	5,15	0,85	1,25	3,03
Gr. 1J	53	35	66%	21	60%	40%	5,45	1,21	1,40	2,84
Gr. 1K	51	30	59%	17	57%	33%	4,67	0,96	1,11	2,60
Todos	532	410	77%	263	64%	49%	5,56	1,14	1,35	3,07

Tabla 2: Indicadores de rendimiento de los alumnos por grupos.

de la importancia que tiene su opinión en el proceso de mejora del diseño de la asignatura.

Pregunta	75%	25%
He aprendido cosas valiosas para mi formación	73.6	26.3
El profesor me ha facilitado el aprendizaje	56.3	43.7
El material está bien preparado y es adecuado	46.1	53.9
Siempre he tenido claro lo que tenía que hacer	34.1	65.9
Siempre he conocido mi progreso en el curso	61.7	38.3
El trabajo en grupo me ha resultado de gran ayuda	50.3	49.7
La forma de evaluación me ha parecido adecuada	51.5	48.5
Me ha ayudado a mejorar la gestión de mi propio tiempo	32.9	67.1

Tabla 3: Encuesta de opinión al alumnado. Se mide el porcentaje de acuerdo y desacuerdo con la afirmación de la primera columna.

En cuanto a los resultados que muestra la Tabla 3, que incluye también las preguntas de la encuesta, se pueden destacar como positivos que, mayoritariamente, los alumnos han opinado que sí han aprendido cosas valiosas para su formación y que siempre se han sentido bien informados sobre su progreso (o falta de él). Pero, también resulta obvio a partir de esta tabla, que hay que hacer un mayor esfuerzo para

que el alumno tenga claro qué es lo que tiene que hacer (tanto en clase como fuera de clase) y que es necesario mejorar el material de la asignatura.

Los resultados de las dos últimas preguntas no permiten obtener ninguna conclusión porque tienen diferentes interpretaciones: no se puede deducir si el desacuerdo en la forma de evaluación es por algún aspecto particular de las normas de evaluación comunes (parciales, prácticas y notas mínimas exigidas) o por la forma de evaluación de las actividades de seguimiento que se han propuesto en estos grupos. Asimismo, tampoco se puede deducir si no se les ha ayudado a mejorar la gestión que hacen de su propio tiempo porque la planificación semanal realizada no les había sido útil o porque era imposible seguirla debido a la sobrecarga de trabajo en la misma y otras asignaturas del semestre.

4.3 Opinión del profesorado

Como resultado de la reflexión del profesorado con respecto al trabajo que debe realizar el alumno en la asignatura, se consideran como aspectos positivos los siguientes:

- Favorece la actitud positiva de aprendizaje del alumno: motiva al alumno que trabaja dentro y fuera de clase, llevando al día la asignatura.
- Informa al alumno de su progreso, lo que a su vez lo estimula.
- Permite establecer una relación profesor-alumno objetiva y, en muchos casos, positiva.
- Permite el uso de la evaluación como forma de aprendizaje.
- El profesor dispone de la información necesaria para incidir en el proceso de aprendizaje del alumno a lo largo del semestre, así como una evaluación más objetiva y completa.
- Un elevado número de alumnos han seguido la planificación propuesta.

Sin embargo, también se señalan los siguientes aspectos negativos:

- Puede generar demasiada presión y, a veces, frustración sobre alumnos y profesores, teniendo en cuenta que existen más asignaturas para todos.
- Supone una sobrecarga para el profesor por la novedad, pero también por la falta de adecuación del entorno académico a la reforma.

- Son más visibles ahora los desajustes de coordinación intra/entre asignaturas, no sólo a nivel de contenidos sino también de metodologías a emplear y planificación de las actividades que éstas conllevan.
- Ha habido diferencias sustanciales en el nivel de seguimiento de los alumnos en los distintos grupos.
- El nivel de seguridad para evaluar al alumno no es óptimo.

También se han producido algunas circunstancias que han afectado de forma negativa a la marcha del semestre:

- La matrícula no es definitiva al comienzo de las clases, lo que ahora sí es realmente significativo.
- La distribución de alumnos en base a sus conocimientos no es uniforme, lo que supone un sesgo importante a la hora de extraer conclusiones generales.
- La distribución de alumnos de un grupo en dos horarios de prácticas no es equilibrada.
- Las características y dotación de las aulas y laboratorios y la programación de los exámenes debe mejorar, porque ahora resulta imprescindible.

5 Conclusiones y prospectiva

En este artículo se ha presentado la apuesta que han realizado los autores para incluir actividades que favorezcan el aprendizaje cooperativo en la asignatura de grado IIP. Si bien el trabajo de equipo del profesorado ha resultado muy interesante y positivo, los resultados de rendimiento presentados indican claramente que no se puede evaluar la repercusión que la introducción de estas metodologías tiene si no se cumplen unos requisitos mínimos; léase:

- una distribución de alumnos más heterogénea en base a sus conocimientos –recordemos que es justo lo contrario– lo que lleva a diferencias porcentuales en las tasas de rendimiento de los diferentes grupos de hasta un 40%;
- una definición y evaluación compartida por todos los grupos de las actividades de seguimiento –recuérdese que uno de los consensos adoptados para poner en marcha la asignatura

natura era que cada uno de los once profesores de teoría debía sentirse cómodo definiendo las actividades presenciales y no presenciales y su evaluación según sus gustos, intereses, preparación, etc.—.

Estos dos requisitos son tan prioritarios que aspectos menores, como el hecho de que la matrícula no haya sido definitiva al comienzo de las clases o la distribución desequilibrada de los alumnos en los subgrupos de prácticas, etc., resultan por el momento irrelevantes.

A pesar de lo dicho, no nos cuestionamos si ha valido la pena la experiencia de introducir el aprendizaje cooperativo, ni tampoco que el estudiante debe ser parte activa del proceso de enseñanza-aprendizaje. No hay duda que se deben utilizar las tecnologías de la información en los procesos de calificación, tutorización y evaluación del alumnado. Nuestra perspectiva es ahora analizar y ser capaces de elegir entre el amplio abanico de metodologías docentes aquéllas que, dado el contexto en el que se desarrolla la asignatura, sean efectivas.

Referencias

- [1] Anguas, J., Díaz, L., Gallego, I., Lavado, C., Reyes, A., Rodríguez, E., Sanjeevan, K., Santamaría, E. y Valero, M. *La técnica del Puzzle al servicio del aprendizaje de la programación de ordenadores*. En XII Jornadas de Enseñanza Universitaria de la Informática (JENUI 2006), pp. 477 – 484, 2006.
- [2] Bloom, B. S. *Taxonomy of Educational Objectives: The Classification of Educational Goals*. En Proceedings of the Fourth European Conference on Information Systems, pp. 201 – 207, 1956.
- [3] Busquets, J., Roldán, D., Martínez, S. y Del Blanco, D. *PoliformaT: una estrategia para la formación on-line en la Educación Superior*. Virtual Educa, 2006. <http://ihm.ccadet.unam.mx/virtualeduca2006/pdf/177-DRM.pdf>
- [4] Gómez, J. A. *Portal Web de Autocorrección de Programas para Programación de Primer Curso*. En II Jornadas Nacionales de Metodologías ECTS, 2007.
- [5] Pérez, V. y Merín, R. *El Puzzle de Aronson: Una Experiencia de Aprendizaje Cooperativo en la Formación del Profesorado*. En II Congreso Internacional Federación Española de Asociaciones de Docentes de Educación Física, 1998.
- [6] Sanabria, E., Conejero, J.A. y Camp, S. *Organización del trabajo en equipo mediante la técnica del puzzle de Aronson*. En 3er Congreso Internacional de Docencia Universitaria e Innovación (CIDUI), 2004.
- [7] Valero, M. y Navarro, J.J. *Niveles de competencia de los objetivos formativos en las ingenierías*. En Jornadas de Enseñanza Universitaria de la Informática, 2001.
- [8] Valero, M. y Espona, M. *Material del taller "Adaptación de asignaturas al EEES"*. Universidad Politécnica de Valencia, 2010.
- [9] *Guía Docente de Introducción a la Informática y la Programación*. ETSINF, Universidad Politécnica de Valencia, 2010. http://www.inf.upv.es/gradoII/grado_plan_estudios.php.
- [10] *Normativa de Régimen Académico y de Evaluación del Alumnado*. Universidad Politécnica de Valencia, 2010. <http://www.upv.es/up1/U0490393.pdf>.
- [11] *Propuesta para la renovación de las metodologías educativas en la universidad*. Ministerio de Educación y Ciencia, 2006.
- [12] RD 1393/2007 de 29 de octubre de 2007. Boletín Oficial del Estado, núm. 260. pp. 44037-44048. 2007.

Corrección automática de ejercicios de estructuras de datos a través de una plataforma de e-learning

Joan Surrell, Imma Boada, Josep Soler, Ferran Prados, Jordi Poch

Dpto. de Informática y Matemática Aplicada. Universidad de Girona
Campus de Montilivi, Edificio P4, 17071 Girona

joan.surrell,imma.boada,josep.soler,ferran.prados,jordi.poch@udg.edu

Resumen

Las estructuras de datos son una de las materias más importantes en las carreras de Informática. En este artículo presentamos el uso y las ventajas que ofrece la utilización de una plataforma de e-learning en el desarrollo de esta asignatura. Entre las ventajas que nos proporciona esta herramienta destacamos la corrección automática y on-line de los distintos tipos de ejercicios relacionados con la materia: estructuras dinámicas, lineales, árboles, etc. Además, permite al profesor conocer en todo momento el nivel de aprendizaje del alumno y detectar rápidamente posibles deficiencias. Con el uso de esta plataforma hemos logrado llevar un control exhaustivo del trabajo del alumno y motivar al alumnado con el objetivo de mejorar los resultados académicos.

Summary

Data structures are one of the most important subjects in Computer Science. In this paper we present an e-learning tool designed to give support at teaching and learning of this topic. We describe the tool, how it is used and the main benefits that provide us. Amongst them, there are the on-line and automatic correction of different types of exercises related to the subject, such as, dynamic structures, sequential or trees; the capability to track the student progress which allows the teacher to know about their deficiencies.

Palabras clave

E-learning, corrección automática, estructuras de datos, programación.

1. Introducción

El proyecto de construcción del Espacio Europeo de Educación Superior (EEES) y su estructuración

a través del sistema europeo de créditos (ECTS), propone un nuevo modelo educativo en el cual los planes docentes y las metodologías a usar han de estar basados en el aprendizaje de los estudiantes, en contraposición con la tradicional programación centrada en la enseñanza del profesor. Se pretende un cambio en el modelo educativo universitario donde el estudiante sea el protagonista y donde lo más importante sea lo que debe aprender y como podemos garantizar que lo aprenda.

Conscientes de estos cambios, un grupo de profesores del Departamento de Informática y Matemática Aplicada de la Universidad de Girona nos propusimos desarrollar un sistema que facilitase y a su vez motivase al alumno en el aprendizaje de asignaturas con un gran componente práctico. De todos es conocido el fracaso del sistema tradicional donde un profesor, después de explicar una materia, facilita un listado de problemas a los alumnos, desprecupándose tanto de si los saben hacer, como de si realmente los hacen. Ante este problema decidimos desarrollar una plataforma para asignar problemas a cada alumno. Esta plataforma guarda las soluciones enviadas en la base de datos del sistema y las corrige automáticamente. En caso de ser erróneas se facilitan las indicaciones oportunas para su corrección. De esta forma podemos seguir el trabajo del alumno, detectamos sus deficiencias y además podemos realizar una evaluación continuada. En este artículo se presenta el uso que hacemos de esta plataforma y las ventajas que ofrece la corrección automática en la docencia de la asignatura de "Introducción a las estructuras de datos". Esta es una asignatura de segundo curso de las Ingenierías Técnicas en Informática (Gestión y Sistemas) y suele ser una asignatura con un alto número de abandonos y de suspensos.

El artículo se ha estructurado en siete secciones. En la sección 2 se presenta un breve análisis de otras experiencias relacionadas. En la

sección 3 se describen los contenidos que generalmente se imparten en una asignatura de estructuras de datos. En la sección 4 se describe la plataforma ACME con la que venimos trabajando y los tipos de problemas que soporta. En la sección 5 se presenta como se ha usado la plataforma en la asignatura de Estructuras de Datos de nuestra universidad. En la sección 6 se comenta la metodología usada y como utilizamos la plataforma para el seguimiento y evaluación de los alumnos. Finalmente, en la sección 7 se presentan las conclusiones.

2. Experiencias relacionadas

Se han llevado a cabo muchas experiencias consistentes en utilizar entornos virtuales para complementar la docencia presencial de distintas materias. Ya en el ámbito de las estructuras de datos y como experiencias presentadas en estas jornadas podemos citar [1-5]. Aunque en distintos contextos, el denominador común de todos estos trabajos es facilitar el aprendizaje de la materia. En [1] se centran en la generación de recursos educativos a ser usados, en [2] se presenta un entorno colaborativo para la docencia online de programación que permite la implementación del Aprendizaje Basado en Proyectos (ABP). En la misma línea de ABP en [3] se presenta una herramienta para la monitorización de una práctica para esta asignatura. En [4] se presenta una herramienta pensada ya para la evaluación continua y que permite la corrección automática de los ejercicios. En [5] se presenta una plataforma abierta que proporciona mecanismos para realizar una evaluación completa de un ejercicio de programación realizado en C o en Java para una asignatura de programación básica.

La corrección automática de programas informáticos es un tema sobre el que se han realizado numerosos estudios y proyectos. La finalidad de estos es reducir el tiempo de corrección de las prácticas de estas materias y por otro lado facilitar al alumno la validación de sus ejercicios. La estrategia de corrección automática más usada consiste en ejecutar el programa desarrollado por el alumno sobre un conjunto de test de pruebas. Cada uno de estos test está formado por unas entradas y la salida esperada. En caso que el programa del alumno pase todos los

test de pruebas el programa se considera correcto. En esta línea destacamos los trabajos [6-8] que permiten la corrección y evaluación de forma automática de programas escritos en un determinado lenguaje de programación. Nuestro sistema sigue la metodología de estos trabajos permitiendo tanto la corrección como la evaluación continua de programas escritos en diferentes lenguajes. La descripción de la primera versión de nuestro sistema de corrección de ejercicios de programación se detalla en [9].

3. Estructuras de datos

Las asignaturas de Estructuras de Datos suelen impartirse en un segundo nivel de la carrera de Informática después de haber cursado una materia introductoria de programación básica. Los alumnos que acceden a este segundo nivel conocen las bases de la programación (recorridos, búsquedas, ordenación, acciones/funciones, paso de parámetros, etc.) y aspectos elementales de estructuras de datos (tablas y tuplas). En la asignatura de Estructuras de Datos se presentan las estructuras dinámicas encadenadas, las estructuras lineales (pilas, colas y listas) y los árboles binarios. A partir de estos conceptos pueden desarrollarse estructuras más complejas (diccionarios, árboles de búsqueda, grafos, etc.) y los algoritmos asociados a su tratamiento.

La forma tradicional de presentación de la materia dispone de sesiones teóricas en las que se exponen los conceptos del tema ilustrados con ejemplos clarificadores y se complementan con sesiones prácticas de problemas de clase para asentar los conceptos presentados, así como el desarrollo de uno o más programas de tamaño mediano en los que el alumno puede ver la aplicación práctica de los conocimientos adquiridos. El desarrollo de la práctica suele hacerse en lenguajes imperativos o con una componente de orientación a objetos (C, C++, Ada, Java) y en la teoría/problemas se acostumbra a usar un lenguaje algorítmico o una versión simplificada del mismo lenguaje de programación.

Esta materia es acumulativa, pues cada nuevo tema se basa en lo desarrollado en temas anteriores. Difícilmente un alumno comprenderá los temas finales de la materia (algoritmos sobre grafos o estructuras de datos compuestas) si no ha

asimilado los conceptos básicos (estructuras dinámicas y árboles binarios). Al tratarse de conceptos eminentemente prácticos, la resolución de problemas y la implementación práctica de programas que aplican los aspectos desarrollados en las sesiones teóricas, son elementos esenciales para asimilar la materia. Por esta razón la asimilación de los primeros temas condiciona mucho la superación de la asignatura. Sin embargo la organización clásica de la materia no permite al profesor conocer la evolución del grupo debido a la masificación y al desconocimiento del grado de éxito que tienen en la resolución de problemas (si llegan a resolverlos).

Queremos resaltar que en este tipo de asignaturas el trabajo personal del alumno y la resolución de problemas es fundamental. Por ello, el uso de plataformas de soporte a la docencia nos permite mejorar la docencia y el seguimiento del trabajo del alumno. Dadas las características de la plataforma ACME y las necesidades de las asignaturas de Estructuras de Datos podemos obtener muchos beneficios con su aplicación. A continuación, presentamos las principales características de la plataforma ACME.

4. La plataforma ACME

Para presentar la plataforma ACME en primer lugar describiremos las características principales de la misma. A continuación, presentaremos los diferentes tipos de problemas que soporta centrándonos en los que se usan en la asignatura de Estructuras de Datos.

4.1. Características de la plataforma

El desarrollo de la plataforma ACME se realizó con el fin de satisfacer los siguientes objetivos:

- Disponer de un repositorio único de problemas base de distintos tipos, donde los profesores puedan introducir y compartir sus problemas. Los problemas se catalogan según las materias, asignaturas, temáticas y nivel de dificultad de cada uno de ellos.
- A ser posible cada problema base dispone de varios enunciados y de parámetros variables, de forma que combinándolos se puedan hacer múltiples variaciones de un mismo problema. Cada problema base lleva asociado las pautas para su corrección automática. Por ejemplo, un problema de programación lleva asociado

varios test de pruebas con unas entradas y las salidas esperadas para cada una de ellas.

- En cada asignatura en que se utiliza la plataforma, el sistema ofrece a cada alumno un cuaderno de problemas distinto. Este cuaderno está organizado por temas en los que se añaden los distintos problemas a medida que avanza el curso. El alumno envía soluciones a los problemas asignados y ACME los corrige automáticamente.
- El sistema guarda todas las soluciones enviadas por el alumno pudiendo ser consultadas por el profesor.
- En función del número de soluciones enviadas hasta obtener la solución correcta, el profesor puede hacer una primera valoración de las habilidades del alumno.
- El sistema debe soportar la corrección automática y on-line de los distintos tipos de problemas. Cada tipo de problemas tiene su módulo corrector específico.
- Todas las funcionalidades de la plataforma se deben realizar desde cualquier navegador.

A modo de ejemplo en las figuras siguientes se muestran dos interfaces habituales de trabajo de la plataforma ACME. En la Figura 1 puede verse, para un grupo de alumnos, el estado de resolución de los problemas de un determinado tema. También podemos ver el número de soluciones enviadas hasta llegar a la solución correcta. La Figura 2, visualiza el historial de un alumno, mostrando el número de tema / problema y los errores cometidos. Seleccionando un problema pueden verse las distintas soluciones enviadas.

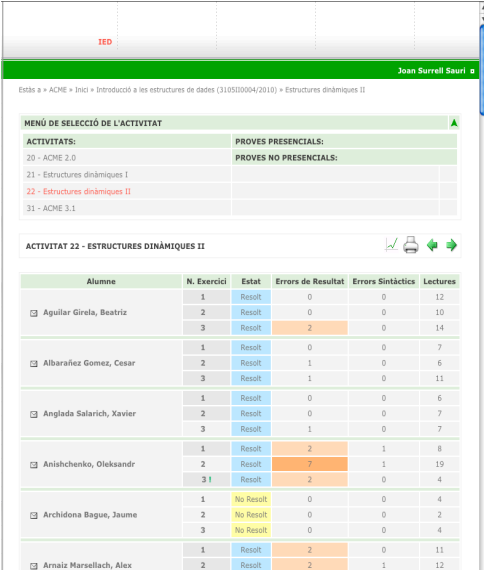
Comentar que en su última versión ACME ya está totalmente integrado en la plataforma estándar Moodle, de forma que fácilmente se puede implantar en los centros que dispongan de ella. Desde esta plataforma los profesores pueden escoger en el apartado de actividades cualquier problema ACME.

4.2. Tipos de problemas

La estructura modular de la plataforma ACME facilita la integración de nuevos correctores de problemas con lo cual es muy fácil ampliar la tipología de problemas soportados.

Para la incorporación de un nuevo tipo de problema al sistema, sólo hay que definir la interfaz adecuada, la estructura y pautas de

corrección de los problemas y evidentemente desarrollar el corrector pertinente.



Estás a » ACME » Inici » Introducció a les estructures de dades (310510004/2010) » Estructures dinàmiques II

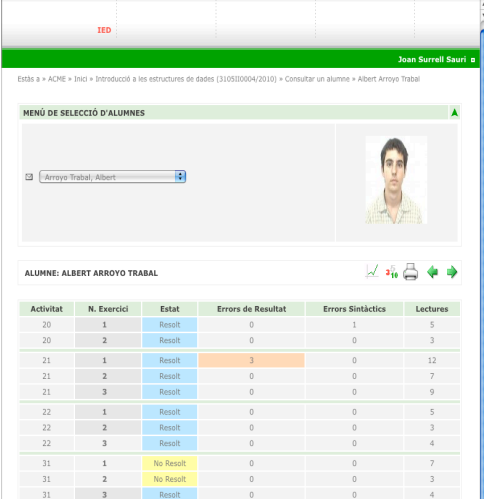
MENÚ DE SELECCIÓ DE L'ACTIVITAT

ACTIVITATS:	PROVES PRESENCIALS:
20 - ACME 2.0	PROVES NO PRESENCIALS:
21 - Estructures dinàmiques I	
22 - Estructures dinàmiques II	
31 - ACME 3.1	

ACTIVITAT 22 - ESTRUCTURES DINÀMIQUES II

Alumne	N. Exercici	Estat	Errors de Resultat	Errors Sintàctics	Lectures
☑ Agullar Girela, Beatriz	1	Result	0	0	12
	2	Result	0	0	10
	3	Result	2	0	14
☑ Albarañez Gomez, Cesar	1	Result	0	0	7
	2	Result	1	0	6
	3	Result	1	0	11
☑ Anglada Salarich, Xavier	1	Result	0	0	6
	2	Result	0	0	7
	3	Result	1	0	7
☑ Anisshchenko, Oleksandr	1	Result	2	1	8
	2	Result	7	1	19
	3	Result	2	0	4
☑ Archidona Bague, Jaume	1	No Result	0	0	4
	2	No Result	0	0	2
	3	No Result	0	0	4
☑ Arnau Marsellach, Alex	1	Result	2	0	11
	2	Result	2	1	12

Figura 1. Estado de resolución de los problemas



Estás a » ACME » Inici » Introducció a les estructures de dades (310510004/2010) » Consultar un alumne » Albert Arroyo Trabal

MENÚ DE SELECCIÓ D'ALUMNES

☑ [Arroyo Trabal, Albert]

ALUMNE: ALBERT ARROYO TRABAL

Activitat	N. Exercici	Estat	Errors de Resultat	Errors Sintàctics	Lectures
20	1	Result	0	1	5
20	2	Result	0	0	3
21	1	Result	3	0	12
21	2	Result	0	0	7
21	3	Result	0	0	9
22	1	Result	0	0	5
22	2	Result	0	0	3
22	3	Result	0	0	4
31	1	No Result	0	0	7
31	2	No Result	0	0	3
31	3	Result	0	0	4

Figura 2. Historial de un alumno

En la actualidad la plataforma corrige problemas estándar, es decir los que se encuentran

en cualquier plataforma como son los tipos test, elección múltiple o rellenar blancos. Además integra correctores automáticos para materias científicas/técnicas específicas de carreras universitarias. De esta manera ACME permite en estos momentos la corrección de:

- Cualquier problema que requiera un planteamiento matemático, desde derivadas, integrales, ecuaciones diferenciales, hasta problemas de física, economía, química, etc. [10].
- Corrección de ejercicios de programación escritos en distintos lenguajes informáticos y también en pseudo-código [9].
- Ejercicios de bases de datos, desde la corrección de diagramas entidad-relación, de esquemas de BD relacionales, sentencias SQL, álgebra relacional, etc. [11].

5. Aplicación de ACME en la asignatura de Estructuras de Datos

En una asignatura de estructura de datos es esencial que los alumnos aprendan a gestionar la información en memoria interna. Deben saber crear estructuras de tamaño variable, insertar datos, borrar, buscar y modificar. Además deben conocer las ventajas de las distintas maneras de estructurar los datos, así como los algoritmos particulares de cada representación.

Podemos considerar, por tanto, que para un alumno es fundamental ejercitarse en estructuras dinámicas y sus algoritmos característicos. Deben conocer las posibilidades de las estructuras lineales y el estudio de los árboles binarios con toda la problemática que conlleva la recursividad. La consolidación de estos aspectos básicos es fundamental para poder afrontar con éxito el estudio de técnicas más complejas (diccionarios de datos, técnicas de dispersión, árboles balanceados, grafos). La corrección de los problemas relacionados con esta materia requiere mucha dedicación por parte del profesor por lo cual no se pueden plantear tantos como sería deseable. Es en este punto donde la plataforma ACME nos ayuda.

A continuación se describen los tipos de problemas que permite la plataforma y que son usados en la asignatura de Estructuras de Datos.

5.1. Problemas de programación básica

Los alumnos que acceden a la asignatura de Estructura de datos ya han cursado una asignatura de programación en el nivel anterior. Con objeto de consolidar los conocimientos adquiridos anteriormente, se proponen dos o tres ejercicios de programación básica en las sesiones iniciales de la asignatura. De esta forma los alumnos pueden repasar conceptos adquiridos como son recorridos, búsquedas, codificación de clases y a la vez familiarizarse con el entorno ACME, en el caso de que no lo hayan usado anteriormente. El profesor aprovecha estas sesiones iniciales para avanzar suficiente materia en las sesiones teóricas para poder realizar los programas que realmente pertenecen a la materia. Podemos considerar que en esta fase ACME se usa como herramienta para la consolidación de conceptos básicos que se suponen ya adquiridos.

5.2. Problemas de estructuras dinámicas

Cuando ya se ha introducido suficiente materia de la asignatura, el estudiante debe enfrentarse a los primeros problemas básicos de estructuras de datos. Para ello se le plantean dos conjuntos de ejercicios con dificultad creciente en los que debe demostrar su capacidad para trabajar con estructuras de datos dinámicas. En estos ejercicios iniciales el programa principal lo proporciona el propio enunciado, así como la representación de la estructura de datos con la que se trabajará. El alumno debe codificar sólo los métodos indicados de la clase que encapsula la estructura dinámica. Una vez codificados los métodos por parte del alumno, y verificados en su ordenador, debe enviarlos mediante la página web del problema para que el sistema ACME pueda determinar si son correctos o no.

En la Figura 3 se muestra un ejemplo de problema. Como puede verse el enunciado del problema no es simplemente un texto sino que también se adjuntan detalles del código, como debe hacerse la entrada y salida de datos, la declaración de la estructura a representar así como representaciones gráficas que ayuden a la interpretación del problema. El alumno puede descargarse el código del problema desde la propia web, debe completarlo, verificarlo y enviarlo para su corrección por parte del sistema.

Para determinar el buen funcionamiento del problema, el profesor adjunta diversas secuencias de entrada de datos y sus correspondientes salidas que permiten verificar el funcionamiento del problema. Estas secuencias no son visibles por parte de los alumnos. Cuando un alumno envía un problema para su corrección el sistema compila el código, lo ejecuta y aplica las secuencias de verificación. Como resultado de este proceso, el sistema proporciona al alumno de este tipo de problemas una respuesta que puede ser de tres niveles:

- Errores de compilación: no suelen ser habituales si el estudiante ha verificado inicialmente el problema en su ordenador, aunque pueden darse por confusión en el envío de fichero u otras circunstancias.
- Errores de ejecución: son los más habituales y suelen darse en casos no previstos por los alumnos (estructuras vacías, borrado del primer o último elemento,...). El sistema visualiza al alumno un ejemplo de entrada, la salida esperada y la obtenida, marcando con colores rojo y verde las diferencias detectadas. En la Figura 4 se muestra un ejemplo del mensaje de respuesta que recibe el alumno ante una solución incorrecta. Como puede verse se muestra en verde la solución correcta y en rojo el resultado obtenido por el alumno.
- Correcto: todos los test proporcionados con el enunciado son superados, el sistema acepta como correcto el problema.

EXERCICI: Omplir estructura de llistat Estructures dinàmiques II

Es disposa d'una estructura dinàmica circular implementada amb sentinella i un punter a l'inici tal i com es representa a la figura que hi ha al final del text. Implementar en C++ els mètodes `addElement` que afegeix un element a l'estructura i `OmplirFinal` que posa un conjunt de valors en ordre creixent al final de l'estructura. Els valors a posar són els enters consecutius des de 1 fins al valor indicat pel paràmetre. Utilitzar les operacions anteriors en el següent programa (`main.cpp`):

```
#include <iostream>
#include "estructuraDinamica.h"
using namespace std;

void main() {
    estructuraDinamica a;
    int primer, anterior, n;
    cin >> n;
    while (n != 0) {
        a.OmplirFinal(n);
        cin >> n;
    }
    a.Llistar();
    cout << "\nELEMENTS: " << a.NELEMENTS() << endl;
}

Cal tenir en compte que OmplirFinal no verifica si ja existeixen els elements afegits. De fet, en casos successius, es registren els enters afegits. El límit de capacitat de l'estructura (estructuraDinamica.h):
```

```
#define TD_estructuraCircularSentinella_h
#define TD_estructuraCircularSentinella_h

struct node {
    int dada;
    node *seguint;
};

class estructuraDinamica { // Circular Sentinella
    node * inici;
public:
    estructuraDinamica();
    void OmplirFinal(int n);
    int NELEMENTS() const;
    void Llistar() const;
};

#definif

El constructor per defecte i el mètode Llistar són els següents:
```

```
estructuraDinamica::estructuraDinamica() {
    inici = new node; inici->seguint = inici; inici->dada = 0;
}

void estructuraDinamica::Llistar() const {
    if (inici != inici->seguint) {
        node * p = inici->seguint;
        while (p != inici) {
            cout << p->dada << " "; p = p->seguint;
        }
        cout << "\n" << inici->dada << endl; // llista el sentinella
    }
}
```

Cal descarregar el codi adjunt al problema.

Figura 3. Problema de estructures dinàmiques

Figura 4. Missatge de resposta.

En este tema se plantean al alumno entre 5 y 7 problemas repartidos en dos módulos (2 o 3 de algoritmos básicos de estructuras dinámicas y 3 o 4 de algoritmos más complejos). Se realizan este tipo de problemas durante 3 semanas del curso.

5.3. Problemas de árboles binarios

Las estructuras de datos basadas en árboles merecen un capítulo aparte pues en ellas confluyen dos aspectos de gran dificultad para los alumnos: estructuras dinámicas y recursividad. Por esta razón el enfoque es parecido al de las estructuras lineales, pero con una dificultad progresiva de tratamiento. Cada problema dispone de 3 enunciados iguales, pero con datos de

naturaleza cada vez más compleja: árboles con sólo dos niveles, árboles con diversos niveles pero con dos hijos en cada nivel y árboles generales que pueden tener diversos niveles con 1 o 2 hijos en cada nivel. Esta presentación de forma progresiva permite al estudiante refinar un algoritmo recursivo en caso que tenga dificultad en diseñarlo de forma completa.

Por lo que respecta al resto de los aspectos, estos problemas no se diferencian de los de estructuras dinámicas. El estudiante dispone de parte de código en el enunciado, debe completar uno o varios métodos de la clase, puede probar el código en su ordenador antes de enviarlo y, cuando lo envía, recibe una respuesta del sistema en los mismos 3 niveles descritos anteriormente (error de compilación, error de ejecución y correcto). En la Figura 5 se muestra un ejemplo de enunciado de árboles binarios. En él se puede observar el programa principal (incompleto), indicaciones de cómo debe proporcionarse la salida, la declaración de la clase a desarrollar y los métodos a implementar.

EXERCICI: Constructor arbre complet de l'activitat ACME 3.1

Es disposa d'un arbre binari de caràcters amb un punter a l'arrel tal i com es representa a la figura que hi ha al final del text. Implementar en C++ el constructor a partir d'una cadena amb el recorregut en preordre i tots els algorismes necessaris per poder executar el següent programa (`main.cpp`):

```
#include <iostream>
#include "ArbreBinari.h"
using namespace std;

void postordre_binari(const ArbreBinari &a);

void main() {
    ArbreBinari a;
    char entrada[100];
    cin.getline(entrada, 100);
    a = new ArbreBinari(entrada);
    cout << "Postordre: "; postordre_binari(a); cout << endl;
}

Cal tenir en compte que postordre_binari mostra per pantalla el recorregut en postordre de l'arbre que rep com a punter, separant els elements amb espais en blanc. Cal implementar-lo juntament amb el programa principal, com a soci extern a la classe.

El fitxer de capçalera de la classe és el següent (ArbreBinari.h):
```

```
struct node {
    char dada;
    node *fo, *fd;
};

class ArbreBinari { // Arbre Binari
    node * arrel;
public:
    ArbreBinari(char *, int i); // metode auxiliar del constructor
    ArbreBinari(char *);
    bool arbre_buit() const;
    ArbreBinari * fill_dret() const;
    ArbreBinari * fill_esquerra() const;
    char contingut() const;
    // no s'ha de posar destructor.
};

Excepcionalment i sense que serveti de precedent, en aquesta classe cal no posar destructor. La resta de mètodes s'han d'implementar en el fitxer ArbreBinari.cpp.

Exemple d'execució (cal tenir en compte que en aquest exercici pot haver-hi fills esquerres sense dret o fills dretes sense esquerres):
```

```
A(B(C) F) D(G H)
Postordre: E C F B G H D A
```

Cal descarregar el codi adjunt al problema.

Figura 5. Problema de arbres binaris

Debido a la mayor dificultad que presentan los problemas de árboles se realizan menos ejercicios (sólo 3 o 4) y se desarrollan durante un período de 3 semanas.

5.4. Aplicación a otros problemas

La plataforma permite abordar la resolución de problemas simples como los usados en estructuras de datos básicas y otros más complejos que puedan evaluarse comparando sus entradas y salidas. Para problemas que requieren un conjunto de clases para su resolución, deben testearse individualmente las clases y, cuando ya funcionan todas ellas, realizar un test final del conjunto. Esta posibilidad puede aplicarse en el caso en que las clases son conocidas de antemano, pero no si la resolución del problema implica la determinación de las clases. Este tipo de problemas deben considerarse como problemas de diseño, no de programación, pues el diseño es el aspecto más importante a evaluar. El módulo ACME de programación no es aplicable a problemas de diseño.

6. Metodología usada

El uso de la plataforma ACME en esta asignatura ha pasado por diferentes etapas.

Las primeras experiencias son del curso 2004/05. En esta etapa inicial el uso de la plataforma ACME en la asignatura de Estructura de Datos estaba centrado en la parte de estructuras dinámicas y punteros. Se asignaban 3 problemas a los alumnos y estos debían resolverlos en unos límites de tiempo preestablecidos. Los primeros resultados fueron satisfactorios por lo cual se amplió el uso de la plataforma progresivamente a árboles binarios y a estructuras lineales. La idea de partida era usar la plataforma para resolver problemas sobre estos aspectos de forma exclusiva.

La introducción de la plataforma no implicó grandes cambios en la metodología docente. El proceso era simple: el profesor proponía un conjunto de problemas del tema. Los alumnos disponían de un período limitado de tiempo para resolverlos, pasado el cual eran evaluados por el sistema según el número de errores que la resolución del problema hubiera comportado.

Inicialmente sólo se disponía de unos pocos ejercicios de estructuras dinámicas, que fueron

ampliándose con el tiempo. A continuación se desarrollaron ejercicios introductorios de árboles binarios, para completarlo finalmente con ejercicios de estructuras lineales. Aunque la experiencia parecía positiva por lo que respecta a la participación de los alumnos, no así la adquisición de conocimientos, pues algunos aspectos de la programación no pueden aprenderse con la sola verificación de las entradas y salidas de los programas. Estas limitaciones nos llevaron a una segunda fase que implicó un cambio en la metodología.

La experiencia adquirida en la primera fase de aplicación de ACME en la asignatura nos llevó a agrupar las estructuras dinámicas con las lineales. Hasta el momento estas estructuras se trataban de forma independiente. La forma de evaluación también se modificó. Se definió un sistema de evaluación híbrido. Por una parte había un conjunto de problemas que eran evaluados por el entorno ACME. Por otra parte, los profesores evaluaban unos pocos ejercicios presentados en papel (2 o 3 ejercicios). Las dos evaluaciones se complementaban.

El mismo sistema se adoptó para el caso de los árboles. Así pues, los ejercicios iniciales se desarrollaban en el entorno ACME y se complementaban con los ejercicios que debían presentarse en papel y que eran corregidos por el profesor.

Este nuevo sistema de evaluación nos presenta diferentes ventajas, entre ellas:

- El alumno resuelve de forma autónoma un elevado número de problemas.
- El profesor sólo debe evaluar un reducido número de los mismos.
- Los problemas que recibe el profesor han pasado varios test de prueba que permiten asegurar un funcionamiento mínimo y, por tanto, cierto tipo de errores ya se han eliminado.
- A pesar de que el entorno ACME funciona de forma autónoma, el alumno puede consultar al profesor la causa de los errores que le detecta el sistema y éste puede interactuar con el mismo código con el que trabaja el alumno.
- La presentación de algunos ejercicios en papel ayuda a disminuir la inevitable copia de ejercicios. Aunque ACME dispone de un sistema de comparación de ficheros que permite detectar ejercicios iguales o bastante

parecidos, no se ha usado. Debe ser el propio alumno quien se responsabilice de su aprendizaje. Presentar ejercicios originales en la corrección automática permite al alumno afrontar con más garantías los ejercicios de evaluación manual y los exámenes. ACME es una herramienta de aprendizaje más que de evaluación. La evaluación sólo se proporciona para motivar al alumno.

Desde un punto de vista metodológico la integración de ACME en la asignatura no supuso grandes cambios. Fue necesario preparar el material y los problemas para poder trabajar a través de la plataforma. Desde el punto de vista académico los resultados que se obtuvieron también son satisfactorios. Como hitos destacables remarcamos el mejor aprovechamiento de la asignatura por parte de los alumnos y la mayor asistencia a las sesiones teóricas y de laboratorio. Además, conseguimos un mejor rendimiento académico, tal como muestran los datos de la Tabla 1 en la que se comparan los dos últimos cursos respecto al último curso (2003/04) en el que todavía no se usaba la plataforma ACME. En esta Tabla para cada curso diferenciamos entre los alumnos que han seguido la evaluación continuada de los que han realizado un único examen final. En ambos cursos puede observarse que los alumnos que optaron por la evaluación continua en la que se usaba ACME como parte de la evaluación obtuvieron mucho mejor resultado que los que optaron por el examen final. A remarcar que los alumnos que seguían esta última opción eran los que no podían asistir regularmente a clase.

Curso	Evaluación	Aptos	No aptos
2003/04		75	104
2008/09	Continua	58	15
	Ex. Final	3	39
2009/10	Continua	63	33
	Ex. Final	3	34

Tabla 1. Relación de alumnos que superan la asignatura en función de seguir evaluación continua o examen final

7. Conclusiones

En este documento se ha descrito el uso de la plataforma ACME como herramienta de corrección automática de problemas en una asignatura de estructura de datos. Se ha descrito la experiencia docente realizada desde el curso 2004/05 en la Universidad de Girona con objeto de potenciar las bases de la asignatura, especialmente los aspectos relacionados con las estructuras dinámicas y los árboles binarios. Como valoración final de la experiencia descrita se resalta la mejor consolidación de los conocimientos por parte de los alumnos, una mayor asistencia a las sesiones de teoría y laboratorio así como una mejora significativa del porcentaje de alumnos que supera la materia.

Referencias

- [1] Sarasa A. *Generación de recursos educativos para la enseñanza de la asignatura de Estructuras de datos y de la información en un campus virtual*. JENUI 2004.
- [2] Gallego M., Gortázar F. *EclipseGavab, un entorno de desarrollo para la docencia online de la programación*. JENUI 2009.
- [3] Fiol G., Fiol C., Miró M. *La práctica monitorizada: una herramienta válida en el aprendizaje activo de la asignatura Estructuras de la Información*. JENUI 2007
- [4] Moltó G., Galiano M., Herrero C., Prieto N., Sapena O. *Uso de herramientas TIC para la mejora de la interacción profesor-alumno, la evaluación continua y el aprendizaje autónomo*. Jornadas de Innovación UPV 2009: Metodologías Activas para la Formación en Competencias, 2009.
- [5] Romero F.P., Serrano J., Pérez de Inestrosa H. *CUESTOR: Una nueva aproximación integral a la evaluación automática de prácticas de programación*. JENUI 2010
- [6] Higgins C., Gray G., Symeonidis P. and Tsintsifas A. *Automated assessment and experiences of teaching programming*. ACM Journal of Educational Resources in Computing. Vol 5(3), pp 1-21, 2005.
- [7] Joy M., Griffiths N. and Boyatt R. *The BOSS on-line submission and assessment system*.

- ACM Journal on Educational Resources in Computing. Vol 5(3). 2005.
- [8] Saikkonen R., Malmi L. and Korhonen A. *Fully Automatic Assessment of Programming Exercises*. Proc. of Conference on Innovation and Technology in Computer Science Education (ITiCSE). pp 133-136, 2001.
- [9] Boada I., Soler J., Prados F., Poch J. *A teaching/learning support tool for introductory programming course*. IEEE Proc. 5th International Conference on Information Technology based Higher Education and Training ITHET 2004.
- [10] Prados F., Boada I., Soler J., Poch J. *Automatic Generation and Correction of Technical Exercises*. Proc. International Conference on Engineering and Computer Education ICECE 2005.
- [11] Soler J. *Entorno virtual para la docencia y la evaluación automática en Bases de Datos*. Tesis doctoral. 2010.

Sesión 2A: Evaluación del alumnado II

Puntuación entre iguales para la evaluación del trabajo en equipo

Alberto Abelló Xavier Burgués
Dept. d'Enginyeria de Serveis i Sistemes d'Informació
Universitat Politècnica de Catalunya
C/ Jordi Girona 1-3, 08034 Barcelona
{aabello,diafebus}@essi.upc.edu

Resumen

La entrada en el EEES y la adopción de un sistema de evaluación basado en competencias, algunas de ellas no técnicas, hace que nos tengamos que plantear algún tipo de cambio, no solo en la forma de enseñar, sino también en la forma de evaluación. Evaluar, por ejemplo, la actitud ante el trabajo, el trabajo en equipo o la capacidad de innovación mediante un examen resulta a todas luces poco apropiado, si no imposible.

Es en este sentido que hemos experimentado durante dos semestres la posibilidad de evaluación entre iguales para la competencia genérica “trabajo en equipo”. En este trabajo, presentamos la experiencia y conclusiones extraídas.

1. Introducción

La implantación del grado en informática y la consecuente adaptación de los estudios al EEES ha resultado, no solo en una oportunidad de innovación docente, sino también en una necesidad de adaptarnos a las nuevas exigencias. Hemos pasado de asignaturas (y estudios) basadas en objetivos a unas basadas en competencias. La mayoría de estas competencias son de índole técnica o tecnológica. No obstante, algunas de ellas son de tipo genérico y transversales al contenido de las asignaturas.

Así, en la Facultad de Informática de Barcelona de la UPC se han incluido en el grado las siguientes competencias genéricas [7] (algunas dictadas por el ministerio, otras por la universidad y el resto introducidas por la propia facultad):

- Iniciativa e innovación
- Sostenibilidad y compromiso social
- Tercera lengua (inglés)
- Comunicación eficaz oral y escrita
- Trabajo en equipo

- Uso solvente de recursos de información
- Aprendizaje autónomo
- Actitud apropiada ante el trabajo
- Hábitos de pensamiento apropiados (capacidad de razonamiento crítico, lógico y matemático)

Está claro que desarrollar y evaluar estas competencias con los métodos tradicionales resulta totalmente inapropiado. En este trabajo proponemos el uso de la evaluación entre iguales para promover la reflexión y evaluar una de estas competencias: “Trabajo en equipo”, claramente destacada en [9] y [2]. Esta competencia se ha definido en nuestra facultad como “ser capaz de trabajar como miembro de un equipo, ya sea como un miembro más o realizando tareas de dirección, con la finalidad de contribuir a desarrollar proyectos de una forma pragmática y con sentido de la responsabilidad asumiendo compromisos teniendo en cuenta los recursos disponibles”.

El artículo está organizado en 6 secciones. La sección 2 introduce el marco concreto en el cual se ha desarrollado la experiencia. La sección 3 presenta la motivación y el ánimo que nos ha movido a llevar a cabo la experiencia, así como las dificultades que presentan este tipo de actividades. La sección 4 describe lo que propusimos concretamente a los estudiantes. La sección 5 muestra un análisis estadístico de los resultados obtenidos desde diferentes puntos de vista. Finalmente, la sección 6 resume las conclusiones que hemos extraído y como proponemos mejorar la puesta en práctica el próximo semestre.

2. Contexto de la experiencia

En esta sección explicamos el marco en el cual se ha desarrollado la experiencia, en primer lugar en lo referente a las herramientas utilizadas y posteriormente en cuanto a la organización y evaluación de la asignatura sobre la que se ha llevado a cabo.

2.1. Moodle y LEARN-SQL

Desde el curso 07/08, las asignaturas del área de bases de datos se vienen apoyando en Moodle¹ [1], que es el *Learning Management System* escogido por la UPC para dar soporte a su Campus Virtual. Se ha ido personalizando y complementando Moodle para implementar y facilitar los cambios que se han ido aplicando a las diferentes asignaturas. Hay que destacar el desarrollo de una herramienta que tiene como interfaz de usuario al propio Moodle, llamada LEARN-SQL [5], cuyo objetivo es corregir automáticamente cualquier tipo de sentencia SQL (consultas, actualizaciones, procedimientos almacenados, disparadores, etc.) y discernir si la respuesta aportada por el estudiante es o no correcta con independencia de la solución concreta que éste proponga. Además, esta herramienta ayuda a los profesores a diseñar las pruebas de evaluación, permitiendo también la opción de revisar cualitativamente las soluciones aportadas por los estudiantes. Por último, el sistema proporciona ayuda a los estudiantes para que aprendan de sus propios errores, dando realimentación de calidad. Los profesores generan un archivo de preguntas y posteriormente se generarán cuestionarios a partir de las preguntas almacenadas. Los estudiantes pueden acceder al cuestionario a través de un navegador y proponer soluciones a las preguntas, soluciones que serán evaluadas por el sistema. Esta evaluación es automática, interactiva (el estudiante obtiene realimentación en cuestión de segundos) e informativa (el estudiante obtiene información sobre los errores que ha cometido en su solución y una posible orientación para solucionarlos). Otros aspectos que podemos citar son el carácter multilingüe (tanto de la interfaz como de los enunciados de las preguntas; téngase en cuenta que algunos semestres la asignatura se imparte en inglés) y un módulo que permite a los estudiantes crear parejas de prácticas sin repetir compañeros anteriores y que también permite emparejar automáticamente a los estudiantes desde un primer momento o cuando no lo hayan hecho ellos mismos al acabar un cierto plazo. Además, el módulo de cuestionarios reconoce estas parejas y les permite ver la solución del compañero así como adjuntarle comentarios, permitiendo la resolución en equipo sin necesidad de coincidir en el tiempo ni el espacio.

¹www.moodle.org

2.2. Estrategia de evaluación de la asignatura

Diseño y Administración de Bases de Datos² es una asignatura opcional correspondiente a objetivos formativos comunes de la categoría 2.2 de [2], que se cursa durante el quinto semestre de la *Ingeniería Informática* y en el tercero del *Master en Tecnologías de la Información* de la Facultad de Informática de Barcelona. Esta asignatura está siendo objeto de mejoras desde 2005 en el marco de los proyectos de mejora docente tanto de la propia universidad, como del gobierno autonómico. En su estado actual, los estudiantes trabajan en la asignatura asistiendo a dos sesiones (2 horas + 1 hora) de teoría y una sesión (2 horas) de laboratorio a la semana, además de unas horas más de trabajo personal. En todos los casos, exceptuando la primera sesión, las sesiones de laboratorio proporcionan una calificación. La evaluación de la asignatura se basa mayoritariamente, con un peso del 80%, en las pruebas realizadas en parejas durante las sesiones de laboratorio (el 20% restante corresponde a la calificación obtenida en el examen individual realizado al final del semestre). Generalmente, en cada sesión, los estudiantes, además de responder un cuestionario LEARN-SQL, tienen que contestar algunas preguntas complementarias en un formulario en papel para reforzar la validez de la evaluación automática. Habida cuenta del peso de la nota de laboratorio, los estudiantes cambian de pareja varias veces durante el curso (normalmente 5 veces). También se tiene en cuenta la dificultad e importancia de cada sesión, asignándose a cada una un peso de 1, 2 ó 4.

La dedicación autónoma de los estudiantes, aparte del estudio, se canalizaba hasta el curso 07/08 por medio de ejercicios propuestos en una lista de problemas (sin soluciones) a disposición de los estudiantes. A partir del curso 08/09, los estudiantes acceden a los ejercicios a través de cuestionarios en LEARN-SQL. Durante una semana, los estudiantes tienen acceso a través de internet a un cuestionario de entrenamiento para estudiar y practicar un tema. Siete días después de la publicación de este cuestionario de entrenamiento, en sesión cerrada de laboratorio, se lleva a cabo en el aula la correspondiente prueba puntuable. A partir del curso 09/10, dado el poco peso del examen final (únicamente el 20%),

²<http://www.fib.upc.edu/es/infoAca/estudis/assigatures/DABD.html>

para estimular la dedicación personal a la asignatura e incidir en los aspectos teóricos de la misma, se ha puesto en marcha una actividad donde los estudiantes proponen preguntas aptas para figurar en un examen (descrita en [6]). Con esta actividad se pretende que, además de resolver el cuestionario semanal, los estudiantes estudien el material teórico para estar en disposición de formular dichas preguntas.

3. Motivación, objetivos y dificultades

En general, a los estudiantes, les pedimos que adquieran ciertos conocimientos a nivel meramente memorísticos y otros a un nivel más profundo que les permita no solo repetirlos, sino también aplicarlos a nuevas situaciones y casuísticas (correspondiente a los niveles de conocimiento y aplicación en [4], respectivamente). Para los primeros es suficiente la lectura y el estudio, mientras que para los segundos es imprescindible la aplicación y práctica. En el caso de las competencias genéricas (igual que muchas otras), está claro que no podemos pedirles un conocimiento basado en la simple memorización de una serie de definiciones y consejos (que no tendría demasiado sentido), sino que hemos de pedirles que practiquen y adquieran una serie de actitudes.

El problema es que estas competencias genéricas, por el hecho de serlo, son totalmente independientes de la materia propia de la asignatura y acostumbran a estar más relacionadas con actitudes que con aptitudes o conocimientos explícitos. Esto lleva a que su aprendizaje sea siempre colateral o complementario al aprendizaje de la materia en cuestión y consecuentemente, su evaluación debe ser también totalmente independiente de la evaluación de los conocimientos técnicos de la asignatura. Parece evidente que no tiene demasiado sentido añadir preguntas a un supuesto examen final sobre un tema como el trabajo en equipo, pero quizás el caso más claro sea el de la competencia de lengua extranjera. No se trata de hacer clases de inglés y añadir preguntas de gramática en el examen final, sino de utilizar de alguna manera el inglés a lo largo de la asignatura y evaluar como lo utilizan en las actividades evaluatorias de las competencias técnicas. Idealmente, debería hacerse de forma similar con el resto de competencias genéricas.

La forma de introducir la competencia “trabajo en equipo” parece clara, ya que podemos pedir a los es-

tudiantes que realicen cualquier actividad en equipo. Sin embargo, esta competencia no puede evaluarse simplemente viendo los resultados de la actividad (como podemos hacer con el inglés leyendo una respuesta de examen o informe de práctica en esa lengua). Por ejemplo, el resultado de una actividad puede ser perfecto sin necesidad de que participen todos los miembros del equipo o totalmente al contrario, obtener un resultado desastroso a nivel técnico habiendo realizado un buen trabajo en equipo. Si el profesor únicamente valora el resultado de la actividad no podrá evaluar la competencia genérica en sí misma.

Así pues, la principal razón para poner en marcha esta experiencia fue la detección de una falta de información por parte del profesor para evaluar ciertas competencias genéricas. En el caso concreto del “trabajo en equipo”, los propios compañeros tendrán siempre mucha más información que el profesor y, en cualquier caso, la información que tendrán será siempre de primera mano.

La revisión o evaluación entre iguales está reconocida como un deber muy importante de los profesionales del futuro que intenta “aceptar y promover la revisión objetiva, crítica y documentada del trabajo de los demás” [8]. Además de esto, promueve la reflexión crítica sobre el trabajo propio y el de los demás, al mismo tiempo que permite que los estudiantes obtengan información útil sobre ellos de sus propios compañeros.

Como todas, esta técnica de evaluación también presenta algunos inconvenientes. En [3] se describen cuatro tipos de evaluación sesgada que pueden observarse en la evaluación entre iguales:

- (I) Puntuación por amistad: Se asigna una puntuación más elevada a los amigos.
- (II) Coincidencia de puntuaciones: Todos los miembros del grupo dan la misma puntuación al resto.
- (III) Puntuación de decibelios: Los estudiantes más populares o con personalidades dominantes reciben puntuaciones más altas que el resto.
- (IV) Puntuación parásita: Estudiantes que no han contribuido lo suficiente se benefician del hecho que el grupo reciba una única puntuación (este hecho no es debido puramente a la pun-

Category	4	3	2	1
Contributions	Always provides useful ideas when participating in the group discussion.	Usually provides useful ideas when participating in the group discussion.	Sometimes provides useful ideas when participating in the group discussion.	Rarely provides useful ideas when participating in the group discussion.
Problem Solving	Actively looks for and suggests solutions to problems.	Builds on solutions suggested by others.	Does not suggest solutions, but is willing to try out solutions suggested by others.	Does not try to solve problems or help others solve problems. Lets others do the work.
Attitude	Always has a positive attitude towards the task.	Often has a positive attitude towards the task.	Usually has a positive attitude towards the task.	Often has a negative attitude towards the task.
Preparedness	Brings needed materials to class and is always ready to work.	Almost always brings needed materials to class and is ready to work.	Often brings needed materials but sometimes needs to settle down and get to work.	Often forgets needed materials or is rarely ready to get to work.
Working with others	Almost always listens to, shares with, and supports the efforts of others.	Usually listens to, shares, with, and supports the efforts of others.	Often listens to, shares with, and supports the efforts of others.	Rarely listens to, shares with, and supports the efforts of others.
Focus on the task	Consistently stays focused on the task and what needs to be done.	Focuses on the task and what needs to be done most of the time.	Focuses on the task and what needs to be done some of the time.	Rarely focuses on the task and what needs to be done. Lets others do the work.

Figura 1: Rúbrica para la evaluación del trabajo en equipo

tuación entre iguales, sino al hecho de asignar puntuaciones a grupos en lugar de individuos).

El problema (I), intentamos evitarlo generando parejas de prácticas de forma aleatoria (excepto la primera pareja del curso, que permitimos que la escojan para evitar que los habituales abandonos tempranos generen problemas o tensiones innecesarias). Respecto a (II), pedimos expresamente a los estudiantes que asignen puntuaciones diferentes a todos sus compañeros (el módulo Moodle que hemos desarrollado -cuyas funcionalidades se explican brevemente en la siguiente sección- nos ayuda a garantizar que lo hacen de esta manera). Por contra, pensamos que (III) es difícilmente evitable, pero intentamos minimizarlo dándoles pautas claras de evaluación (tal como se explica en la sección 4) y haciendo que la puntuación sea totalmente anónima. Finalmente, (IV) no tiene por qué aparecer en la evaluación del trabajo en equipo, si cada estudiante puntúa a los individuos y no a los equipos con los que ha trabajado. En todo caso la puntuación parásita afectaría en nuestro caso a la evaluación de las competencias técnicas, ya que las parejas sí que reciben una puntuación única en cada práctica. Para minimizar este efecto, cada estudiante trabaja, en general, con cinco parejas diferentes a lo largo del curso.

4. Caso de estudio

Con la idea de captar la atención de los estudiantes y hacerles reflexionar sobre su actitud hacia el trabajo en equipo, les mostramos una “rúbrica” (*fal-*

se friend de la palabra inglesa *rubric*, que significa “conjunto de instrucciones”). La figura 1 muestra la tabla de conceptos y puntuaciones que se dio a los estudiantes. Ésta fue inicialmente obtenida de Rubistar³ y posteriormente modificada para que se adaptase mejor al funcionamiento de la asignatura. De esta manera, se les pedía a los estudiantes que analizaran el comportamiento de sus compañeros (y el suyo propio) desde seis puntos de vista diferentes:

Contribución: Aporta ideas en las discusiones.

Habilidad para la resolución de problemas:

Busca y sugiere soluciones.

Actitud: Tiene una actitud positiva hacia la tarea a realizar.

Preparación: Hace los entrenamientos con anterioridad a la sesión puntuable.

Capacidad para trabajar en equipo: Escucha, comparte ideas y valora el esfuerzo de los otros.

Capacidad de concentración: Dedicar continuamente su atención a la tarea que se está haciendo en cada momento.

El primer semestre que llevamos a cabo la experiencia les dimos la rúbrica el día del examen sin previo aviso y les pedimos en ese mismo momento que puntuaran a sus compañeros de prácticas (normalmente entre 4 y 5 estudiantes diferentes) de forma

³<http://rubistar.4teachers.org>

Puntúa a tus compañeros de prácticas valorando los aspectos siguientes:

- Contribución: Aporta ideas en las discusiones
- Habilidad para la resolución de problemas: Busca y sugiere soluciones
- Actitud: Tiene una actitud positiva
- Preparación: Hace los entrenamientos con anterioridad a la sesión puntuable
- Capacidad para trabajar en equipo: Escucha, comparte ideas y valora el esfuerzo de los otros
- Capacidad de concentración: Dedicar continuamente su atención a la tarea que se está haciendo en cada momento

Tu puntuación, Estudiante (Coincidencia) Comentario

3 u 6 ((T.Equipo A de Creador de Equipos 1))		
5 u 8 ((T.Equipo D de Creador de Equipos 2))		
- u 7 ((T.Equipo D de Creador de Equipos 3))		
- u 4 ((T.Equipo D de Creador de Equipos 4))		
- u 3 ((T.Equipo A de Creador de Equipos 5))		

Seleccionar todos

Estudiante	Puntuación	Comentario
u 8	5	

Enviar

Puntuación mínima: 0
Puntuación máxima: 10
Puntúa a todos con diferentes valores

Figura 2: Captura de pantalla del módulo Moodle desarrollado

totalmente anónima. Hacerlo el día del examen nos garantizaba la asistencia masiva de los estudiantes y el hecho de no avisarlos evitaba que se pusieran de acuerdo previamente. Por contra, impedía una reflexión más profunda sobre el propio comportamiento.

Así, en el siguiente semestre, decidimos cambiar de estrategia y les avisamos al inicio de curso de que serían evaluados por sus compañeros de prácticas (facilitando así la autoreflexión) y les dimos la rúbrica que deberían utilizar para hacerlo. Dado que la mayoría de parejas de prácticas son definidas de forma aleatoria, creemos que es realmente difícil que se pongan de acuerdo a la hora de puntuarse (prácticamente deberían ponerse de acuerdo todos los estudiantes dentro de un mismo grupo de la asignatura, aproximadamente 15 personas) y definimos el plazo de puntuación entre el último día de clase y el día del examen final. De esta manera, los estudiantes tenían más de dos semanas para puntuar a sus compañeros y podían hacerlo desde casa a través de la misma interfaz que habían utilizado a lo largo de todo el curso para el seguimiento de la asignatura, gracias a que desarrollamos un módulo de Moodle (cuya captura de pantalla se puede ver en la figura 2) que nos ayudó también a la gestión de las puntuaciones.

Además del hecho de controlar a qué compañeros puntuaba cada estudiante (en nuestro caso únicamente esos con los que haya realizado alguna práctica a lo largo del curso), el módulo Moodle que hemos

desarrollado ofrece las siguientes funcionalidades:

1. Control de la nota máxima y mínima que puede poner un estudiante a otro.
2. Control de repetición de nota por parte del mismo estudiante evaluador.
3. Función de agregación para el cálculo de la nota (promedio, máximo, mínimo o suma).
4. Normalización de las notas asignadas por cada estudiante.
5. Normalización de las notas resultantes para todos los estudiantes del curso.

La experiencia demuestra que los estudiantes únicamente participan en una actividad si ven algún beneficio o están realmente muy motivados. Dado que en nuestro caso no hemos querido modificar aún el método de evaluación, el resultado de la actividad resultaba únicamente en una modificación positiva de la nota final de la asignatura. De esta manera, la participación es totalmente voluntaria y los estudiantes que no lo hacen no se ven perjudicados, pero aún así incentivamos la participación.

5. Experimentación y resultados obtenidos

En primer lugar, cabe decir que tal como refleja la figura 3, la participación en la actividad ha sido to-

	09-10Q2	10-11Q1
Matriculados	24	34
Participantes	20	32

Figura 3: Participación en la actividad

	09-10Q2	10-11Q1
AVG por evaluador	8'33	8'05
STDDEV por evaluador	1'20	1'74
AVG por evaluado	8'27	8,12
STDDEV por evaluado	1'28	1'29

Figura 4: Puntuaciones proporcionadas por los estudiantes

talmente masiva. Los cuatro estudiantes que no la realizaron en la primera ocasión fueron aquellos que no se presentaron al examen, mientras que los dos que no lo hicieron en la segunda, fueron aquellos que abandonaron la asignatura antes del final del semestre. Cabe resaltar que en el segundo caso, la actividad se realizó desde casa y que en ningún momento estudiante alguno mostró reparos a su realización.

La figura 4 muestra los promedios obtenidos en cada caso. En la primera ocasión que se realizó la actividad no se controló que los estudiantes otorgaran la misma puntuación a todos sus compañeros, mientras que en la segunda ocasión (gracias al uso del módulo de Moodle desarrollado), sí que se pudo hacer. A pesar de eso, el promedio de puntuaciones otorgadas por cada estudiante (en la primera fila de la tabla) es muy similar en ambos casos, aunque la desviación estándar sí que aumente significativamente. Las dos últimas fila de la tabla muestran el promedio y la desviación estándar de las notas recibidas por cada estudiante. Vemos que en la segunda experiencia la desviación estándar de las notas recibidas es claramente inferior a la de las notas otorgadas (la diferencia en la primera columna no es tan relevante).

La figura 5 muestra las notas otorgadas por cada estudiante a sus compañeros el semestre 0910Q2. En las ordenadas vemos los valores de las notas (entre 0 y 10) y en las abscisas los diferentes estudiantes ordenados por la amplitud del intervalo de notas asignado (aquellos intervalos menores que 4 corresponden a estudiantes que por una razón u otra tuvieron

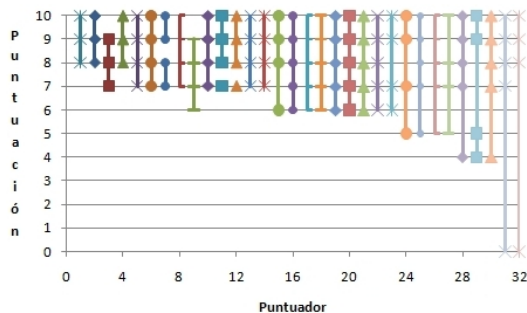


Figura 5: Puntuaciones otorgadas por cada estudiante

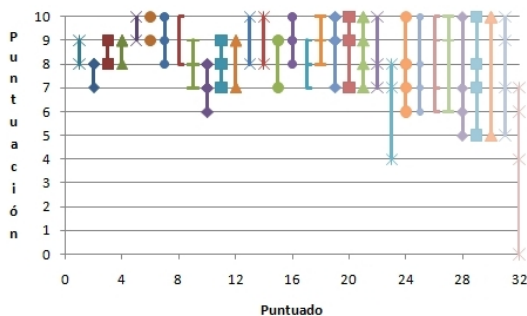


Figura 6: Puntuaciones recibidas por cada estudiante

menos de 5 compañeros de prácticas). Se puede ver claramente la tendencia a otorgar notas altas también observada en el promedio, aunque ahora también vemos que un 25 % de estudiantes también otorgan notas por debajo de 6 o incluso de 5. Nótese que pese a la restricción introducida para que hayan de asignar puntuaciones diferentes a todos sus compañeros, al tener un máximo de 5 compañeros, podrían puntuarlos a todos entre 6 y 10. Creemos que los estudiantes tienen tendencia a otorgar las puntuaciones máximas a sus compañeros (probablemente para no perjudicarlos), excepto en casos de actitud claramente reprochable (como por ejemplo no asistir a una sesión puntuable sin razón aparente ni aviso previo).

La figura 6 muestra las notas recibidas por cada estudiante. Igual que en la figura anterior, en las ordenadas vemos los valores de las notas (entre 0 y 10) y en las abscisas los diferentes estudiantes ordenados por la amplitud del intervalo de notas recibido. Se puede observar que en más de la mitad de los casos el tamaño del intervalo de notas recibidas es menor o igual que 2. Es decir, las notas de cada estudiante están claramente agrupadas. A pesar de

que casi todos los estudiantes (28 de 32) han otorgado 4 o 5 notas diferentes, la gran mayoría (25 de 32) ha recibido únicamente 2 o 3 notas diferentes, o lo que es lo mismo, han recibido más de dos notas repetidas y las no repetidas son consecutivas. Además, hay algún estudiante con un intervalo mayor de 4, pero ninguno con más de 4 notas diferentes. Todo esto nos lleva a la conclusión de que las puntuaciones otorgadas a cada estudiante son coherentes con su comportamiento a lo largo del curso, ya que, en general, no se aprecian grandes discrepancias en las opiniones de los diferentes compañeros que ha tenido cada uno.

No obstante, hemos observado también que las notas otorgadas a la primera y última pareja del curso son sensiblemente superiores a las otorgadas a las otras tres parejas (8'82 y 7'76 respectivamente). Realizando un test de *Student*, vemos que la probabilidad de que esta diferencia en el promedio se deba al azar es menor que 0'02%. Seguramente diversos factores (como la dificultad de la práctica) podrían haber influido en esta diferencia, pero nosotros opinamos que la causa principal es que la primera y última pareja de prácticas la escogieron ellos en la mayoría de casos, mientras que las otras tres parejas fueron asignadas totalmente al azar.

6. Conclusiones e intenciones futuras

Tras detectar la dificultad de evaluar competencias genéricas, en este artículo hemos presentado la utilización de la puntuación entre iguales para paliar esta dificultad. De esta manera, hemos explicado como en los dos últimos semestres hemos propuesto a nuestros estudiantes que evalúen el trabajo en equipo de sus compañeros. Pensamos que de esta manera se verán aún más motivados a tener una actitud positiva ante el trabajo en equipo. Además, el hecho de afectar la nota del compañero también debe mejorar su responsabilidad. Creemos que es importante que aprendan a evaluar el trabajo de los compañeros (y consecuentemente el suyo propio) de forma crítica y responsable.

La participación (totalmente voluntaria) en esta actividad ha sido masiva, no planteándose ningún tipo de duda por parte de los estudiantes (pensamos que la rúbrica es suficientemente clara en los aspectos a evaluar). Además, la distribución de notas obtenidas nos parece coherente en el sentido de que

	09-10Q2	10-11Q1
AVG por curso	4'91	5'07
STDDEV por curso	1'77	1'39

Figura 7: Puntuaciones normalizadas

los estudiantes que reciben notas altas las reciben de la mayoría de sus compañeros y lo mismo ocurre con los que reciben notas bajas (uno de los cuales sí expresó su descontento con las puntuaciones obtenidas). No obstante, hemos detectado una tendencia a otorgar únicamente las puntuaciones más altas disponibles y un cierto favoritismo hacia los compañeros de práctica escogidos por el propio estudiante frente a los asignados de forma aleatoria.

Actualmente hemos propuesto la inclusión de las puntuaciones entre compañeros en la evaluación de la asignatura con un peso del 10%. Para evitar favoritismos, intentaremos generar todos los equipos de prácticas al azar (en lugar de únicamente 3 de los 5). Además para corregir la tendencia de los estudiantes a otorgar puntuaciones altas a sus compañeros proponemos también normalizar las notas otorgadas por cada estudiante. La figura 7 muestra cual sería el resultado en el promedio de notas obtenidas por todos los estudiantes (normalizando las notas otorgadas por cada estudiante con una esperanza de 5 y una desviación estándar de 2'75) en las dos experiencias realizadas. El objetivo de esta normalización no es tanto corregir directamente la distribución de las notas (que no parece demasiado significativo en estos dos casos), sino corregir la actitud de los estudiantes y que sean ellos quienes discriminen más al saber que en cualquier caso sus puntuaciones serán normalizadas.

Reconocimientos

Este trabajo ha sido financiado con los proyectos de "Millora de la Docència" de la *Universitat Politècnica de Catalunya* y 2009MQD00251 de la *Generalitat de Catalunya*. Queremos agradecer también a Carlos Romeu su dedicación en la implementación del módulo Moodle que ha facilitado la recogida de datos.

Referencias

- [1] ALIER, M. *A social constructionist approach to learning communities: Moodle in Open Source for Knowledge and Learning Management*. Idea Group, 2007.
- [2] ANECA (AGENCIA NACIONAL DE EVALUACIÓN DE LA CALIDAD Y ACREDITACIÓN). *Libro Blanco: Título de grado en ingeniería informática*. Omán Impresores, 2005.
- [3] BROWN, S., AND KNIGHT, P. *Assessing learners in higher education*. Teaching and higher education series. Routledge Falmet, 1994.
- [4] B.S. BLOOM. *Taxonomy of educational objectives*. David McKay, 1956.
- [5] A. ABELLÓ, E. RODRÍGUEZ, T. URPI, X. BURGUÉS, M. CASANY, C. MARTÍN, AND C. QUER. LEARN-SQL: Automatic Assessment of SQL Based on IMS QTI Specification. In *Proceedings of 8th IEEE International Conference on Advanced Learning Technologies (ICALT)* (2008), pp. 592–593.
- [6] A. ABELLÓ, X. BURGUÉS, AND E. RODRÍGUEZ. Utilización de glosarios de Moodle para incentivar la participación y dedicación de los estudiantes. In *Actas de las XVI Jornadas de Enseñanza Universitaria de la Informática (JENUI)* (2010), pp. 309–316.
- [7] CONSELL DE GOVERN DE LA UPC. Marc per al disseny i la implantació dels plans d'estudis de grau a la UPC. Document DG 16/4 2008, 2008. <http://www.upc.edu/bupc/hemeroteca/2008/b105/16-04-2008.pdf>.
- [8] KERN, V. M., SARAIVA, L. M., AND DOS SANTOS PACHECO, R. Peer review in education: Promoting collaboration, written expression, critical thinking and professional responsibility. *Education and Information Technologies* 8, 1 (2003), 37–46.
- [9] SÁNCHEZ, F., SANCHO, M., BOTELLA, P., GARCIA, J., ALUJA, T., NAVARRO, J., AND BALCÁZAR, J. *TICAI 2008: TICs para a aprendizagem da engeharia*. IEEE, Sociedad de Educación: Capítulos Español, Portugués y Colombiano, 2009, ch. Competencias profesionales para el Grado en Ingeniería Informática, pp. 147–154.

Una recapitulación sobre la autoevaluación de los alumnos en estudios de Informática: formas, utilidad y aplicación

Agustín Cernuda del Río, Miguel Riesco Albizu

Departamento de Informática
Universidad de Oviedo
C/ Calvo Sotelo, S/N
33007 Oviedo
{guti, albizu}@uniovi.es

Resumen

Se realiza una recapitulación ordenada sobre diversos métodos y planteamientos para la autoevaluación de los alumnos, aplicados en la práctica a la docencia de la Informática en España en los últimos tiempos y presentados en JENUI, realizando además una reflexión sobre sus respectivos objetivos y beneficios, que pueda servir como orientación y guía de aplicación. Se estudia el estado del arte, los resultados alcanzados y las conclusiones documentadas. Se presentan y comentan también experiencias realizadas y documentadas por estos y otros autores, como apoyo a dichas reflexiones.

Summary

This paper offers an organized summing-up of several methods and approaches for student self-assessment, actually applied to computer science teaching in Spain in recent years and presented at JENUI, making some reflections on their respective goals and benefits, that can be of interest as a hint for applying them. The state of the art, reached results and documented conclusions are studied. Several experiences are presented and commented as well, made by these and other authors, supporting these reflections.

Palabras clave

Autoevaluación, revisión, estado del arte, aplicación, resultados experimentales.

1. Motivación

Este trabajo pretende ofrecer una panorámica de la autoevaluación de los alumnos como método de aprendizaje; panorámica que no pretende centrarse

en definiciones teóricas sino, por el contrario, en la experiencia real llevada a cabo durante los últimos años por la comunidad docente en Informática y presentada en JENUI, señalando y resumiendo los resultados a los que se ha llegado.

Tradicionalmente, la evaluación se ha venido considerando sobre todo como un medio para poder emitir un juicio sobre los resultados del proceso formativo y acreditar la capacitación de un alumno (la denominada *evaluación sumativa*). Hace ya tiempo que este enfoque se somete a revisión, y el profesorado es consciente de que la evaluación tiene otras vertientes importantes; se habla, pues, de la *evaluación formativa* como un medio didáctico, que permite al alumno tanto conocer la marcha de su proceso de aprendizaje como, directamente, aprender.

No sólo eso; el esquema clásico de evaluación sumativa se centraba en una prueba final, una sola verificación individual de habilidades y conocimientos adquiridos por el alumno (y esta herencia resulta evidente, por ejemplo, echando un vistazo a la organización del calendario escolar, con sus “períodos de exámenes”). No obstante, hace ya mucho tiempo que se trabaja con otros enfoques temporales (evaluación continua), otros métodos (proyectos, equipos), otros tipos de examen (prácticos, tests, portafolios, presentaciones orales...)

El panorama de la evaluación es, pues, muy amplio y variado, y existen infinidad de propuestas y trabajos sobre el particular. Esto puede plantear dificultades a la hora de llevar a la práctica una experiencia docente, pues no resulta fácil elegir entre tantas ideas, con fortalezas y debilidades diversas, con requerimientos y necesidades variadas, con desigual consumo de recursos, y cuya viabilidad, en consecuencia, resulta incierta en según qué casos.

Tomando esto como punto de partida, parece oportuno realizar también trabajos de revisión y recapitulación [13], que faciliten una visión general del estado de la cuestión. Se trata de ofrecer una reflexión sobre lo que ya se ha hecho y dar a conocer de forma organizada lo que ya se ha probado, con la esperanza de que tal reflexión facilite tanto la adopción de esas técnicas como posteriores investigaciones.

El presente artículo pretende ser uno de tales trabajos, centrándose sólo en la cuestión de la *autoevaluación* (RAE: evaluación que alguien hace de sí mismo o de algún aspecto o actividad propios [20]), es decir: en el interés didáctico que puede tener el que un alumno se vea implicado de forma explícita en la revisión del trabajo que se le encomienda. Todo alumno, supuestamente, revisa aquello que hace; pero no siempre se le señala esto como una actividad específica.

Se trata aquí de reflexionar sobre el papel del alumno como evaluador de su propio trabajo, sobre el interés que tal actividad puede tener en su vertiente formativa y sumativa, y en las posibilidades que se ofrecen al profesor para utilizar este recurso. Se comentarán diversas experiencias, poniéndolas en relación con los aspectos positivos y negativos de la autoevaluación, y analizándolas también desde el punto de vista de la aplicabilidad que tales experiencias puede tener en cada caso. En especial, si la aplicabilidad importa, y dada la posible diferencia entre sistemas educativos y ámbitos socioculturales, parece procedente realizar esa reflexión con especial atención a las experiencias e investigaciones realizadas directamente por docentes de la informática en España (y en este ámbito son las JENUI la fuente principal de tales conocimientos).

2. Interés de la autoevaluación

La primera cuestión, no trivial, es si realmente merece la pena emplear recursos en actividades de autoevaluación.

Los resultados obtenidos apuntan a que en general así es. En [3] se enumeran algunos beneficios genéricos: responsabilizar al alumno, proporcionarle realimentación sobre sus puntos débiles, motivación... En [12] se realiza un estudio bastante riguroso probando dos enfoques de autoevaluación distintos (pruebas objetivas y la

herramienta Duck), y se concluye que introducir actividades de autoevaluación tiene efectos significativos en el aprendizaje, por encima de la forma específica de autoevaluación que se utilice.

La autoevaluación, aparte de efectos beneficiosos genéricos para el aprendizaje, puede resultar de interés por motivos más concretos y específicos. En [3], por ejemplo, se justifica la relación de este método con el enfoque docente del EEES (que exige precisamente mayor trabajo autónomo y responsabilidad, cosas que encajan con la autoevaluación). En esta línea, en [24] se plantea la necesidad de proporcionar medios de autoevaluación ligados al trabajo autónomo del alumno, y en especial en la actividad experimental de este en relación con la simulación, que es una forma de autoevaluación especialmente indicada en algunos casos (en este se aplica a asignaturas de redes).

Los resultados de aprendizaje son vitales, pero el alumno puede llegar a esos resultados por un camino más o menos satisfactorio según su propia experiencia; existen indicios documentados [4][6][15][22] de que los propios alumnos valoran muy positivamente el uso de estas técnicas.

3. Variables en la autoevaluación

3.1. Autoevaluación vs evaluación continua

En este campo, aparecen mezcladas frecuentemente ideas que tienen relación, pero son distintas. Un caso claro es el de la autoevaluación y la evaluación continua (valoración del proceso de aprendizaje del estudiante a partir del seguimiento continuo del trabajo que realiza [9]), que en la literatura se funden con cierta asiduidad.

En principio, no hay que confundir evaluación continua y autoevaluación. Es cierto que la realimentación al alumno suele aparecer en ambos instrumentos, y que esa realimentación temprana es uno de los beneficios asociados frecuentemente a la evaluación continua; pero son independientes. No hay que olvidar otro aspecto de la evaluación continua; la realimentación *al profesor*, que gracias a ella puede detectar de manera temprana dificultades de comprensión en la clase [21].

3.2. Autoevaluación individual / cruzada / en grupo

Otra decisión que hay que tomar es si se realizan actividades de autoevaluación individual, o si se considera al alumnado en conjunto como el agente que se evalúa a sí mismo, de modo que cada alumno revise el trabajo de otros, además de (o en lugar de) el suyo. Hay varias formas de hacer esto.

Se han realizado experiencias con bastante profundidad sobre la evaluación cruzada (también denominada “revisión entre iguales” o “co-evaluación”). En [22] se refieren los resultados de su aplicación intensiva, durante cuatro cursos, para pruebas de respuesta corta, de test y de trabajo práctico (resultando más provechosa, en concreto, la primera de ellas).

Una característica de la autoevaluación cruzada entre iguales es que, por lo general, implica un apreciable coste de gestión, y es frecuente que se aborde con apoyo tecnológico [22]. En algunas aplicaciones concretas no es así; por ejemplo, en [11] se menciona su aplicación a las presentaciones orales, donde la “revisión entre iguales” ocurre por definición y de manera natural (un orador siempre es evaluado por su público).

Otro ámbito en el que la autoevaluación cruzada tiene especial significado es en el seno de equipos de trabajo (“evaluación por compañeros”); y también cabe plantearse el interés de relacionar estas evaluaciones con el papel que cada cual desempeña en el equipo (como se hace, por ejemplo, en [14]). ¿Cómo evalúa el jefe a los demás? ¿Cómo lo evalúan sus compañeros a él?

Otra forma de aplicar la autoevaluación en el seno de un grupo es entenderla como una actividad más de aprendizaje colaborativo; los alumnos realizan actividades por separado, para después realizar una corrección en grupo durante la puesta en común (como se plantea, por ejemplo, en [4]).

3.3. Autoevaluación formativa / sumativa

A lo largo de este artículo, y en muchos de los citados, se plantea la utilidad formativa de la autoevaluación. Puede surgir la cuestión de darle además un valor sumativo, teniendo esta calificación un peso en la calificación final.

Ante los evidentes conflictos de interés que pueden aparecer en múltiples formas, parece oportuno

establecer en ese caso limitaciones y mecanismos preventivos, como se hace en [14]. Aun así, la autoevaluación (generalmente cruzada) como medio sumativo rutinario está sujeta a dudas [3]; en [17] se muestra experimentalmente, y con cierto rigor estadístico, que estas calificaciones difirieron significativamente de las del profesor en un estudio realizado para calificar trabajos de respuesta abierta.

El problema de la representatividad de métodos de evaluación diversos se complica, además, porque se ha visto que diversos métodos de evaluación arrojan calificaciones distintas para un mismo alumno [18]; la correlación entre ellos hace desaconsejable ceñirse a uno, y los autores recomiendan combinar varios. Con la autoevaluación hay que tener esto en cuenta, más aún por la mencionada discrepancia entre calificaciones “de alumno” y “de profesor”.

Otro factor a recordar es que si se recurre a herramientas *online* de autoevaluación con propósito sumativo es necesario considerar en ellas aspectos de seguridad [15].

Teóricamente, puede darse una evaluación continua puramente sumativa, sin realimentación alguna al alumno. O una *autoevaluación* “final” y única; recordemos que son conceptos independientes. En [25] se discute la necesidad de desligar los conceptos de *información* (i.e., *realimentación*) continuada y *calificación* continuada: la evaluación sumativa debe ser muy precisa y fiable, pero en la evaluación formativa importa la prontitud, y no tanto la precisión. El docente tendría, pues, que aclarar qué beneficios espera obtener de la autoevaluación, y plantearse si la pone en relación (o no) con una estrategia de evaluación continua. Son dos recursos de los que puede disponer por separado. Debe clarificar también el propósito (formativo o sumativo) de sus métodos, y en razón de sus prioridades dar forma al proceso.

3.4. Forma de autoevaluación

Una variable obvia, pero compleja, es la forma que adopta el propio ejercicio de autoevaluación. Se puede recurrir a *tests* clásicos, ejercicios de los que se facilita posteriormente la solución, actividades en grupo... En el caso de disciplinas técnicas muy concretas, la propia herramienta de desarrollo y / o simulación da pautas para la autoevaluación [24]. En otros casos, las mismas

herramientas de verificación, como pruebas unitarias o pruebas de cobertura, pueden constituirse también en herramientas de autoevaluación [16].

4. Aspectos propios de la Informática

4.1. Autoevaluación vs calidad

Otra dimensión en la que puede haber confusión, y que el docente debe clarificar, es el propósito de la autoevaluación en lo que se refiere a las habilidades, y en especial en ciertas habilidades muy presentes en la Informática.

Por una parte, se podría pensar en la autoevaluación como un medio para que el alumno revise su volumen de conocimientos, o dicho de otro modo, su *cobertura* del temario, mediante cuestionarios que ayuden al alumno a diagnosticar su grado de conocimiento de los conceptos que se manejan, y también avanzar en su comprensión.

Pero por otra parte se podría pensar en la autoevaluación como una verificación de la calidad del trabajo de un alumno, de la solidez de sus hábitos, de lo avanzado de sus habilidades. Es conocido, por ejemplo, que un alumno puede tener los conocimientos necesarios de teoría y algorítmica y ser capaz incluso de aplicarlos a la resolución de un problema, pero ser, sin embargo, un mal programador, porque el código que produce cae en los errores clásicos de la disciplina (deficiente documentación, ilegibilidad, inconsistencia, mala nomenclatura, mantenimiento imposible).

Por tradición, tanto los alumnos como los profesores son conscientes de la importancia de los conocimientos, y la autoevaluación no ha sido ajena a ello (las autoescuelas, mismamente, utilizan este método desde tiempo inmemorial). Desde hace un tiempo, el mundo docente también hace hincapié en las habilidades y / o competencias [23], pero la autoevaluación aplicada a estos aspectos no ha sido, quizás, igualmente cultivada. Se puede conseguir la comprensión del temario, se puede llegar a la solución de un problema, se puede alcanzar aparentemente el objetivo marcado, y sin embargo haber fracasado, muchas veces inadvertidamente, en la elegancia o solidez del trabajo realizado para llegar a ello [19].

La calidad en el trabajo propio puede ser muy difícil de inculcar por la vía del temario; requiere de una cierta forma de compromiso personal, de actitud autocrítica, y de una vigilancia permanente. Parece una dimensión ideal para aplicar técnicas de autoevaluación, que precisamente convierte al alumno en crítico de sí mismo y lo implica intelectualmente en el proceso de revisión.

4.2. Autoevaluación vs control de proyectos

En la docencia de la Informática, el término “proyecto” puede aparecer en referencia a cierto método docente de uso general (*project based learning*, PBL) y también como una actividad consustancial a la ingeniería informática y específica de esta; actividad que puede desarrollarse –o no– en el marco de una metodología de PBL.

Ciñéndonos a los proyectos “de informática”, en todo proyecto de ingeniería es fundamental la actividad de control. Es la que permite diagnosticar la marcha del proyecto, detectar desviaciones y actuar en consecuencia. Sin embargo, cuando los alumnos están inmersos en el desarrollo del proyecto (que por sí mismo es un reto intelectual para alguien en formación) no resulta fácil conseguir que planifiquen el proyecto con los adecuados hitos, que realicen las actividades de control y que replanifiquen o actúen ante las desviaciones. Las actividades de gestión del proyecto, y en especial las de control, son una especie de actividad paralela al desarrollo, en tensión con este, y puede resultar difícil que un alumno se desdoble en estos papeles antagonistas cuando está concentrado en la consecución de objetivos funcionales.

Puede ser que estos mismos alumnos, que encuentran dificultades para atender todos los aspectos de la gestión de su propio proyecto (con los que aún se están familiarizando) sí sean receptivos a actividades de autoevaluación. Ese desdoblamiento que encuentran difícil abarcar resulta, sin embargo, natural cuando se plantea como evaluación, puesto que ese esquema de control “externo” a sus actividades académicas sí que no es nuevo para ellos en ningún momento.

5. El coste de la autoevaluación

5.1. Factores de coste

Uno de los problemas notables de la autoevaluación es su coste de implantación, debido a varios factores (en [22] se ofrece una cuantificación de algunos de tales costes).

- La gestión por parte del profesor. La recogida y gestión de las autoevaluaciones, por sí mismas, ya puede representar un coste nada despreciable. Este coste puede aumentar drásticamente si se abordan esquemas que requieren mucha organización, como la evaluación por iguales: anonimato, distribución de ejercicios, alegaciones...
- La corrección. Dependiendo del caso, la autoevaluación puede requerir una verificación posterior, puesto que no se puede dar por seguro que el alumno sea capaz de autoevaluarse correctamente. Esto hace que el profesor deba corregir las autoevaluaciones y, en su caso, dar realimentación al alumno, lo que se traduce en mayor trabajo para el docente.
- El soporte de herramientas. Muchos enfoques aquí presentados requieren herramientas de simulación o evaluación muy específicas del dominio del problema [1][2][5][7][10][12][19][21][22][24]. En otros casos no hay nada de esto, pero se utiliza soporte tecnológico para tareas de gestión ya mencionadas (distribución y recogida de ejercicios), o para la mera realización (en línea) de los ejercicios de autoevaluación. En todo caso, esto implica también un coste, temporal y / o económico, para la puesta en marcha y mantenimiento de esas herramientas. En [2], por ejemplo, se hace especial hincapié en el coste de aprendizaje de la herramienta y en el de la generación de contenidos, que exige alcanzar una masa crítica para obtener beneficios.

El coste es, pues, un factor a tener en cuenta. Sin embargo, no hay que perder de vista que el descartar ciertas implementaciones por su alto coste no implica descartar cualquier forma de autoevaluación. Curiosamente, en [25] y en [16] se propone, de hecho, la autoevaluación y la co-evaluación como medio para reducir para el profesor el coste de dar información con prontitud (a costa de precisión y fiabilidad). Por otra parte,

en ocasiones un enfoque *ligero*, es decir, fácil de implementar y gestionar (a costa de no ser exhaustivo), puede resultar de interés y aportar beneficios que lo justifiquen.

5.2. El enfoque ligero: un caso práctico

Como ejemplo cabe citar lo realizado en las prácticas de laboratorio de una asignatura de Estructuras de Datos y de la Información (EDI). Se trata de una asignatura anual de segundo curso de la titulación informática, una asignatura de cierta dificultad. A esta se suma el hecho de que requiere ciertas habilidades de programación, y EDI comparte con otras muchas asignaturas que resulta casi imposible superar las prácticas sin un trabajo continuado. Por ello se califican mediante evaluación continua.

Durante las prácticas, los alumnos desarrollan un proyecto individual, con una base común y una parte libre, que entregan en dos o más entregas parciales. Cada semana, se fijan objetivos concretos de desarrollo para la siguiente clase, y se hace una revisión rápida de lo realizado por los alumnos desde la semana anterior.

Esta evaluación continua en clase no suele permitir la revisión de los ejercicios de todos los alumnos todos los días, por razones de tiempo; y las admoniciones o directrices generales adolecen de falta de efectividad. Pero si no se hace un seguimiento estrecho y semanal de los objetivos, se corre el riesgo de que los alumnos se desvíen de los objetivos, quizás sin saberlo; y un par de semanas de desviación pueden tener efectos muy negativos en su planificación.

Alumno:.....

Sesión 04 – Gestión de errores / Lista

Funcionalidad

- La clase Sintactico incorpora tratamiento de errores
- Se detectan errores léxicos
 - Tokens incorrectos (ejemplo: un número)
 - Tokens incompletos (ejemplo: una cadena sin cerrar, y
 - Se detectan errores sintácticos y se genera los de toda la

Pruebas

- Se han realizado las pruebas unitarias pertinentes con JUnit
- Se ha probado todo lo indicado en el apartado "funcionalidad" efectivamente se genera en cada caso la excepción / error esperado
- Se tienen pruebas implementadas para todos los tipos de error
- Todas las pruebas son automáticas (no requieren del programa informe de JUnit)
- En las pruebas no se imprime NADA por pantalla
- Se ha probado a fondo la lista
 - Inserción de elemento en lista vacía
 - Borrado del último elemento que quedaba en la lista
 - Comprobación de que están los elementos insertados
- Se ha probado a fondo el iterador
 - Iteración por lista vacía
 - Iteración por lista con un solo elemento
 - Iteración por lista con varios elementos, comprobando adecuados
- Sé depurar con facilidad un caso de prueba de JUnit

Figura 1. Fragmento de hoja de autoevaluación

Surge entonces la idea de aplicar un enfoque de autoevaluación *ligero*. Aquí, autoevaluación y evaluación continua (apdo. 3.1) van de la mano, pero se persigue, ante todo, el control individual (3.2) de la marcha del trabajo propio (4.2). El propósito es puramente formativo (3.3) puesto que la vertiente sumativa se cubre sobradamente con el control periódico del profesor (aunque no revise el ejercicio *todas* las semanas). Ya que se trata, ante todo, de dar realimentación temprana y detectar hábitos erróneos (4.1), basta un modelo de lista de comprobación, que no necesita ser corregido por el profesor; sólo explicado para que el alumno lo cumplimente y saque sus propias conclusiones.

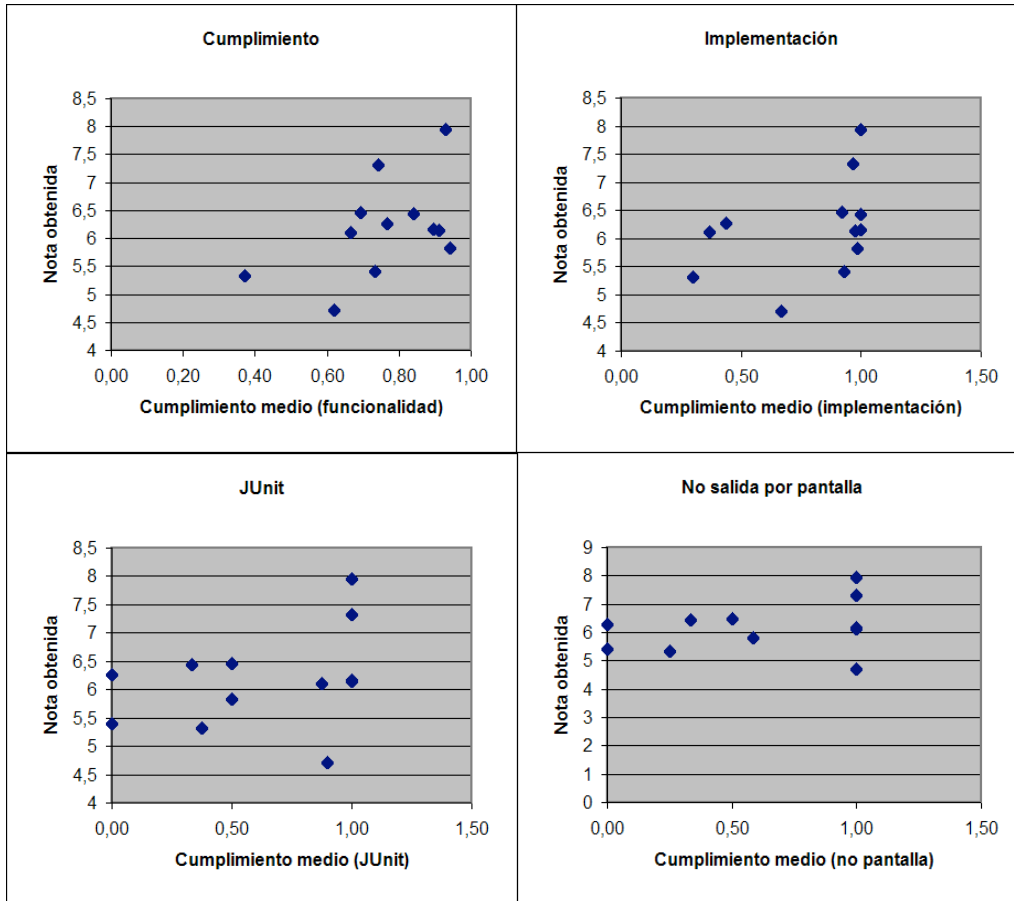


Figura 2. Nota vs cumplimiento funcional (correlación de Pearson: 0,498), cumplimiento de implementación (0,412), cumplimiento de JUnit (0,334) y cumplimiento de no salida por pantalla (0,296).

Durante el curso 2010-2011 se decidió poner en práctica, pues, el esquema de autoevaluación puramente formativa que consiste en:

- En cada clase se facilita a los alumnos una lista de comprobación, en papel (una cara, tamaño A5; figura 1).
- La hoja contiene elementos de respuesta simple (sí / no) referentes a los objetivos concretos de esa semana, y a la marcha de la implementación.
- Se busca que los alumnos sepan qué se esperaba de ellos para esa clase, y qué tipo de cosas habría examinado el profesor (de tener tiempo suficiente) para comprobar la marcha de su ejercicio.

- Esta hoja debe cumplimentarse, con el nombre del alumno, y entregarse al profesor, aunque queda claro que no influye en la calificación.

Con ello se busca en primer lugar que todos los alumnos, a título individual, sean conscientes de que están bajo observación; y de que, independientemente de que el profesor tenga tiempo o no de revisar su ejercicio, conozcan en qué punto deberían encontrarse su proyecto y sus conocimientos.

Por ejemplo, un aspecto en el que se hizo especial hincapié, de cara al éxito final de los alumnos en el desarrollo, fue la realización de pruebas unitarias mediante JUnit (relacionado con el apdo. 4.1). No sólo eso, sino que estas pruebas se hubiesen realizado de forma útil, es decir, automática. Los alumnos tienen una fuerte

tendencia a imprimir valores por pantalla y verificarlos manualmente, y se pretendía evitar ese comportamiento. Puesto que era de prever que costase bastante el que se habituaran a esta práctica nueva para ellos, se incluían todas las semanas preguntas referentes a esta cuestión.

Al final del semestre, se realizó un somero análisis de los datos obtenidos para ver si estas autoevaluaciones eran significativas y, en próximas ediciones, cabría insistir en su validez como predictores del éxito o fracaso en la asignatura.

La experiencia se realizó en un solo grupo de prácticas, con una veintena de alumnos, y después de descartar datos no utilizables (por razones de asistencia o similares) quedan 12 observaciones completas. Por tanto, no resulta indicado realizar un análisis estadístico muy profundo, y con los datos disponibles una simple prueba de correlación es lo más razonable.

De las encuestas se ha calculado el grado medio de cumplimiento semanal de cada alumno, siendo 0 una respuesta negativa y 1 una respuesta positiva. Se han agrupado los datos en cuatro bloques:

- Cumplimiento funcional. Se refiere al grado de cumplimiento de los objetivos funcionales señalados para esa clase.
- Implementación. Se refiere al grado de cumplimiento de los detalles de implementación sugeridos en la clase anterior para alcanzar la funcionalidad anterior.
- Realizadas las pruebas con JUnit. Se refiere al grado de cumplimiento de la realización de pruebas unitarias para lo que se ha implementado esa semana.
- No salida por pantalla. Se refiere específicamente a la pregunta sobre la existencia de mensajes de prueba por pantalla (se pretende que no haya ninguno y las pruebas sean plenamente automáticas).

Se decidió poner los datos que representan estas respuestas en relación con la calificación obtenida en la entrega parcial del primer semestre. Es sabido que modelar numéricamente algo tan complejo como un proceso de enseñanza / aprendizaje con alumnos, sujeto a tantos factores difíciles de medir, es muy arriesgado [8], pero en este caso era aún más difícil obtener resultados significativos por varias razones (el tamaño de la

muestra es muy pequeño; los valores que se representan no se prestan a una cuantificación sencilla; el proceso de calificación en sí mismo es, como todos, discutible y está por demostrar que los datos que arroja sean muy estables).

No habría sido raro, en estas circunstancias, obtener una correlación prácticamente nula; sin embargo, resultó ser consistentemente positiva (figura 2). De esto cabe deducir que, efectivamente, incluso una autoevaluación tan sencilla como esta tiene valor como elemento de control (véase apartado 4.2).

6. Conclusiones

Se ha hecho un recorrido de recapitulación por las investigaciones realizadas en los últimos años, ciñéndose a docencia en informática en España, sobre la aplicación práctica de la autoevaluación. Se ha presentado una recopilación organizada de los temas tratados y las principales conclusiones alcanzadas.

En primer lugar, cabe destacar la necesidad de separar claramente la autoevaluación de otros conceptos y métodos, como la evaluación continua, con los que frecuentemente se ve mezclada.

Se ha puesto de manifiesto la utilidad de la autoevaluación como medio formativo, renunciando a la precisión y fiabilidad a cambio de la prontitud. Como medio sumativo, hay indicios experimentales de que diverge notablemente de la calificación que otorgaría un profesor; sumando esto al hecho de que métodos diversos de evaluación pueden dar resultados distintos, si se utiliza autoevaluación sumativa parece necesario combinarla con otros métodos de calificación variados.

Se han descrito diversas variantes de aplicación: autoevaluación individual / cruzada / en grupo, tests, ejercicios con solución, herramientas... La autoevaluación, y formas concretas de la misma, está especialmente indicada en algunos ámbitos propicios, como las presentaciones orales, el entorno de equipos de trabajo o el control en el desarrollo de proyectos.

Respecto al coste de implantación, es un factor a tener en cuenta, porque aumenta el esfuerzo de los implicados y especialmente del profesor; sin embargo existen formas de autoevaluación ligera que arrojan resultados de

interés. De hecho, se ha propuesto y utilizado la autoevaluación no como medio didáctico en sí mismo, sino como medio accesorio para contrarrestar el coste de adopción de otras técnicas no relacionadas. Por otra parte, se ha documentado la postura claramente positiva de los alumnos ante los métodos de autoevaluación.

Existen diversas herramientas que pueden ayudar a reducir el coste de adopción, pero hay que recordar que conllevan su propio coste de implantación y aprendizaje, y en ocasiones son específicas de un dominio o materia concretos.

Referencias

- [1] Abelló, A. et al. *LEARN-SQL: Herramienta de gestión de ejercicios de SQL con autocorrección*. Actas de JENUUI 2009.
- [2] Antelm, J. M. et al. *SAM: Sistema de Autoevaluación Multimedia*. Actas de JENUUI 2003.
- [3] Aznar Gregori, F. et al. *Una experiencia metodológica con la autoevaluación como telón de fondo*. Actas de JENUUI 2006.
- [4] Badía, J.M. et al. *Y los estudiantes, ¿qué opinan?* Actas de JENUUI 2008.
- [5] Burgués, X. et al. *Uso de LEARN-SQL en el aprendizaje cooperativo de Bases de Datos*. Actas de JENUUI 2008.
- [6] Canivell Castillo, V.; Jacob Taquet, I.; Oliver Bernal, J. *Valoración y opiniones de los alumnos sobre las técnicas docentes aplicadas en el aprendizaje de bases de datos*. Actas de JENUUI 2008.
- [7] Castel de Haro, M. J. et al. *e-VALUACIÓN en tiempo real*. Actas de JENUUI 2009.
- [8] Cataldi, Z.; Salgueiro, F.; Lage, F. J. *Predicción del rendimiento de los estudiantes y diagnóstico usando redes neuronales*. Actas de JENUUI 2007.
- [9] CRUE. Preguntas frecuentes sobre el EEES / metodología docente. http://www.crue.org/espacioeuropeo/FAQs/metodologia_docente.html
- [10] Díaz, M. A. et al. *E-valúate: un modelo de autoevaluación para la mejora de la enseñanza y el aprendizaje*. Actas de JENUUI 2003.
- [11] Escribano Otero, J. J.; García García, M. J. *Aventuras y desventuras en métodos docentes. Experiencias aprendidas*. Actas de JENUUI 2008.
- [12] Gayo Avello, D.; Fernández Cuervo, H.; Torre Cervigón, F. *La autoevaluación como método de aprendizaje*. Actas de JENUUI 2002.
- [13] Gómez Albarrán, M. *Una revisión de métodos pedagógicos innovadores para la enseñanza de la programación*. Actas de JENUUI 2003.
- [14] González Morales, D.; Roda García, J. L.; Moreno de Antonio, L. M. *Aplicando diferentes técnicas de evaluación*. Actas de JENUUI 2008.
- [15] Luján-Mora, S.; Mingot Latorre, I. *Sistema para la (auto)evaluación de los alumnos a través de la web*. Actas de JENUUI 2003.
- [16] Mosqueira-Rey, M. *La evaluación continua y la autoevaluación en el marco de la enseñanza de la programación orientada a objetos*. Actas de JENUUI 2010.
- [17] Oliver, J.; Canivell, V. *Evaluación entre compañeros: estudio de su correlación con la evaluación del profesor*. Actas de JENUUI 2009.
- [18] Pérez de la Cruz, J.L.; Millán, E. *Estudio comparativo de diversos métodos de evaluación*. Actas de JENUUI 2009.
- [19] Pérez Pérez, J. R. et al. *Entorno Web de desarrollo para el aprendizaje de paradigmas de programación*. Actas de JENUUI 2003.
- [20] Real Academia Española. *Diccionario de la Lengua Española*. 23ª edición (avance).
- [21] Riesco, M.; Díaz, M. *Sistema docente de re-alimentación inmediata en clases prácticas*. Actas de JENUUI 2005.
- [22] Riesco Albizu, M.; Díaz Fondón, M. *La revisión entre iguales como herramienta de aprendizaje y evaluación en la asignatura de sistemas operativos*. Actas de JENUUI 2007.
- [23] Rodríguez del Pino, J. C. et al. *Hacia la evaluación continua automática de prácticas de programación*. Actas de JENUUI 2007.
- [24] Sáiz Díez, J. M.; López Zozal, C.; Marticorena Sánchez, R. *Utilización de herramientas para el autoaprendizaje y autoevaluación en redes de ordenadores*. Actas de JENUUI 2009.
- [25] Valero García, M.; Díaz de Cerio, L. M. *Evaluación continuada a un coste razonable*. Actas de JENUUI 2003.

[26] Velasco, L.; Villavieja, C. *Cómo evaluar continua e individualmente en asignaturas*

basadas en proyectos. Actas de JENUI 2009.

La sobre-evaluación

Jesús Serrano-Guerrero, Francisco P. Romero,
Emilio Fdez-Viñas, José A. Olivas

Departamento de Tecnologías y Sistemas de Información
Universidad de Castilla-La Mancha
Paseo de la Universidad 4
13071 Ciudad Real

{jesus.serrano, franciscop.romero, emilio.fernandez,
joseangel.olivas}@uclm.es

Resumen

Este trabajo resume una experiencia negativa vivida durante la impartición de una asignatura encuadrada dentro del Espacio Europeo de Educación Superior (EEES) debido a una mala planificación de la evaluación que desemboca en lo que denominaremos *sobre-evaluación*. Por sobre-evaluación se entiende el excesivo número de pruebas a las que se somete al alumno y que le obligan a pasar más tiempo preparándolas que adquiriendo o asentando conocimientos. Y como consecuencia de esa experiencia negativa se presentan los medios que se han puesto para evitar repetir dicha experiencia en el curso siguiente en la misma asignatura así como las algunas de las conclusiones alcanzadas por los profesores de dicha asignatura.

Summary

This work summarizes a negative experience happened during the classes of a subject within the European Space for Higher Education, due to a wrong planning which provoked the so-called phenomenon: *over-assessment*. *Over-assessment* means the excessive number of tasks that a student must do. As a result of this fact, each student spends more time preparing his tasks than learning new knowledge. From this negative experiment, the following course the same teachers applied several modifications with respect to that subject in order to avoid past errors. These modifications allowed improving the results obtained by the students, especially, due to the better planning of the tasks proposed to assess students.

Palabras clave

Sobre-evaluación, fundamentos de programación, planificación temporal, bloques temáticos, trabajo incremental

1. Introducción

Dentro del marco de las pruebas piloto para la adaptación de asignaturas y planes de estudio a los requerimientos del Espacio Europeo de Educación Superior (EEES) llevadas a cabo en la Escuela Superior de Informática de Ciudad Real, son muchas las experiencias adquiridas y las lecciones aprendidas. Dentro de las lecciones aprendidas este trabajo pretende resaltar una de ellas, el problema de *la sobre-evaluación* entendida como la evaluación excesiva de la actividad del alumnado. Las experiencias aquí descritas se basan en la asignatura de Fundamentos de Programación impartida en la carrera de Ingeniería en Informática, que ahora ha pasado a llamarse Grado en Informática. Esta misma asignatura se denominaba Metodología y Tecnología de la Programación hasta el curso 2009/10.

Son muchos los métodos que se pueden utilizar para evaluar a un alumno [1, 2, 3, 4] pero el problema surge cuando hay que elegir cuál es el más adecuado para evaluar cada actividad. Cada uno de ellos puede aportar distintas cosas al alumno pero dependiendo de la actividad que vaya a ser evaluada unos son más adecuados que otros. El problema surge cuando todos ellos son utilizados para evaluar todos los conocimientos que debe adquirir un alumno. En este caso el número de pruebas que se realizan es excesivo; este fenómeno será mencionado a lo largo de este artículo como *sobre-evaluación*, es decir, la

realización de una serie de pruebas de forma desproporcionada con el fin de evaluar a los alumnos. Este hecho genera una serie de problemas, tanto para el alumno como para el profesor, que deberían ser evitados, pero que a veces surgen por una mala interpretación del concepto de evaluación continua.

El presente artículo comenta algunos errores cometidos por algunos profesores en el intento por implantar la metodología ECTS en la Escuela Superior de Informática de C. Real durante el curso 2008/09. Estos errores provocaron una excesiva evaluación de los alumnos que desembocó en una serie de efectos poco deseables y conflictos entre alumnos y profesores. Dichos hechos son comentados a continuación, así como algunos de los mecanismos puestos para remediar, durante el curso 2009/10, los errores anteriores y los resultados obtenidos con ellos.

El presente trabajo está organizado en 7 secciones. Después de esta primera sección introductoria, la segunda sección muestra el contexto en el cual se ha trabajado. La tercera sección muestra una idea acerca del concepto de *sobre-evaluación* mientras la sección 4 narra las principales consecuencias que tuvo para los profesores como para los alumnos dicha *sobre-evaluación* durante la impartición de la asignatura Metodología y Tecnología de la Programación en la Escuela Superior de Informática de C. Real durante el curso 2008/09. La sección 5 presenta algunas pautas seguidas durante el curso siguiente en la misma asignatura, con el fin de evitar parte de los efectos negativos de la *sobre-evaluación*, mientras que en la sección 6 se comentan las mejoras logradas al aplicar las pautas propuestas en la sección 5 con respecto al curso anterior. Finalmente algunas conclusiones son comentadas en la última sección.

2. Contexto de la asignatura

Como se comentó anteriormente, el trabajo versa sobre las experiencias de algunos de los profesores de la asignatura Metodología y Tecnología de la Programación (MYTP) impartida en el primer curso de Ingeniería Informática y que consta de 13 créditos ECTS. Esta asignatura formó parte de la prueba piloto para la adaptación de asignaturas y planes de estudio a los requerimientos del Espacio Europeo de Educación

Superior (EEES) de la Escuela Superior de Informática de C. Real durante el curso 2008/09. A partir de entonces la asignatura se mantiene conforme a los requerimientos del Espacio Europeo de Educación Superior (EEES).

Durante el curso 2008/09, en la evaluación de esta asignatura se contemplaron como métodos de evaluación: tests y ejercicios a realizar en clase sin material bibliográfico ni apuntes en horario lectivo, ejercicios a realizar en casa en papel con todo el material disponible, y una serie de prácticas a implementar en el ordenador en casa. Además dentro de la evaluación destaca la obligación de asistir al menos al 80% de las clases y presentar al menos el 80% de trabajos propuestos obligando a alcanzar una nota mínima en al menos el 80% de las pruebas presentadas con el fin de que los alumnos no presenten por presentar sus trabajos.

A los alumnos se les proporcionó una serie detallada de pruebas que se les va a realizar antes de comenzar el curso. Todos los temas tenían varias pruebas, bien tests, prácticas, ejercicios para casa, etc. No se contempló la posibilidad de realizar una prueba final sobre todo el temario de cada asignatura salvo para el caso de los alumnos que no llegasen a una nota mínima a lo largo del curso, los cuales tenían la oportunidad de aprobar la asignatura mediante un examen final en la convocatoria extraordinaria de Julio que valía el 50% de la nota final, y el otro 50% restante de la nota final se correspondía con todas las actividades llevadas a cabo durante el curso. Así pues, un alumno podía aprobar con un cinco la asignatura incluso sin necesidad de haber asistido nunca a clase, aunque este hecho era bastante improbable. El cálculo de la nota final se hizo en base a todas las actividades realizadas a lo largo del curso mediante una fórmula matemática que ponderaba cada actividad y que se proporcionó a principio de curso.

Como puede observarse, el número de pruebas a las que se le sometió a cada alumno, es número excesivo en el afán de los profesores por controlar todos los conocimientos de sus alumnos.

Sin embargo, a pesar de que durante el curso 2009/10 la metodología de evaluación fue similar, es decir, se siguieron realizando tests, ejercicios en clase y clase y prácticas; se introdujeron algunas modificaciones en la temporalidad entre las pruebas y algunas restricciones de las

asignatura que se verán en la sección 5 con el fin de evitar algunos de los errores cometidos y que dieron lugar a resultados mucho más satisfactorios.

3. Sobre-evaluación

La metodología ETCS es comentada por muchos profesores como una marcha atrás, el regreso al instituto, la vuelta a las recuperaciones, los exámenes tema a tema porque los alumnos no dan más de sí, los padres de nuevo preguntándole al profesor como van sus hijos, y así una serie de comentarios algo alejados de lo que realmente propone dicha metodología. Muchos profesores realmente sienten y se contagian de este espíritu negativo en torno a la nueva adaptación al Espacio Europeo de Educación Superior (EEES). Este hecho hace que realmente muchos profesores sientan que tienen que tratar a los alumnos como niños de instituto y llevar una supervisión de todo lo que hacen día a día para que no se desvíen por el mal camino.

Hay docentes que realmente sienten que para adaptarse a los nuevos tiempos es necesario tener el control sobre todas las actividades del alumno, conocer si entiende todos los conceptos explicados en clase y si no los entiende ponerle más tareas para que finalmente los entienda. Esta concepción de la metodología ETCS hace que el profesor sienta que tiene que supervisar todos los movimientos del alumno y para ello la única herramienta de la que disponen es la evaluación. La evaluación sirve de registro para que queden patentes todas los conocimientos del alumno y contra mayor sea este control sobre el alumno más seguro se sentirá el docente de ser justo a la hora de juzgar si realmente el alumno adquirió o no los conocimientos esperados.

Esta concepción errónea de la evaluación desemboca en un fenómeno que hemos denominado como *sobre-evaluación*, la excesiva realización de pruebas con el fin de controlar los conocimientos del alumno. Este fenómeno surge a raíz de la obsesión por obligar al alumno a trabajar constantemente con el fin de que éste no desvíe su atención de una asignatura concreta y que a su vez quede registro de todas sus actividades para poder juzgarlo.

Un ejemplo de sobre-evaluación es la rápida evaluación en bloques temáticos pequeños. Es

fácil pensar en la evaluación por bloques temáticos con la intención de, nada más terminar de explicar una serie de conceptos, obligar al alumno a que los domine de forma rápida para poder volcarse el profesor con otros nuevos conceptos. Sin embargo, esta práctica puede conllevar el riesgo de una evaluación demasiado rápida si el bloque temático es demasiado pequeño. Este riesgo puede darse en aquellas partes del temario donde haya temas cuyo contenido es poco solapado, por ejemplo, en la asignatura Metodología y Tecnología de la Programación tenemos Tema 3 “Matrices”, Tema 4 “Principios de recursividad”, Tema 5 “Ordenación y búsqueda” y Tema 6 “Objetos” donde una evaluación rápida puede provocar que los alumnos no hayan asentado sus conocimientos de forma razonable.

El mismo fenómeno puede darse en los bloques temáticos solapados en contenido. El profesor considera que para seguir con un bloque temático es necesario conocer perfectamente el anterior por lo que comienza a hacer pruebas al alumno con el fin de obligarlo a que domine unos determinados conceptos. Esta práctica puede ser contraproducente, ya que el alumno puede ser que no tenga suficiente tiempo para adquirir conceptos que son complicados y requieren de otros conceptos posteriores para poder complementarlos y conocerlos en toda su amplitud.

4. Consecuencias

Como consecuencia de nuestra experiencia con la sobre-evaluación se comentan a continuación una serie de aspectos negativos detectados.

4.1. Consecuencias para los alumnos

Los alumnos, a lo largo del curso 2008/09, fueron presentando una serie de efectos que podrían evitarse con una mejor planificación de la evaluación:

- Ansiedad.
Los alumnos asumen cada nueva prueba como un nuevo reto sobre el que volcar sus esfuerzos dado que muchos de ellos no saben cuantificar el valor real de cada prueba a realizar.

- **Mala planificación.**
Los alumnos, especialmente en primer curso, no saben gestionar el tiempo del que disponen y por ese motivo vuelcan todo su esfuerzo en cada prueba, olvidando los trabajos del resto de asignaturas. Los alumnos de último año de carrera tienen más experiencia y al menos ya saben que asignaturas van a prepararse cada cuatrimestre, su planificación es más coherente pero en primer curso puede llegar a tener efectos muy negativos.
- **Fracaso.**
La falta de tiempo para poder asimilar conceptos, la ansiedad y otros factores impiden al alumno rendir plenamente y provocan en muchas ocasiones que los resultados sean bastante negativos.
- **Olvidan más rápido que aprenden.**
Los objetivos a corto plazo hacen que el alumno crea que es capaz de mantener opciones de aprobar la asignatura por lo que realiza un gran esfuerzo para sacar una buena nota en una actividad o incluso recurre a agentes externos para que le ayuden a hacerlo como compañeros que les hacen los ejercicios o les ayudan en el examen.
Estos hechos dan la sensación de conocimiento pero que pronto desaparecerá bien porque el alumno no tenía dicho conocimiento o porque fue adquirido de forma rápida y posteriormente fue olvidado a la misma velocidad que se aprendió.
- **Falsa concepción de la asignatura.**
Los alumnos que no son capaces de mantener el ritmo de la asignatura tienden a abandonar la asignatura transmitiendo a sus compañeros la idea de alta dificultad de la misma aunque esto no sea así. De igual manera, tal y como se comentó anteriormente, el alumno tiende a intentar aprobar cada prueba realizada y olvidar posteriormente por lo que nunca alcanza una idea global de la asignatura que cursa.
- **Pérdida de tiempo en tareas inútiles.**
Restricciones como la entrega de al menos el 80% de las tareas propuestas por el profesor provocan que el alumno crea que simplemente por entregar un trabajo o presentarse a realizar un test aunque la nota que saque sea bajísima puede aprobar una asignatura. Este hecho es negativo, aparte de por la pérdida de tiempo en preparar un trabajo del que no va a aprender apenas nada, porque la sensación del alumno es que puede todavía aprobar la asignatura cuando realmente la sensación que debería tener sería la de que no tiene los conocimientos necesarios para proseguir con el resto de la asignatura.
Esta última sensación sería provocada por la nota final de la actividad realizada pero se distorsiona con la idea de que ha entregado el trabajo y puede seguir presentando los distintos trabajos, prácticas, exámenes, etc., que restan hasta final de curso.
Otra restricción que provoca la pérdida de tiempo en tareas cuyo aprovechamiento es bajo, es la asistencia obligatoria, que hace que muchos alumnos pierdan parte de su tiempo en asistir a clases de asignaturas que finalmente abandonarán, asisten por inercia más que por interés. Este hecho además provoca que haya alumnos en el aula molestando a sus compañeros y al profesor. Este fenómeno es mucho más significativo entre los alumnos de primer curso. Los alumnos de últimos cursos cuando abandonan una asignatura no vuelven a asistir a clase porque son conscientes de que la asistencia es requisito para aprobar la asignatura.
- **Abandono de los aprobados.**
Aquellos alumnos que ya saben, haciendo uso de la fórmula matemática dada a principio de curso, que van a aprobar la asignatura según todas las calificaciones obtenidas hasta un determinado momento del curso no siguen realizando las últimas actividades del curso para centrarse en otras asignaturas. Este hecho se da especialmente en el caso de los alumnos repetidores que sacan buenas notas durante los primeros temas y luego abandonan a final de curso.
- **Competencias globales no alcanzadas.**
Es fácil pensar que mediante una evaluación constante el profesor puede tener constancia de todos los conocimientos de sus alumnos, lo cual hasta cierto punto es cierto pero puede ser contraproducente. Controlar todas las actividades de los alumnos coarta sus libertades para pensar e impide la adquisición de competencias horizontales [5, 6, 7] como la capacidad de análisis, síntesis y selección de la información adecuada dentro de grandes

cantidades de información, dado que la mayoría de las pruebas versan sobre un único tema de teoría.

4.2. Consecuencias para los profesores

Por el contrario los profesores presentaron los siguientes síntomas por culpa de la *sobre-evaluación*:

- Sobrecarga de trabajo para el profesor. Excesiva dedicación para la preparación y corrección de exámenes, así como para la planificación y replanificación de las fechas para cada una de las actividades programadas en la guía docente.
- Conflictos con otros profesores. Los alumnos solicitan a los otros profesores que aplacen sus pruebas de evaluación para fechas posteriores porque tiene varias pruebas esta asignatura. Esto genera malestar sobre entre los profesores más benévolos y sobre los alumnos con los profesores menos benévolos porque estos no ceden ante sus pretensiones.
- Retrasos en la planificación docente. La excesiva dedicación a la realización de pruebas, impide la realización de otras actividades como tutorías, clases magistrales, corrección de ejercicios, etc.; y como consecuencia un atraso en la planificación del temario y una mayor precipitación posterior para poder terminar el temario.
- Frustración inicial. Los profesores de primer curso observan que la mayoría de los alumnos a pesar de hacer cada vez los exámenes o pruebas con un menor contenido siguen suspendiendo, lo cual genera una sensación de insatisfacción y frustración por el trabajo realizado que les hace plantearse incluso rebajar las exigencias de cada una de las pruebas a las que someten a sus alumnos.
- Alumnos molestos y desinteresados. La asistencia obligatoria a clase puede ser otro ejemplo de sobre-evaluación, un control excesivo de los alumnos, genera que estos vayan a clase con el único fin de poder seguir siendo evaluados, sin motivación alguna. Esto provoca como se comentó anteriormente que el profesor sufra constantes perturbaciones por parte de este grupo de alumnos.

En líneas generales podría comentarse que los alumnos de primer curso intentan abordar todas las asignaturas con la misma intensidad, especialmente las del primer cuatrimestre, sin embargo durante el segundo cuatrimestre se observa que los alumnos ya han asumido la mayoría de sus fracasos y deciden centrarse en unas pocas asignaturas, salvo honrosas excepciones que mantienen esperanzas de aprobar todas o al menos la mayoría de la asignaturas de las que se matricularon.

La evaluación excesiva tiene como característica positiva que el profesor conoce totalmente la actividad de todos sus alumnos siempre y cuando las clases contengan un número reducido de alumnos tal y como propone el EEES. Algunos alumnos, especialmente repetidores, la ven como positiva porque no son capaces por sí mismos de obligarse a estudiar de forma ordenada a lo largo del curso y la ven como una imposición para su beneficio.

Por el contrario, si trasladamos esta misma idea a alumnos de último curso de carrera, una gran mayoría comenta sentirse insultados porque hacer pruebas con tan poco tiempo es propio de estudiantes de educación secundaria y que da la sensación de que nunca serán capaces de asumir responsabilidades de gran envergadura.

5. Experiencia durante el curso 2009/10

Como consecuencia de todas las experiencias narradas anteriormente, durante el curso siguiente, 2009/10, se plantearon algunas modificaciones con el fin de corregir los errores cometidos:

- Definición coherente de los bloques temáticos sobre los que se realizan las evaluaciones. Se ha reorganizaron los temas de la asignatura en 3 bloques temáticos con el fin de agruparlos para abarcar varias competencias tanto horizontales como verticales. Estos bloques principalmente incluían los elementos básicos de programación como variables, sentencias de selección y bucles por un lado, elementos referentes a estructuras de datos como matrices, registros y ficheros por otro lado, y finalmente los elementos referentes a orientación a objetos y herencia. Era un error intentar evaluar un grupo demasiado pequeño de competencias. Dado

que los bloques eran demasiado pequeños, los alumnos tendían a aprender los conceptos de forma rápida y a olvidarlos de forma más rápida.

- Mayor espacio de tiempo entre la finalización de un bloque temático y las pruebas correspondientes.

Establecer un tiempo razonable entre la finalización de cada bloque temático y la realización de la evaluación mediante tests o ejercicios sin material bibliográfico que les permita a los alumnos asentar sus conocimientos, tiene especial interés en las fechas de finalización del curso académico cuando el temario termina de forma precipitada y los últimos exámenes y trabajos se mandan sin darle apenas tiempo al alumno. En este caso, se dieron entre dos semanas desde la finalización de cada bloque temático para la realización de tests o ejercicios en clase sin material bibliográfico; mientras que las prácticas y ejercicios en casa se fueron intercalando a medida que se iban impartiendo las distintas unidades docentes de cada bloque temático.

La evaluación del último bloque temático se realizó, una vez finalizado el periodo de clase, con el fin de poder darles algo más de margen a los alumnos aunque las notas se publicaran un poco más tarde.

- Flexibilización del concepto de asistencia obligatoria clase ligándolo a la idea de participación en el aula porque la asistencia sin más no tiene excesivo sentido.

Se propuso la realización tanto de ejercicios obligatorios como optativos en las clases con el fin de mantener la atención de los alumnos. La realización de estos ejercicios podía suponer un 5% de la nota final para los alumnos que los entregasen y podía suponer el aprobado o el suspenso para aquellos alumnos que estuvieran cerca de aprobar a final de curso.

- Flexibilización el concepto de entrega de un número mínimo de trabajos ligándolo con el concepto de calidad de los mismos porque la entrega de trabajos simplemente para cumplir unos mínimos no aporta nada al alumno.

Se exigió que cada actividad entregada para su corrección, tuviera al menos un nota de 3.5, en caso de que no fuera así, el profesor podría

considerar mandar al alumno al examen planificado para la convocatoria ordinaria con el fin de que realizara en dicho examen algún ejercicio relacionado con la prueba suspensa.

- Realización de pruebas con carácter global, es decir, que engloben varios apartados del temario con el objetivo de tener una percepción global del conocimiento final del usuario. Las pruebas pequeñas pueden favorecer muchas competencias verticales pero no es así con el caso de las competencias horizontales.

Dada la asignatura que se trabaja, Metodología y Tecnología de la Programación, es imposible aprender en el primer tema las variables, y en el octavo tema “Herencia y Polimorfismo” haberlas olvidado. Por lo que para obligar a dominar todos los conocimientos anteriores en cada momento se propuso un tipo de prácticas incrementales en requisitos. Estas prácticas consistieron en la realización de un juego de tipo “sudoku” en el que en la primera entrega incorporaba una serie pequeña de requisitos implementables mediante variables y estructuras de control, posteriormente, en las siguientes entregas se iban añadiendo nuevas funcionalidades que se podían implementar mediante sentencias iterativas o recursivas aparte de corregir los fallos detectados en la primera entrega, para finalmente adaptarse esta solución inicial a una solución orientada a objetos donde los objetos, la herencia y los ficheros jugaban el papel principal.

- No entender el concepto de prueba como una herramienta para obligar al alumno a trabajar. Los alumnos deben trabajar a diario para adquirir una serie de conocimientos no porque el docente plantee la realización de una prueba. Los alumnos no tienen por qué aprender más por el hecho de que haya 20 pruebas en lugar de 5.

Por esa razón se redujeron a más de la mitad las pruebas que se realizaron durante el curso 2009/10 con respecto al curso anterior.

6. Comparativa entre ambos cursos

Comparando los resultados obtenidos durante el curso 2008/09 y los resultados en el curso

2009/10, podemos observar significativas mejoras.

Durante el curso 2009/10, casi un 30% de los alumnos matriculados aprobaron la asignatura sin necesidad de ir al examen final. Y una vez realizado el examen final comentado en la sección 2, el número de aprobados fue de casi un 40%, lo cual fue un dato considerablemente bueno con respecto a los años anteriores, en los que sólo había un examen final y únicamente aprobaron en torno al 10% de los alumnos matriculados.

Sin embargo, durante el curso 2009/10 el número de alumnos aprobados durante la evaluación ETCS, sin necesidad de ir al examen final, fue casi de un 50% de los alumnos matriculados. Pero también hay que destacar que el número de alumnos que aprobaron gracias al examen final llegó al 5%, es decir, sólo aquellos alumnos que se encontraban casi en condiciones de aprobar durante el transcurso del curso aprobaron, el resto de alumnos suspendió, además con unas notas considerablemente bajas, la mayoría ni siquiera alcanzaba un 2 en el examen final.

Por otro lado, las notas alcanzadas por los alumnos del curso 2008/09 fueron considerablemente más bajas con respecto a las notas del año siguiente. En el primero se obtuvo de nota media casi un 6 entre los alumnos aprobados mientras que en el segundo curso la nota alcanzada supera ligeramente el 7.

Así pues, este hecho manifiesta una considerable mejoría con respecto al año anterior en cuanto a términos estadísticos se refiere aunque tampoco conviene generalizar, dado que sólo se tiene datos de dos años y estamos en los comienzos de la implementación del modelo de Bolonia.

Observando las actitudes de los alumnos en ambos cursos, claramente se observa una menor ansiedad durante el segundo curso y una mejor planificación, aunque los alumnos que se lo dejan todo para el último día siguieron existiendo y fueron en su mayoría los que suspendieron.

Una de las razones principales por la que en el curso 2009/2010 se mejoró la nota media fue la necesidad de sacar una nota mínima en cada prueba, lo que evitó que mucho alumnos, especialmente repetidores, abandonaran la asignatura cuando ya sabían que su nota media era superior a 5.

La existencia de una pequeña puntuación, 5%, para aquellos alumnos que salían a la pizarra o el hecho de que el profesor los obligase a salir, hizo que muchos alumnos, especialmente a finales de curso cuando se decía la nota final y algunos veían que podían no llegar a un 5 en su nota final, se tomaran mucho más en serio las clases e incluso se pelearan por salir a la pizarra. Los buenos estudiantes mantuvieron, en líneas generales, esta actitud de participación en clase durante el curso.

Otro de los grandes aciertos que consideramos los profesores que llevaron al éxito alcanzado en el curso 2009/10, fue el concepto de prueba global e incremental, es decir, el juego “sudoku” que se realizó en casa y que obligó en todo momento a los alumnos a mantener, incrementar y aplicar todos los conocimientos adquiridos a lo largo del curso. Este hecho da sobre todo al docente, la sensación de dominio continuo y mejorado de los conocimientos anteriores por parte de alumno, y hace que lo vea con otros ojos cuando puede palpar su evolución positiva.

Desde el punto de vista de los profesores, ni comentar cabe que el número de pruebas disminuye el trabajo a realizar y permite posiblemente diseñar mejor las pruebas evaluables. Aparte la sensación de frustración desaparece con respecto al año anterior, porque como demuestran los datos, el número de alumnos que aprueba es mayor y la calidad de los trabajos también mejora. Este hecho también provoca que el profesor se haga más exigente con los alumnos al ver que ellos responden de forma positiva, lo cual a veces hay que controlarlo porque podemos volvernos más exigentes de lo que deberíamos serlo.

7. Conclusiones

La experiencia aquí narrada presenta una serie de situaciones negativas que pueden darse a la hora de planificar la evaluación de las actividades que conforman una asignatura y cómo pueden solucionarse, al menos parcialmente. Así pues puede ser considerado como un documento a tener en cuenta antes de comenzar la planificación de la evaluación de cualquier asignatura, independientemente del curso en el cual se imparta.

La evaluación debe ser considerada como un mecanismo para conocer el estado de los

conocimientos de los alumnos pero no es una herramienta para obligar a los alumnos a estudiar, es necesario buscar otros elementos para motivar al alumno, un gran número de pruebas casi siempre tiene efectos negativos sobre el rendimiento del alumnado.

Bajo la experiencia de los autores del presente artículo, la sobre-evaluación pone de manifiesto que todavía hay muchos conceptos que quedan por ser totalmente asimilados por muchos profesores hasta conseguir una evaluación equilibrada de sus asignaturas. Es necesario un cambio en la concepción de muchos conceptos de la metodología ECTS por parte de muchos docentes.

Agradecimientos

Este proyecto está financiado gracias al proyecto FIDELIO (TIN2010-20395, MEC-FEDER), proyecto de excelencia SCAIWEB-2 (PEIC09-0196-3018, JCCM), proyecto PLINIO (POII10-0133-3516, JCCM), proyecto de Innovación Docente HIPERION (013702011, UCLM) y el proyecto HECAOT (TC20112330, UCLM).

Referencias

- [1] González Morales, D.; Roda García, J.L. y Marina Moreno de Antonio, L. Aplicando diferentes técnicas de evaluación. En Actas

de las XIV Jornadas de Enseñanza Universitaria de Informática, Jenui 2008, pp. 403–410, Granada, Julio 2008.

- [2] Guitart, I; Rodríguez, M.E; Jordi Cabot y Serra, M. Elección del modelo de evaluación: caso práctico para asignaturas de ingeniería del software. En Actas de las XII Jornadas de Enseñanza Universitaria de Informática, Jenui 2006, pp. 191– 198, Bilbao, Julio 2006.
- [3] Millán Valdeperas, E. y Pérez de la Cruz, J. J. Estudio comparativo de diversos métodos de evaluación. En Actas de las XV Jornadas de Enseñanza Universitaria de Informática, Jenui 2009, pp. 225–231, Barcelona, Julio 2009.
- [4] Pérez, M. T. Estrategias de evaluación y autoevaluación en el marco del EEES. cimanet.uoc.edu/mel/content/view/46/82/lang,spanish/
- [5] Sánchez, F. et al. Competencias profesionales del Grado en Ingeniería Informática. Actas de las XIV Jornadas de Enseñanza Universitaria de Informática, Jenui 2008, Julio, Granada.
- [6] Villa, A y Poblete, M. Aprendizaje basado en competencias. Editorial Mensajero. Universidad de Deusto. 2007
- [7] Voorthess, R. Measuring what matters: competency-based learning models. Higher Education. Jossey Bass. 2001

Sesión 2B:
Métodos pedagógicos innovadores I

Utilización de una herramienta de comunicación online para la mejora docente. Dos casos prácticos.

Francisco Grimaldo-Moreno, Miguel Arevalillo-Hérraez, Emilia López-Iñesta

Departament d'Informàtica

Universitat de València

Avda. Vicent Andrés Estellés, s/n

46100 Burjassot

{francisco.grimaldo,miguel.arevalillo}@uv.es, eloi@alumni.uv.es

Resumen

En este artículo, describimos el uso de una herramienta de comunicación online para dos propósitos diferentes en dos contextos distintos. En particular, la herramienta Elluminate Live! se ha utilizado para a) complementar las tutorías presenciales en una asignatura de primer curso de programación, mediante una serie de tutorías virtuales que se realizan a través de Internet y b) como medio de apoyo a la docencia presencial en una asignatura de máster oficial universitario, para permitir la participación de profesores desde el extranjero. En ambos casos, la experiencia ha resultado satisfactoria.

Summary

In this article we describe the use of an online communication tool in two different contexts. In particular, Elluminate Live! has been used to a) provide virtual tutorial sessions as a complement to individualized tutorials and b) allow lecturers living abroad to participate in a Masters module. In both cases, we have obtained satisfactory results.

Palabras clave

e-learning, blended learning, herramientas de comunicación síncronas, Elluminate Live!, enseñanza online.

1. Introducción

La adaptación al Espacio Europeo de Educación Superior (EEES) implica cambios significativos en la educación universitaria, especialmente en lo

referido a las metodologías docentes. Tal es la importancia del cambio que la LOMLOU [1] contempla entre sus acciones impulsar la realización de programas dirigidos a la renovación metodológica de la enseñanza universitaria para el cumplimiento de los objetivos de calidad del EEES (Art. 89-5). Además, la mayoría de las universidades han creado unidades administrativas para la gestión de la convergencia Europea, o han implementado experiencias piloto enfocadas a este proceso.

Por otro lado, uno de los principales acuerdos contemplados en la declaración de Bolonia consiste en promover la dimensión europea de la educación superior y en particular, el desarrollo curricular, la cooperación institucional, esquemas de movilidad y programas integrados de estudios, de formación y de investigación [3].

Los avances tecnológicos en materia de enseñanza, han dado lugar a herramientas de comunicación específicamente diseñadas para la docencia. En este artículo presentamos dos casos en los que una de estas herramientas de comunicación, Elluminate Live! [5], ha sido utilizada para dos propósitos distintos. En el primer caso, se utiliza para facilitar la implantación de metodologías docentes basadas en el aprendizaje activo y la realización de actividades prácticas, en una asignatura de programación de primer curso. En el segundo caso, la herramienta se utiliza para permitir que profesorado localizado en el extranjero pueda participar en la docencia sin que ocurran desplazamientos, en una asignatura de Máster sobre sistemas inteligentes distribuidos.

Con estos dos casos, pretendemos resaltar la importancia de que las universidades centren parte

de sus esfuerzos en dotar a alumnos y profesorado de una infraestructura tecnológica de apoyo a la docencia que proporcione herramientas de comunicación síncronas. Presentamos para ello dos de entre muchas utilidades prácticas de este tipo de aplicaciones en el ámbito docente. Ambas están orientadas al cumplimiento de los objetivos del EEES, donde destaca la relevancia de la tutoría virtual [9], así como el rol y el papel del tutor [8].

Otros trabajos han evaluado los beneficios de la utilización de herramientas de comunicación síncronas y asíncronas en el contexto docente [4][7]. Sin embargo, este artículo no tiene como objetivo ofrecer una comparativa entre las diferentes plataformas que proporcionan estas funcionalidades de comunicación, sino exponer algunas de las posibles utilidades que las herramientas síncronas pueden tener en docencia, resaltando las ventajas e inconvenientes en el contexto de dos experiencias prácticas.

La elección de Elluminate Live! como herramienta de comunicación online se debe principalmente a decisiones estratégicas de la propia universidad. En 2007, se puso en marcha un proyecto piloto de un año de duración cuyo objetivo principal era evaluar la utilidad docente de la herramienta. Dicho proyecto concluyó con una evaluación positiva, que derivó en la adquisición de 30 licencias de uso simultáneo de la plataforma. Estas licencias estaban completamente financiadas por la universidad, y tenían como objetivo proporcionar el servicio bajo demanda de forma gratuita al profesorado que lo solicitase [10].

El resto del artículo se estructura de la siguiente forma. En la sección 2, describimos Elluminate Live!. En las secciones 3 y 4 describimos los dos casos prácticos que se presentan. Para cada uno de ellos, describimos la asignatura, el problema que intentamos solucionar, el enfoque adoptado y los resultados. Finalmente, la sección 5 expone las conclusiones del artículo.

2. La herramienta

Elluminate live! es un entorno virtual que integra un conjunto de utilidades especialmente

construidas para la docencia, entre las que se incluyen las siguientes:

- Herramientas de comunicación. Elluminate live! proporciona comunicación bidireccional mediante VoIP, con uso opcional de video. Además, incorpora un chat que puede utilizarse cuando el usuario no dispone de la infraestructura necesaria para utilizar voz.
- Transferencia de archivos. Mediante una utilidad interna, permite el envío y recepción de archivos de forma simple e integrada.
- Pizarra interactiva. La plataforma dispone de una pizarra compartida donde los usuarios pueden dibujar y escribir. Esta pizarra puede utilizarse para apoyar las explicaciones que se realizan, utilizando una tableta digitalizadora.
- Compartición de aplicaciones. La herramienta permite compartir una aplicación entre varios usuarios. El usuario puede otorgar acceso a una aplicación en concreto a otro usuario o al profesor, para que pueda ayudarle en cierta actividad o simplemente visualizar cómo está realizando la tarea.
- Visualización de presentaciones. Elluminate Live! permite visualizar presentaciones y acompañarlas de explicaciones. Esta utilidad permite simular un entorno presencial en el que el profesor realiza explicaciones sobre una presentación previamente preparada.
- Realización de encuestas. La aplicación permite realizar encuestas en tiempo real, y visualizar inmediatamente los resultados. Esto es especialmente útil, por ejemplo, para captar opiniones con rapidez o evaluar el nivel de conocimientos alcanzado por los alumnos durante la sesión.
- Grabación de sesiones. La herramienta permite grabar sesiones y hacerlas posteriormente disponibles a través de la plataforma. De esta forma es posible que alumnos que no han asistido a una sesión puedan visualizarla en su equipo posteriormente.
- Control de permisos. El profesor puede controlar los permisos de cada estudiante en cada momento. Por ejemplo, el profesor puede dar o denegar permiso para escribir en la pizarra virtual, y cada usuario pueden controlar quién tiene acceso a sus aplicaciones compartidas.

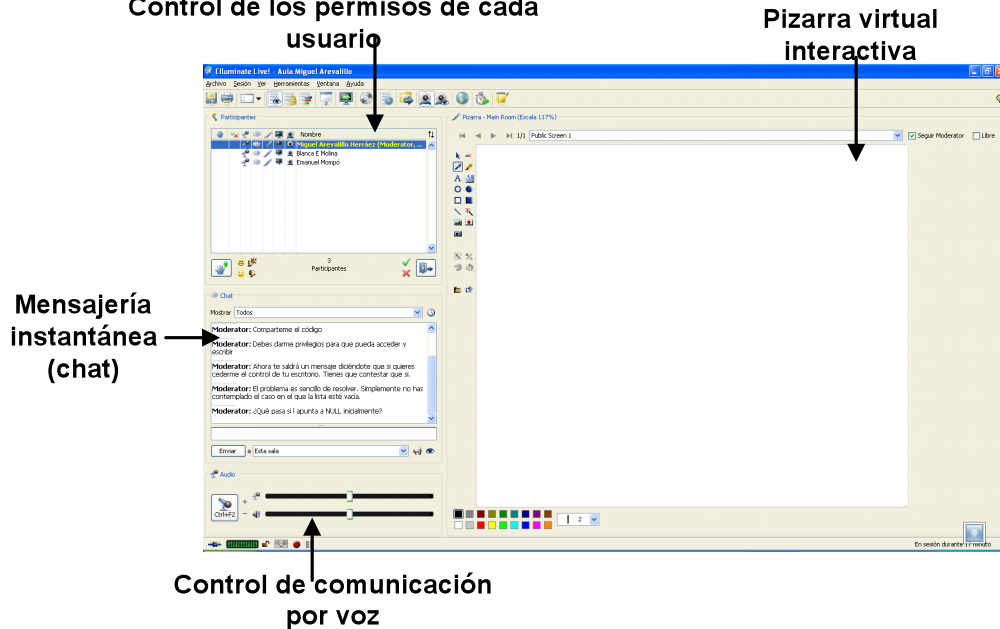


Figura 1. Captura de pantalla del entorno de la aplicación

De esta forma, Elluminate live! permite a profesores y estudiantes interactuar y colaborar de forma síncrona y en tiempo real. Con propósitos ilustrativos, la Figura 1 muestra una captura de pantalla del entorno de la aplicación.

3. Caso práctico 1: Asignatura de programación

3.1. Descripción de la asignatura

La asignatura “Informática” es una asignatura de primer curso de la titulación de Matemáticas, cuyos contenidos se centran principalmente en la enseñanza de la programación en C/C++. En el caso de la licenciatura, la asignatura se componía de 10,5 créditos (6 teóricos y 4,5 prácticos). En el caso del grado, su duración ha sido reducida a 6 créditos, siendo la distribución de trabajo presencial en el aula la que se expone a continuación: clases de teoría (22,5 horas), clases de práctica (30 horas), seminarios y tutorías (7,5 horas) y actividades de evaluación (7,5 horas).

Desde sus comienzos, un objetivo principal del profesorado de la asignatura ha sido incrementar el seguimiento y apoyo directo al

alumnado. Con este propósito, y en consonancia con la inclusión de la asignatura en un Plan de Innovación Educativa que afectaba a la Licenciatura de Matemáticas, se desarrollaron nuevas estrategias docentes centradas en el estudiante, incluyendo a) la instauración de laboratorios abiertos adicionales y fuera del horario habitual de la asignatura; b) la incorporación de los denominados alumnos-tutores que actuaban como ayudantes del profesor colaborando en la corrección de trabajos, provisión de realimentación al alumno y supervisión de los laboratorios abiertos; c) la inclusión de prácticas competitivas.

3.2. Problema

Aunque las clases teóricas constituyen el primer paso en el aprendizaje de cada concepto de programación, el papel más activo del alumno se produce durante las sesiones prácticas de la asignatura. En ellas los alumnos, en parejas, deben enfrentarse a la resolución de problemas de programación que utilizan las estructuras de datos/construcciones aprendidas durante las sesiones de teoría. Para ello, se utiliza el entorno de desarrollo de código abierto Dev-C++ [2].

Desafortunadamente, la atención simultánea a grupos de prácticas compuestos de 30 alumnos no es tarea sencilla, y está sujeta a los siguientes problemas:

1. En ocasiones, un pequeño problema de compilación puede paralizar la tarea del estudiante hasta que éste es atendido.
2. Gran parte de las cuestiones/dudas comunes se atienden a nivel individual. Hasta que se advierte que la duda es común a un grupo extenso de estudiantes y se procede a realizar una explicación a nivel global puede haberse incurrido en una pérdida de eficiencia significativa.
3. Dependiendo de la situación, los tiempos de espera pueden alargarse y causar frustración en el alumno debida a una sensación de falta de apoyo.
4. Los alumnos con menos conocimientos suelen acaparar la mayor parte del tiempo del docente con cuestiones que, en muchos casos, se deben a su falta de esfuerzo.

Además, las prácticas han sido diseñadas para que el alumno deba dedicar parte de su trabajo personal fuera del aula a su terminación. Los problemas a los que puede tener que enfrentarse fuera del aula (complicación, resultados incorrectos que no entiende, etc.) pueden requerir la asistencia a tutorías individualizadas. En este caso, el alumno se verá obligado a alterar su planificación de trabajo, deteniendo su tarea hasta poder ser atendido (que pueden ser días según la disponibilidad del docente), e incurrir en las pérdidas de tiempo asociadas al desplazamiento. La asiduidad de dudas puntuales que requieren una breve intervención del docente pero que paralizan la actividad del alumno constituye una fuente habitual de sensación de desamparo que puede tener efectos negativos sobre la motivación del estudiante por la asignatura.

3.3. Enfoque adoptado

En este caso, la herramienta Elluminate Live! fue utilizada para la realización de tutorías virtuales. Entre los deberes de los profesores a tiempo completo en las universidades presenciales se encuentra la obligación de estar disponibles durante un número de horas semanales para la realización de tutorías individualizadas. En la mayoría de los casos, el alumno concentra el uso de esta disponibilidad en fechas próximas a

pruebas de evaluación o entregas de trabajos, siendo escasamente utilizadas durante el resto del curso. Por ello, durante el curso 2008/09 se propuso que dos de las horas dedicadas a tutorías fueran online y se atendieran a través de la herramienta. La intención inicial es que los alumnos fueran capaces de utilizar estas dos horas semanales para completar bajo supervisión el trabajo iniciado en las prácticas, pero las ventajas del uso de la herramienta fueron más allá. A continuación comentamos algunas de las más relevantes.

- *Atención simultánea.* Durante las sesiones prácticas, las explicaciones se realizan de forma individualizada a parejas de estudiantes. Al realizar las explicaciones a través de la plataforma, todos los alumnos conectados tienen la opción de escucharlas. Asimismo, cuando se detecta un problema común a varios alumnos, puede realizarse una explicación dirigida al grupo completo. Con respecto a una tutoría presencial, la herramienta permite atender a varios alumnos que trabajan de forma simultánea, y compartir explicaciones a problemas comunes.
- *Resolución rápida de problemas.* La utilidad de compartición de aplicaciones supone una ayuda inestimable en el contexto de esta asignatura, el permitir que el alumno comparta su entorno de desarrollo (Dev-C++) con el profesor. Esto facilita la resolución rápida de problemas de compilación, o la detección de errores lógicos en los programas; y evita el caso habitual de alumnos asistiendo a tutorías en repetidas ocasiones durante un mismo día acompañados de su portátil, para resolver pequeños problemas de compilación o de uso del entorno. En este sentido, la aplicación permite una “atención discontinua” al alumno, bastante adecuada en este contexto.
- *Evita desplazamientos.* El alumno puede continuar con la realización de la práctica desde casa o desde la biblioteca, con el único requisito de tener disponibles una conexión a Internet y un navegador Web. Además, permite que alumnos que no han podido asistir a las sesiones prácticas por motivos justificados realicen las actividades encomendadas bajo supervisión directa.
- *Fomenta la participación y la cooperación entre los alumnos.* Durante las sesiones de

tutorías virtuales, los alumnos colaboran en la resolución de los problemas que les van surgiendo. Se ayudan entre ellos en la detección de problemas de compilación, comentan errores a través de la plataforma y se explican algunos conceptos. Estas acciones, además de tener un efecto positivo sobre el aprendizaje, descargan la labor del docente.

- *Facilita el seguimiento del alumno.* Las tutorías virtuales ayudan al profesor a detectar posibles lagunas conceptuales en el aprendizaje y a conocer mejor a su grupo de estudiantes.
- *Ayuda a la planificación.* La plataforma constituye un entorno más distendido y relajado que el aula, permitiendo que el alumno trabaje a su propio ritmo y obtenga atención cuando la necesite. Esto elimina la necesidad de asistir a tutorías individualizadas para resolver dudas puntuales que paralizan su actividad. Mediante las tutorías virtuales, el alumno tiene al profesor a su disposición de un modo fácilmente accesible, pudiendo obtener ayuda de forma prácticamente inmediata y continuar con su actividad.
- *Soporte para enseñanza a distancia.* Aunque no ha sido el caso en esta asignatura, la herramienta también facilitaría el seguimiento de la asignatura a estudiantes discapacitados o que se encuentran en el extranjero cursando programas de intercambio internacional.

3.4. Resultados

El uso de la herramienta se puso en marcha durante el segundo cuatrimestre. Tras cuatro semanas desde su implantación, todos los estudiantes que asistían a las sesiones de tutorías presenciales con regularidad habían hecho uso de la herramienta, reduciendo su asistencia a las sesiones de tutorías presenciales un 75%. Según las opiniones de los alumnos obtenidas a partir de conversaciones directas mantenidas con ellos, encontraban el uso de la herramienta muy conveniente para la resolución de problemas puntuales o de compilación, pero preferían la interacción directa con el profesor cuando requerían explicaciones de mayor envergadura. Por esta razón continuaban asistiendo a tutorías personalizadas para resolver este tipo de cuestiones. En general consideraban las tutorías virtuales como un complemento ideal para las

tutorías presenciales, pero no como un reemplazo de las mismas.

Desafortunadamente, no hemos podido evaluar objetivamente el impacto del uso de la herramienta en los resultados académicos de los alumnos. La introducción del uso de la herramienta coincidió con la adopción de nuevas técnicas docentes en el marco de un proyecto de innovación educativa que afectaba a todo el primer curso de la titulación de Matemáticas. Este hecho impide poder aislar el impacto de la utilización de la plataforma.

4. Caso 2: Asignatura de máster oficial universitario

4.1. Descripción de la asignatura

La asignatura “Sistemas inteligentes distribuidos y basados en agentes” forma parte del módulo obligatorio en “Aprendizaje, percepción y procesado de información visual y multimodal” del Máster oficial universitario en Computación Avanzada y Sistemas Inteligentes de la Universidad de Valencia. La asignatura estudia técnicas modernas de Inteligencia Artificial, en concreto bajo el paradigma de agentes, tal como están siendo usadas hoy día para abordar la resolución de problemas complejos.

La asignatura cubre 5 créditos ECTS que, según la equivalencia establecida por la universidad, corresponden a 125 horas de trabajo del alumno. Esta carga ha sido organizada en 31 horas de asistencia a clases teórico-prácticas y 10 horas de asistencia a seminarios de investigación, repartiéndose el resto de horas en distintas tareas no asistenciales. El número de estudiantes matriculados en el curso académico en el que se realizó la experiencia docente fue 10

El máster en Computación Avanzada y Sistemas Inteligentes, es el resultado de la adaptación al Real Decreto (RD) 1393/2007 del programa de doctorado en Informática y Matemática Computacional (RD 778/1998). En su docencia participa tanto el Departamento de Informática como el Departamento de Matemática Aplicada, involucrando a un total de 5 profesores para la asignatura que estamos considerando.

El máster tiene como objetivo principal la formación investigadora de científicos e ingenieros en los aspectos relacionados con la

computación eficiente de problemas complejos. Por tanto, se considera muy relevante la colaboración de científicos y profesores de otras universidades nacionales y extranjeras. Es por ello que, además del profesorado indicado anteriormente, la asignatura cuenta con la participación estable en los seminarios de investigación de un profesor que reside en el extranjero.

4.2. Problema

En este caso, surgen una serie de problemas relacionados con los perfiles del profesorado y del alumnado de la asignatura, así como con las aplicaciones científicas utilizadas durante el aprendizaje.

De un lado, la convergencia en un solo rol de las misiones docente, investigadora y de transferencia del conocimiento ha demostrado que aporta grandes beneficios a la docencia, especialmente a la de postgrado. Sin embargo, diversos eventos pueden también provocar que la vida investigadora de un profesor interfiera con la impartición de sus clases. Por ejemplo: la asistencia a congresos, las reuniones científicas, las estancias en otras universidades o centros de investigación, etc. El problema de coordinación se agrava cuando la docencia se comparte con un nutrido grupo de profesores, incluso provenientes de otros centros, como en el caso que nos ocupa.

Además, los estudiantes de máster a menudo se encuentran ya inmersos en el mercado laboral, lo que limita su flexibilidad a la hora de cambiar horarios y planificaciones docentes. En nuestro caso, el 80% del alumnado simultaneaba el estudio del máster con algún tipo de trabajo, lo que reducía considerablemente su disponibilidad.

Así pues, tanto el profesor universitario como su alumnado pueden encontrarse ocasionalmente en lugares diversos del mundo, lo que dificulta que compatibilicen su vida laboral o investigadora con la docente. El uso de herramientas que ayuden a salvar esta distancia puede ayudar a solucionar dichos inconvenientes.

Por otro lado, frecuentemente el profesorado necesita apoyarse en un conjunto de aplicaciones científicas, de uso habitual en el contexto de la disciplina impartida. En concreto, la asignatura utiliza una plataforma de desarrollo de sistemas multi-agente (Jason [6]) para la simulación de distintos problemas como la dinámica de

poblaciones o la evolución de distintos modelos de económicos. La ejecución de estos sistemas complejos requerirá del profesor tanto la demostración de funcionamiento como la intervención ante posibles errores de reproducción por parte de los alumnos. Por esta razón, es muy conveniente el uso de una herramienta que supere las capacidades típicas de la videoconferencia y aporte un entorno de trabajo colaborativo donde los usuarios puedan compartir dichas aplicaciones científicas.

4.3. Enfoque adoptado

En este caso, la herramienta Elluminate Live! se utilizó para la realización de dos clases teórico-prácticas (a cargo del profesorado del máster) y de un seminario de investigación (realizado por el profesor externo). En todas las sesiones, el docente se encontraba en el extranjero mientras que a los alumnos, a parte de la posibilidad de conectarse desde sus domicilios, se les ofreció un aula universitaria donde un profesor adicional les brindaba su apoyo ante pequeños problemas técnicos surgidos por el uso de la herramienta. Además, dicho profesor se encargaba de moderar, gestionar y organizar la sesión. Todas las sesiones se estructuraron de acuerdo con las siguientes tres fases:

1. El profesor en el extranjero realizaba una presentación en la pizarra interactiva, sobre la que planteaba una serie de preguntas que debían ser resueltas por los estudiantes, previa obtención del permiso de edición de la pizarra.
2. El profesor en el extranjero mostraba el funcionamiento de un sistema multi-agente, diseñado para simular algún ejemplo concreto mediante la compartición de aplicaciones.
3. Se planteaba una extensión del ejemplo anterior como ejercicio individual. Durante su realización, el profesor en el extranjero respondía dudas particulares mediante el chat privado y, cuando la situación lo requería, los alumnos compartían sus aplicaciones para recibir una supervisión adicional.

De manera general, la aproximación utilizada en este caso ofrece los siguientes beneficios:

- *Compatibiliza la vida docente, investigadora y laboral.* La docencia a distancia permite un modelo de movilidad más flexible, tanto del profesorado como del alumnado matriculado en los másteres universitarios.

- *Promueve la internacionalización.* La facilidad de incorporar expertos de otros centros de investigación y universidades nacionales o extranjeras favorece la calidad de la docencia impartida.
- *Reduce costes.* La posibilidad de participar en la docencia de manera remota evita el desplazamiento y reduce tanto los costes económicos como los temporales.
- *Facilita la supervisión del trabajo del alumno.* La capacidad de compartir aplicaciones de Elluminate Live! es una funcionalidad muy útil para la demostración del funcionamiento de aplicaciones científicas complejas y para la supervisión del trabajo del alumno.

4.4. Resultados

La experiencia que se ha descrito ha sido catalogada como muy positiva tanto por parte del equipo docente como por parte de los estudiantes. Conviene remarcar el entusiasmo con el que los alumnos recibieron la iniciativa y el alto grado de participación alcanzado durante las sesiones. Al finalizar las mismas, los alumnos mostraron su interés en repetir la experiencia así como, si fuera posible, ampliarla para que incluya a otros investigadores extranjeros que son referencia en el campo de los sistemas inteligentes basados en agentes.

No obstante, cabe destacar que los alumnos hicieron un uso conservador de la herramienta, ya que una media del 60% de los alumnos optó por acudir al aula de la universidad. Por tanto, como objetivo para el presente curso se encuentra el reproducir la experiencia completamente a distancia, de tal manera que tanto los profesores como los alumnos estén separados físicamente.

5. Conclusiones

Aunque la mayoría de las universidades actualmente utilizan algún tipo de plataforma para dar soporte a la docencia, la utilización de herramientas de comunicación síncrona no es tan habitual. En este artículo hemos presentado dos casos en los que se ha utilizado una aplicación de este tipo para mejorar la docencia en una asignatura de grado y en otra de máster oficial.

Aunque este artículo se ha concentrado en dos casos prácticos puestos en marcha por los autores, el uso de este tipo de herramientas no se limita a

estos casos particulares. A modo de ejemplo, el departamento ha hecho uso de la misma herramienta para permitir a un alumno presentar su proyecto final de carrera a través de la plataforma, mientras se encontraba en el extranjero; y personal de esta misma universidad la ha utilizado para proporcionar tutorías virtuales durante los fines de semana. Además, existen otras universidades que utilizan Elluminate Live! como sistema de videoconferencia en contextos docentes [11].

El uso de plataformas que incorporen herramientas de comunicación síncrona específicamente diseñadas para la docencia, abre nuevas posibilidades en este ámbito. Son especialmente relevantes posibles aplicaciones para facilitar el acceso a la educación a discapacitados, o a individuos que deben compaginar su vida estudiantil y laboral, especialmente en el contexto de la educación a distancia.

El relativo éxito del uso de Elluminate Live! ha llevado a la universidad a negociaciones con la empresa suministradora para obtener una licencia de Campus, y a contemplar otras alternativas, como Adobe Connect, principalmente como medio de soporte a la docencia presencial.

Agradecimientos

Los autores agradecen la financiación recibida desde el Vicerrectorado de Convergencia Europea y Calidad de la Universitat de València, a través de los proyectos DocenTIC y Finestra Oberta con códigos 08/DT/04/2009, 18/DT/05/2010 y 47/FO/35/2010; y desde el Ministerio de Educación y Ciencia y FEDER, mediante el proyecto Consolider Ingenio 2010 CSD2007-00018.

Referencias

- [1] Ley Orgánica 4/2007, de 12 de abril, B.O.E. de 13 de abril de 2007.
- [2] Dev C++. Disponible en <http://www.bloodshed.net>, Febrero de 2011.
- [3] Documento Marco del Ministerio de Educación, Cultura y Deporte. *La integración del sistema universitario español en el Espacio Europeo de Enseñanza Superior*, Febrero de 2003. Disponible en

guiándolo durante todo el proceso a través de diferentes mecanismos de evaluación.

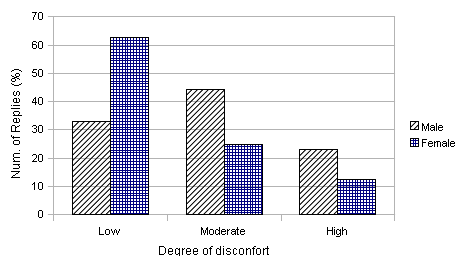


Figura 5. Histograma que describe el grado de incomodidad del estudiante (en función del género) al ser evaluados en clase.

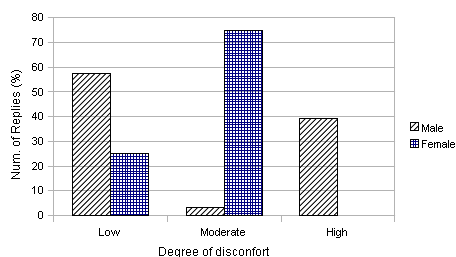


Figura 6. Histograma que describe el grado de incomodidad del estudiante (en función del género) al ser evaluados durante las horas de tutoría.

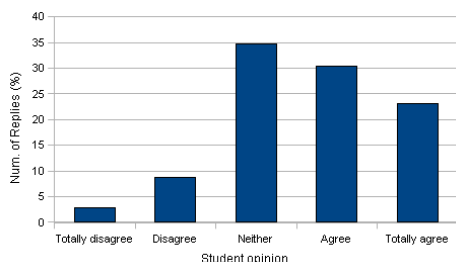


Figura 7. Histograma acerca de la importancia de la retroalimentación del instructor como un mecanismo para mejorar sus capacidades de auto.

6. Conclusiones

En este trabajo hemos analizado las capacidades de auto-evaluación de los alumnos como herramienta de aprendizaje, en el ámbito de un

curso de Redes de Computadores de una titulación de Ingeniería Informática. Nos hemos centrado en las habilidades de auto-evaluación de nuestros estudiantes, así como de sus preferencias sociales relacionadas con la retroalimentación obtenida por parte del instructor. A partir de los datos recibidos de una población de 70 estudiantes de Ingeniería Informática, los resultados muestran que, aunque todos los estudiantes tienden a sobrestimar su calificación del examen, las estudiantes son mucho más optimistas (una diferencia de puntuación de 10% más que los chicos en comparación con el resultado real obtenido). Además, algunos estudiantes (masculinos) creen merecer una nota que está por debajo del resultado real obtenido (hasta un 10% menos). Entre las estudiantes de género femenino no encontramos casos pesimistas en cuanto a la nota que creen merecer.

En general, el estudio realizado revela que las mujeres son más propensas a sobrestimar la calidad de su trabajo. Aunque este resultado está en clara contradicción con trabajos anteriores, consideramos que puede ser debido a que la población de estudiantes de Ingeniería Informática tiene características especiales que la hacen diferente respecto a patrones encontrados en muestras de otras poblaciones.

En cuanto a las preferencias sociales, encontramos que las estudiantes se sienten más cómodas que los varones cuando se está evaluando en clase (63% vs 33%). Sin embargo, hay un grupo significativo de estudiantes que, independientemente de su sexo, se sienten muy incómodo siendo evaluados frente a otros estudiantes (23% hombres y 12% de estudiantes mujeres). Cuando se analizan los resultados de la evaluación realizada en horarios de tutorías, la tendencia es la opuesta: el 57% de los estudiantes masculinos se sienten cómodos, mientras que este porcentaje se reduce al 25% en el caso de las alumnas. Además, alrededor de un 11,6% de los estudiantes tienen una opinión negativa acerca de la retroalimentación del instructor, aspecto que está en clara contradicción con la interacción natural estudiante-profesor de base. Este factor merece un estudio más detallado.

Como trabajo futuro, planeamos realizar estudios similares en próximos cursos que

Cómo empezar fácil con PBL

Miguel Valero-García
Dept. Arquitectura de Computadores
Universidad Politécnica de Cataluña
Esteve Terradas n. 7
08860 Castelldefels
miguel.valero@upc.edu

Javier García Zubia
Dept. Tecnologías Industriales
Universidad de Deusto
Avda. de las Universidades n. 24
48007 Bilbao
zubia@deusto.es

Resumen

Se propone un esquema para el diseño de una actividad de aprendizaje basado en proyectos que tenga en cuenta los factores clave para el éxito. El esquema puede adaptarse a cualquier asignatura y en todo caso puede proporcionar una experiencia modesta pero potencialmente exitosa sobre la cual reflexionar para abordar retos más ambiciosos.

Summary

We propose a scheme to design an activity for project based learning taking into account the key issues to maximize the probability of success. The scheme can be adapted to any subject and can provide a guided experience to reflect on it and to stimulate more ambitious challenges.

Palabras clave

Aprendizaje basado en proyectos, planificación de actividades y entregas, evaluación.

1. Introducción

El aprendizaje basado en proyectos (en inglés Project Based Learning o PBL) [1] está mereciendo cada vez mayor atención, especialmente en el contexto de la enseñanza de la ingeniería. Al menos hay dos motivos para ello. Por una parte, son cada vez más los ejemplos exitosos de aplicación a todos los niveles, desde asignaturas (o partes de asignaturas) de unos pocos créditos [2], pasando por grupos de dos o tres asignaturas que se asocian para favorecer el despliegue de PBL [3] hasta incluso instituciones que utilizan esta metodología como principio organizativo de alguno de (o todos) sus planes de estudio [4].

La segunda razón del creciente interés en PBL es que esta metodología ofrece una respuesta integral a buena parte de los retos que el EEES ha puesto encima de la mesa, como por ejemplo, la utilización eficaz del tiempo de dedicación del

estudiante dentro y fuera de clase (de acuerdo con los créditos ECTS) o el desarrollo de habilidades genéricas a lo largo del plan de estudios (aprendizaje autónomo, trabajo en grupo, etc.).

Sin embargo, PBL no es fácil de aplicar. Incluso en sus dosis más modestas (un proyecto de 4 ó 5 semanas en una asignatura de pocos créditos) PBL implica cambios profundos en la organización del proceso de enseñanza y aprendizaje y en el rol de profesores y estudiantes. Algunos de estos cambios requieren la utilización de técnicas que, con mayor o menor esfuerzo, todos podemos aprender. Pero otros cambios tienen que ver con nuestros principios y creencias en relación a la labor docente, que no siempre son fáciles de cambiar.

Como consecuencia de ello, no son infrecuentes los casos de experiencias de utilización de PBL con resultados frustrantes que lejos de servir de acicate en un proceso de mejora continuada se convirtieron finalmente en un desestímulo y freno a nuevas aventuras o en una confirmación de hipótesis de partida tales como *“los estudiantes no están preparados para esto”*.

De acuerdo con la experiencia que hemos acumulado en la utilización de PBL en nuestras asignaturas y en labores de promoción de la metodología y formación de profesorado, creemos que es posible ofrecer unas pautas muy concretas que pueden permitir la introducción de PBL de manera que se tengan en cuenta los factores esenciales para el éxito. Si bien es cierto que en docencia no hay “recetas” que sirvan para todos los casos, creemos que estas pautas para la introducción de PBL pueden resultar eficaces para ayudar a cosechar un primer éxito que, aunque modesto, nos ayude a afrontar aventuras más ambiciosas. En este artículo describimos estas pautas e ilustramos su aplicación con un ejemplo. Los autores han utilizado (o han ayudado a utilizar) estas pautas en una variedad de casos y contextos con buenos resultados, por lo que consideran que estas pautas pueden ser de interés para muchos.

CUATRO ERRORES HABITUALES	
#1 Asignar un bajo peso al proyecto en la calificación de la asignatura	Si el peso es bajo los estudiantes no se tomarán en serio el trabajo y los resultados serán decepcionantes (y algunos copiarán).
#2 No realizar un seguimiento del trabajo de los estudiantes	Se dejarán el trabajo para el último momento y tendrán dificultades sin tiempo de reacción (incluso copiarán). Este error se da con frecuencia combinado con el anterior error (y la combinación es explosiva).
#3 Mantener un examen final tradicional (incluso aunque tenga poco peso)	Cuando el peso del proyecto y el seguimiento son adecuados, el esfuerzo de los estudiantes es mayor y los resultados mejores. Pero entonces dedicarán menos esfuerzo a preparar el examen final tradicional (que tendrá un peso pequeño). Además, probablemente realizar un proyecto no es la mejor forma de preparar el examen. En cualquier caso, los resultados del examen serán decepcionantes (y nos asaltarán dudas de que el proyecto sirva para aprender).
#4 Asignar un peso en la nota a cada entrega	A mitad de camino, los estudiantes que hayan acumulado ya muchos puntos se relajarán y los que hayan conseguido pocos se relajarán más (y algunos abandonarán).
CINCO FACTORES CLAVE PARA EL ÉXITO	
#1 Plantear un reto ambicioso (pero asequible) con criterios claros	Los estudiantes (y en general, las personas) se motivan más cuando se les plantea un reto ambicioso (incluso que parece por encima de sus posibilidades). Si además los criterios de calidad del producto y los criterios de evaluación del curso están claros es más fácil que hagan un buen trabajo (y será más fácil para nosotros evaluarlo).
#2 Planificar el trabajo de forma minuciosa	La mejor forma de que los estudiantes perciban que el reto ambicioso es a la vez asequible es presentarles un plan detallado de lo que tienen que hacer semana a semana para alcanzar el objetivo.
#3 Generar interdependencia positiva y exigibilidad individual	Los estudiantes deben tener la percepción de que sólo van a conseguir el éxito si todos los miembros del grupo se implican en el trabajo (interdependencia positiva). Además, debe quedar claro que cada estudiante deberá rendir cuentas individualmente de su nivel de aprendizaje, según los objetivos formativos establecidos (exigibilidad individual).
#4 Realizar un seguimiento del trabajo que realizan los estudiantes	El proyecto debe tener asociado una serie de entregas, no sólo finales sino también a lo largo del proceso. Estas entregas motivarán a los estudiantes a realizar el trabajo de manera continuada y permitirán al profesorado hacer un seguimiento. Especialmente importante es planificar una entrega tipo “primera versión del producto final” que permita al profesorado tener una visión global del proyecto a tiempo de intervenir si es necesario para ayudar a conseguir el éxito final.
#5 Diseñar adecuadamente el método de evaluación	El método de evaluación debe proyectar nitidamente la idea de que para aprobar la asignatura no hay más camino que trabajar de manera continuada, hacer un buen proyecto y superar las exigencias de aprendizaje individual establecidas.

Tabla 1: Errores habituales y factores claves para el éxito en la aplicación de PBL

2. Errores habituales y factores clave

Las experiencias frustrantes en la utilización de PBL no son infrecuentes. En algunos casos el mal resultado puede ser debido a que la metodología no es adecuada para el contexto en el que se ha aplicado (naturaleza de la materia, objetivos formativos, tipo de estudiantes, etc.). Sin embargo, casi siempre el problema radica en que se ha cometido alguno de los errores habituales que se describen en la tabla 1. A partir de esta lista de errores habituales no resulta difícil derivar una lista de factores clave para el éxito en la implantación de PBL (o al menos para reducir las probabilidades de fracaso). Los cinco factores clave que consideramos más importantes se describen también en la tabla 1. En la siguiente sección se propone una guía para la introducción de PBL de manera que se tengan en cuenta los factores clave para el éxito (y se eviten, por tanto, los errores habituales).

3. El esquema propuesto

El esquema propuesto permite diseñar una actividad PBL que puede funcionar bien en grupos de clase de hasta 60 alumnos, con una carga de trabajo asumible para el profesorado en el tiempo correspondiente a la dedicación típica semanal (8 horas de clase y 6 de consulta).

La tabla 2 muestra el esquema para la introducción de la actividad PBL, en forma de planificación de actividades y entregas. Con fondo gris se muestran las actividades y entregas que se realizan en clase y con fondo blanco las que se realizan fuera de clase. En negrita se indican las tareas fuera de clase que los estudiantes deben realizar en grupo base (un grupo base es un equipo de estudiantes que va a realizar el proyecto). Junto a cada entrega se indica (entre paréntesis) si se trata de una entrega individual (I), de grupo base (G) o de grupo temporal (GT).

Tareas	Entregas
Presentación y demostración del proyecto Formación de grupos Presentación del puzzle y reparto	#1 (G) Acta de constitución de grupo
Estudio individual	#2 (I) Dudas y resultados
Reunión de expertos Realización de un ejercicio de profundización	#3 (I ó GT) Resultado del ejercicio de profundización
Preparación individual de explicación, ejercicios y soluciones Explicaciones en grupo, por turnos Realización de los ejercicios con autoevaluación	#4 (I) Ejercicios para compañeros #5 (I) Autoevaluación
Aclaración de dudas Presentación del ejercicio de integración	
Realización del ejercicio de integración	
Demostración del ejercicio de integración Presentación de ampliaciones individuales y reparto	#6 (G) Ejercicio de integración
Realización de la ampliación individual Integración de ampliaciones individuales para producir el primer prototipo	
Demostración del primer prototipo Planificación de la versión final	#7 (G) Primer prototipo #8 (G) Plan
Tareas de acuerdo con el plan (algunas individuales y otras en grupo)	
Demostración de la versión final Realización del ejercicio individual	#9 (G) Versión final #10 (I) Resultado del ejercicio individual

Tabla 2: Esquema propuesto para el diseño de la actividad PBL

El grupo temporal está formado por miembros de grupos base diferentes que se reúnen temporalmente para realizar una tarea concreta. En función del número ECTS de la asignatura, del número de sesiones de clase a la semana y del tiempo de dedicación a cada actividad, el plan puede adaptarse para que la actividad tenga una duración de entre 4 y 10 semanas. En los apartados siguientes se dan más detalles de los diferentes elementos que configuran la propuesta.

3.1. El proyecto

El proyecto (que será el mismo para todos los grupos) debe tener como resultado uno o varios productos que elaboran los estudiantes en equipo (un informe, un programa de ordenador, una presentación oral, etc.).

Hay que prever una primera versión del producto (primer prototipo) y una versión final. El primer prototipo debe contener lo esencial del proyecto y debe entregarse a tiempo de poder hacer rectificaciones de cara a la versión final.

La versión final debe contener los elementos del primer prototipo ya correctamente implementados (atendiendo a la evaluación formativa del primer prototipo), elementos adicionales obligatorios y elementos adicionales opcionales, que incluso pueden ser determinados

por cada grupo según sus intereses. Así queda muy claro lo que es obligatorio para todos y lo que es opcional “para nota”.

El enunciado del proyecto debe venir acompañado con una rúbrica que especifique los criterios de calidad para el producto final. En esa rúbrica debe quedar claro, entre otras cosas, el peso que tienen en la nota los elementos opcionales.

En la sesión en la que se presenta el proyecto y los criterios de calidad y de evaluación debe realizarse, si es posible, una demostración de proyectos de cursos anteriores que normalmente ayudará a proyectar la idea de que el reto es ambicioso.

3.2. Formación de grupos

Los grupos serán de 3 personas (y alguno de 4 si es necesario). La formación de los grupos puede ser aleatoria, como ellos quieran, o según unos criterios establecidos por el profesor. En todo caso, en clase, después de la presentación del proyecto y la formación de grupos (si no han sido formados antes) los grupos se reúnen para elaborar el acta de constitución del grupo, donde deben indicar:

1. Nombre de los miembros que forman el grupo

2. Franjas horarias fuera de clase en las que todos los miembros del grupo están libres para reuniones
3. Acuerdo en cuanto al nivel de ambición del grupo (“*todos vamos a por la máxima nota*” o “*todos vamos simplemente a por el aprobado*”)

Si el grupo no encuentra franjas horarias libres en común o no están de acuerdo en el nivel de ambición entonces se producen los cambios de grupo que sean necesarios.

3.3. El puzle

El trabajo inicial del proyecto debe repartirse en tres tareas independientes. Esas tareas pueden consistir en el estudio de material de teoría, el estudio y puesta a punto de herramientas para el proyecto, la búsqueda de información, etc.

Cada miembro del grupo base se asigna una de esas tareas, que debe realizar de forma individual. En el caso de grupos de 4, uno de los temas (por ejemplo, el más difícil) es preparado independientemente por dos miembros del grupo.

Después del trabajo individual, en clase se realiza la reunión de expertos. Miembros de grupos base distintos pero que han realizado la misma parte del puzle se reúnen (en grupos de 3 ó 4) para compartir dudas y profundizar en el aprendizaje de su parte del trabajo (a través de un ejercicio de profundización proporcionado por el profesor, que puede realizarse de forma individual o en grupos de expertos).

Acabada la sesión de clase, cada miembro del grupo base se prepara en casa una explicación de lo aprendido para los compañeros de grupo. Además prepara algunos ejercicios con soluciones que entregar a sus compañeros para que practiquen con el tema que les explicará.

Después, cada grupo base debe reunirse para que, por turnos, cada miembro explique al resto su parte del trabajo y reparta a los compañeros (con copia al profesor) los ejercicios preparados. Finalmente, cada estudiante intenta resolver los ejercicios recibidos y realiza un informe de autoevaluación a partir de la comparación de sus soluciones con las proporcionadas por los compañeros.

3.4. Ejercicio de integración

Finalizado el puzle y tras posibles aclaraciones que realiza el profesor en la sesión 3, se presenta el ejercicio de integración. Se trata de una tarea

para la que se necesitan las tres partes del puzle y que, por lo tanto, deben hacer los tres miembros del grupo juntos. Además, este ejercicio de integración es la “semilla” del proyecto. Conviene iniciarlo en clase bajo la supervisión del profesor. El ejercicio se completa fuera de clase. Se espera que esta tarea la realicen en grupo (es decir, todos los miembros del grupo juntos realizando el ejercicio).

3.5. Ampliaciones individuales y primer prototipo

Una vez entregado el ejercicio de integración se proponen tres ampliaciones individuales a ese ejercicio (las mismas ampliaciones para todos los grupos), diseñadas de tal manera que el ejercicio de integración más las tres ampliaciones constituyan esencialmente el primer prototipo del proyecto. Idealmente cada una de las tres ampliaciones requiere el uso de los conocimientos de las tres partes del puzle. Cada miembros de grupo hace una de las ampliaciones y luego el grupo se reúne para integrar las ampliaciones y poner a punto el primer prototipo. Debe preverse una cuarta ampliación para el caso de los grupos de 4 estudiantes.

3.6. Versión final

Después de la entrega y evaluación del primer prototipo, cada grupo debe abordar la realización de la versión final, para lo cual debe planificar el trabajo a realizar (definición de tareas, fechas de entregas y reparto de trabajo). Naturalmente, ese trabajo puede incluir el estudio de temas nuevos. En todo caso, en la versión final deben incorporarse:

1. Las mejoras que se deriven de la evaluación del primer prototipo realizada por el profesor
2. Los elementos obligatorios correspondientes a la versión final
3. Los elementos optativos que el grupo haya acordado

3.7. Conocimientos básicos

Conviene identificar los conocimientos básicos que todos y cada uno de los estudiantes deben conseguir en relación a los temas implicados en el proyecto, con independencia del tema en el que se ha especializado cada estudiante como consecuencia de la mecánica del puzle. Esos conocimientos básicos que todos deben tener se

evaluarán de forma individual (por ejemplo, mediante exámenes). En el esquema mostrado en la tabla 2 no se indica en qué momento se realiza la evaluación de los conocimientos básicos (probablemente un examen en clase). Además, es conveniente que los estudiantes tengan al menos dos oportunidades para examinarse de los conocimientos básicos. En todo caso, al adaptar el esquema al caso concreto será necesario ubicar en el tiempo estas actividades de evaluación de conocimientos básicos.

Es importante graduar bien la cantidad de conocimientos básicos. Una cantidad excesiva no permitirá a los estudiantes especializarse en ninguno de los temas, poniendo en riesgo el nivel de ambición del proyecto. Pero una cantidad pequeña pone en riesgo la exigibilidad individual.

3.8. Calificación

La calificación de la asignatura debe obtenerse a partir de tres elementos: las entregas del curso, los resultados del proyecto y los conocimientos básicos. Por una parte, los estudiantes deben poder obtener entre un 10% y un 20% de la calificación final por el mero hecho de realizar las entregas del curso a tiempo. Estamos suponiendo aquí que se han planificado entregas a lo largo de todo el curso y no sólo durante el tiempo que dura el proyecto. Puede ser conveniente establecer algún requisito mínimo de calidad para que una entrega sea aceptada (por ejemplo, que use la plantilla establecida, que no tenga faltas de ortografía, etc.), pero la idea es premiar con esta componente el esfuerzo continuado de los estudiantes.

Por otro lado, debe establecerse el requisito de realizar a tiempo al menos un cierto porcentaje de entregas (por ejemplo un 80%), sin el cual no puede superarse la asignatura. De este modo, resulta evidente que no hay más camino para aprobar la asignatura que llevar a cabo el plan de actividades y entregas establecido.

El porcentaje de calificación asignado a los resultados del proyecto debe ser aproximadamente el 50% (podría ser algo inferior si la actividad se extiende a lo largo de menos de 6 semanas). Esta calificación se obtiene a partir de tres elementos:

1. Calificación de la versión preliminar del producto del proyecto (por ejemplo, el 10%)
2. Calificación de la versión final (por ejemplo, el 25%)
3. Ejercicio individual (por ejemplo, el 15%)

Tanto la versión preliminar como la versión final del producto del proyecto deben evaluarse a partir de la rúbrica que expresa los criterios de calidad y la calificación obtenida será la misma para todos los miembros del grupo. El ejercicio individual consiste en una modificación del producto del grupo que debe resultar sencilla para los estudiantes que dominan ese producto (aunque hayan trabajado más en unas partes que en otras) y debe resultar difícil para el estudiante que se haya desentendido del trabajo asignado a sus compañeros (o del trabajo que le fue asignado a él). Además, la máxima calificación en este elemento de la nota del proyecto sólo debe obtenerse en el caso de que todos los miembros del grupo realicen correctamente el ejercicio individual.

Finalmente, la componente de conocimientos básicos debe tener un peso de alrededor del 30%. En todo caso, lo importante es que cada estudiante debe demostrar a lo largo del curso que ha adquirido todos los conocimientos básicos establecidos. Si no es el caso entonces no puede aprobar la asignatura, aunque haya realizado un gran proyecto y haya realizado a tiempo todas las entregas. Como se indicó antes, los estudiantes deberían tener, durante el curso, al menos dos oportunidades de demostrar cada conocimiento básico.

3.9. Observación final

La tabla 3 explica brevemente de qué manera el esquema propuesto integra los cinco factores clave para el éxito descritos en la sección 2. Por otro lado, como se ha comentado ya, el esquema deberá adaptarse a cada caso particular de aplicación. Por ejemplo, en el caso de un proyecto pequeño, puede haber dos sesiones de clase por semana, pero con tiempo suficiente entre ellas para que los estudiantes realicen el trabajo fuera de clase. En otros casos puede ser conveniente que haya una sola sesión de clase (de dos o tres horas) cada semana. Por otra parte, dependiendo del tiempo disponible y de la naturaleza del proyecto, puede ser conveniente que haya dos semanas o más entre la entrega de la versión preliminar y la entrega de la versión final, de manera que los estudiantes tengan la oportunidad de planificar un bloque significativo de tareas y de tiempo.

CINCO FACTORES CLAVE PARA EL ÉXITO	
#1 Plantear un reto ambicioso (pero asequible) con criterios claros	El reto se percibirá como ambicioso en la medida en que los estudiantes queden impresionados con el enunciado y, especialmente, con la demostración de proyectos de cursos anteriores. Las rúbricas que acompañarán la presentación del proyecto garantizan que los criterios de calidad y evaluación sean claros y conocidos con antelación
#2 Planificar el trabajo de forma minuciosa	El recorrido desde la presentación del proyecto hasta la entrega del primer prototipo está minuciosamente planificado por el profesor. La planificación del trabajo a realizar entre el primer prototipo y la versión final la realizan los estudiantes y deberá ser también minuciosa.
#3 Generar interdependencia positiva y exigibilidad individual	La interdependencia positiva se consigue mediante tres mecanismos: (a) un dimensionado del tiempo de dedicación a cada tarea que se ajuste a la dedicación esperada de cada estudiante de acuerdo con los ECTS de la asignatura, (b) la asignación de roles (hay un experto en cada tema) que hace que se necesiten mutuamente para reunir todos los conocimientos necesarios y (c) el método de calificación que, a través del ejercicio individual final, hace que el éxito total de cada estudiante dependa en parte del éxito de sus compañeros. La exigibilidad individual se garantiza a través de la evaluación de los conocimientos básicos.
#4 Realizar un seguimiento del trabajo que realizan los estudiantes	El esquema contempla 10 entregas razonablemente distribuidas a lo largo del tiempo. Algunas de ellas no requieren un esfuerzo de evaluación importante por parte del profesorado (acta de constitución de grupo o autoevaluación de ejercicios de compañeros). Otras requieren mayor esfuerzo porque son críticas para que el proyecto acabe bien (especialmente, la evaluación del primer prototipo).
#5 Diseñar adecuadamente el método de evaluación	El método de evaluación proyecta claramente los tres mensajes clave: (a) hay que hacer el trabajo planificado, porque sólo se aprueba si se realiza al menos el 80% de las entregas, (b) hay que esforzarse para que el grupo funcione bien y hagan un buen trabajo, porque el proyecto tiene un buen porcentaje de la nota y (c) no se puede desatender el aprendizaje individual, porque sin los conocimientos básicos no se aprueba.

Tabla 3: Verificación de que el esquema propuesto contiene los factores clave para el éxito

4. Ejemplo

4.1. Contexto

El proyecto se muestra como ejemplo se desarrolla entre las semanas 9 y 15 de un curso de introducción a la programación en lenguaje C, de 6 ECTS. La semana 16 corresponde a la semana de exámenes finales. Puesto que la asignatura tiene 6 ECTS, los estudiantes deben dedicar unas 9 horas a la semana (150 horas distribuidas entre 16 semanas), de las cuales 3 son en clase (una sola sesión semanal) y 6 fuera de clase (en adelante, nos referiremos a la sesión de la semana *i* como la sesión *i*). En las semanas 1 a 8 (por tanto, antes del proyecto) se cubren los siguientes temas:

- Tipos de datos elementales y vectores
- Esquemas algorítmicos básicos (recorrido y búsqueda)
- Sentencias condicionales e iterativas

La metodología que se utiliza en estas primeras semanas también se basa en la planificación de actividades y entregas de los estudiantes (en clase y fuera de clase) como por ejemplo ejercicios individuales, autoevaluaciones, ejercicios en grupo).

4.2. El proyecto

El proyecto, que se presenta en la sesión 9, consiste en la implementación de una aplicación que gestiona equipos de fútbol. Un equipo de fútbol tiene asociado varios datos (por ejemplo, nombre, lista de jugadores, etc.). Para cada jugador, tenemos también diferentes datos (por ejemplo, nombre, dorsal, goles marcados, etc.). La aplicación debe realizar varias funciones, mostrando un menú amigable al usuario para que pueda elegir la función que quiere realizar:

- Cargar la lista de equipos en una estructura de datos adecuada desde un fichero de texto
 - Salvar la lista de equipos en un fichero
 - Mostrar en pantalla una lista de equipos, con el nombre y el código de cada equipo
 - Mostrar todos los datos de un equipo a partir del código del equipo
 - Añadir un equipo
 - Dar de baja un equipo
 - Añadir un jugador a un equipo
 - Eliminar un jugador de un equipo
 - Buscar el equipo que haya ganado más partidos.
- La versión final debe incluir también algunas funciones optativas, a criterio del grupo. Algunos ejemplos de funcionalidades extras realizadas por algunos grupos en el pasado son:
- Buscar el jugador más joven

- Comprobar que los códigos son únicos (no existen equipos ni jugadores con el mismo código)
- Determinar qué jugador es el “pichichi” (ha marcado más goles)

El proyecto tiene un peso del 40% en la calificación de la asignatura, distribuido según el esquema siguiente:

- 10% primer prototipo (nota de grupo)
- 20% versión final (nota de grupo)
- 10% ejercicio individual sobre el proyecto (nota individual)

La versión final del proyecto se evaluará según los criterios siguientes:

- *Correcto* (4 puntos): Se han implementado todas las funciones obligatorias
- *Funciones adicionales* (1 punto): En función del número e interés de las funciones adicionales incorporadas a criterio del grupo.
- *Robusto* (1,5 puntos): La aplicación no se cuelga nunca
- *Amigable* (1,5 puntos): Es fácil interactuar con la aplicación
- *Organización* (2 puntos): El código está bien organizado y documentado.

4.3. El puzle

Los tres temas implicados en el puzle son: *Ficheros de texto* (los datos de los equipos están inicialmente en un fichero de texto), *Funciones* (el código debe estar bien estructurado) y *Estructuras de datos avanzadas* (necesarias para trabajar con una lista de equipos)

Cada uno de los temas tiene asociado un documento que presenta las ideas clave del tema y contiene ejercicios y tareas para hacer en casa con el propio ordenador.

La reunión de expertos se realiza en clase (sesión 10). Después de la fase de aclaración de dudas, cada grupo de expertos realiza en clase un ejercicio de profundización de su tema. Ejemplos de ejercicios para cada tema son los siguientes:

- *Ficheros*: Escribir un programa que lea de un fichero de texto los datos de equipos de fútbol (en cada línea del fichero están los datos de un equipo, es decir, nombre, código y año de creación, pero no los jugadores). El programa debe ir escribiendo en un fichero de salida los datos de los equipos cuyo año de creación es posterior a un año dado.

- *Estructuras*: Escribir un programa para buscar en una estructura de datos que implementa una lista de equipos, el equipo más antiguo.
- *Funciones*: Escribir un programa que tenga tres funciones. Una para leer de teclado los datos de un equipo de fútbol (nombre, código y año de creación), otra para escribir en pantalla los datos de un equipo, y una que nos diga si el equipo fue creado en año bisiesto.

4.4. Ejercicio de integración

El ejercicio de integración consiste en un programa que permita realizar las siguientes funciones (mediante un menú amigable):

- *Cargar equipos desde un fichero de texto (sólo nombre del equipo, año de creación y lista con los códigos de los jugadores)
- *Añadir un equipo
- *Buscar el equipo en el que está un jugador determinado (el código del jugador se lee del teclado).
- Dar de baja un equipo
- Añadir un jugador nuevo a un equipo
- *Salvar equipos en el fichero de texto.

En la segunda mitad de la sesión 11, el profesor explica en la pizarra el código de las funciones marcadas con * en la lista anterior. Antes de la siguiente sesión, cada grupo debe haber implementado el código que presentó el profesor en la pizarra y completarlo con las dos funciones que faltan (las que no tienen *). El ejercicio de integración debe estar funcionando correctamente durante la primera hora de la sesión 12.

4.5. Ampliaciones individuales y primer prototipo

Una vez entregado el ejercicio de integración, el profesor presenta las tres tareas individuales siguientes:

- Rehacer las operaciones de cargar y salvar datos en el fichero, teniendo en cuenta que ahora para cada jugador, además de su código, queremos tener su nombre y el número de goles que ha marcado.
- Rehacer las operaciones de añadir equipo, dar de baja equipo y añadir la operación para listar equipos, teniendo en cuenta la nueva información que tienen los jugadores.
- Rehacer las operaciones de añadir un jugador nuevo a un equipo y la función de buscar el equipo al que pertenece un jugador determinado,

teniendo en cuenta la nueva información que tienen ahora los jugadores.

El primer prototipo, que es el resultado de incorporar las tres ampliaciones al ejercicio de integración se presenta en la sesión 13. Se evalúa con los mismos criterios que la versión final, pero sin tener en cuenta el criterio de funciones adicionales y se asignan 5 puntos al criterio *Correcto*.

4.6. Versión final

Durante las dos semanas siguientes, cada grupo debe implementar la versión final, para lo cual debe:

1. Introducir en la aplicación las mejoras sugeridas por el profesor, a partir de su evaluación del primer prototipo.
2. Añadir el resto de funciones para satisfacer todos los requisitos obligatorios de la versión final.
3. Añadir las funciones optativas que el grupo estime oportuno

La versión final debe estar lista durante la primera hora de la sesión final (sesión 15). Los grupos tienen libertad para organizar y repartir el trabajo como quieran, pero debe entregar un plan de trabajo.

4.7. Conocimientos básicos

En la asignatura hay 8 conocimientos básicos que deben ser superados para aprobar el curso. Los que corresponden a la segunda mitad de la asignatura, durante la cual se desarrolla el proyecto, son los siguientes:

- Diseñar la estructura de datos más adecuada para representar la información de un problema dado.
- Codificar un procedimiento o función que cargue una estructura de datos con la información contenida en un fichero de texto o que salve en un fichero de texto la información contenida en la estructura.
- Escribir un procedimiento o función que recorra, busque, elimine o añada información en un vector de estructuras.
- Escribir un programa principal que realice llamadas a procedimientos y/o funciones pasándole adecuadamente los parámetros.

Hay dos oportunidades de superar estos conocimientos básicos: la sesión 14 y la sesión 16 (semana de exámenes finales).

En la sesión 15, después de entregar la versión final, cada grupo realiza una evaluación del proyecto de otros dos grupos. Además, cada estudiante de forma individual hace un ejercicio de modificación del proyecto para demostrar que domina el código realizado por el grupo. Un ejemplo de ejercicio individual puede ser: "Añadir a la información del equipo el presupuesto y recodificar el programa (o una parte) para que siga funcionando correctamente".

5. Conclusiones

Es bien sabido que en docencia no hay recetas que sirvan para todos los casos. No obstante, la receta que se propone en esta ponencia ayudó a muchos a introducirse de forma guiada en el complejo mundo del aprendizaje basado en proyectos. Aunque la propuesta ha tenido en cuenta los errores habituales, el éxito nunca está garantizado. En todo caso su puesta en práctica puede proporcionar una base sobre la que reflexionar y estimular su adaptación a cada contexto particular o su extensión a esquemas más ambiciosos que puedan explotar todo el potencial que tiene el aprendizaje basado en proyectos, especialmente en el contexto de la enseñanza de la ingeniería.

Referencias

- [1] Markham, T. Project Based Learning, a guide to Standard-focused project based learning for middle and high school teachers. Buck Institute for Education, 2003.
- [2] Estruch, V, y Silva, J., "Aprendizaje basado en proyectos en la carrera de Informática", JENUI 2006, p. 339.
- [3] Machado, S., Messeguer, R., Oller, A., Reyes, M.A., Rincón, D., Yúfera, J., "Recomendaciones para la implantación de PBL en créditos optativos basadas en la experiencia de la EPSC", JENUI 2005, p. 21.
- [4] Kjersdam, F., Enemark, S., *The Aalborg Experiment*, Aalborg University Press 1994. <http://adm.aau.dk/fak-tekn/aalborg/engel>

Telecolaboración interuniversitaria en prácticas de bases de datos

Arturo Jaime¹, Ana Sánchez², César Domínguez¹, José Miguel Blanco²

(1) Dpto. de Matemáticas y Computación de la Universidad de La Rioja. Ed. Vives, Luis de Ulloa s/n, 26004 Logroño
{arturo.jaime, cesar.dominguez}@unirioja.es

(2) Dpto. de Lenguajes y Sistemas Informáticos de la Universidad del País Vasco. Fac. Informática
P. Manuel de Lardizabal 1, 20018 Donostia-San Sebastián
{ana.sanchez, josemiguel.blanco}@ehu.es

Resumen

El área de bases de datos permite la aplicación de un modelo de aprendizaje basado en la actividad práctica. Dentro de esta área hemos desarrollado una experiencia entre dos universidades incorporando la telecolaboración de forma sistemática. Se han formado equipos con estudiantes que no se conocen entre sí y que han trabajado a distancia. El proyecto se ha desarrollado en el ámbito de dos asignaturas de bases de datos, una de cada universidad, donde las similitudes y diferencias entre ambas producen equipos heterogéneos. Los alumnos han colaborado en las diferentes fases de la creación de una base de datos, desde la invención de unos requisitos hasta la realización de consultas. Los resultados tanto académicos como de satisfacción de la experiencia son positivos.

1. Introducción

El área de bases de datos (BD) tiene recomendaciones bien definidas para los estudios de Ingeniería Informática [4] en cuanto al desarrollo de competencias: conocimiento y aplicación de las características, funcionalidades y estructura de las BD, que permitan su adecuado uso, y análisis, diseño e implementación de aplicaciones basadas en ellos. Ello no impide que existan diferencias en la asignación de contenidos a asignaturas y cursos en los planes de estudio.

Este trabajo describe una experiencia donde tratamos de aprovechar el contenido común y complementario de la primera asignatura de BD de la Universidad del País Vasco/Euskal Herriko Unibertsitatea, (UPV/EHU), y de La Rioja (UR), que coinciden en el mismo cuatrimestre, para organizar un trabajo colaborativo entre sus estudiantes. La actividad tiene como fin que los conocimientos impartidos queden mejor asentados

a través de la cooperación y la práctica, y también que las nociones y competencias más trabajadas por los alumnos de una universidad puedan ser útiles a los de la otra.

Cada equipo, formado por un estudiante de cada universidad, trabajará en la concepción, diseño, implementación y dos consultas sobre una BD. La distancia física existente entre los miembros de cada grupo (unos 150 km) fuerza a utilizar medios telemáticos para colaborar. Lo mismo sucede con los profesores participantes. Esto propicia el desarrollo de competencias transversales relacionadas con la telecolaboración.

Nuestra propuesta persigue tres objetivos. El primero es la mejora del rendimiento académico de los participantes (aumento del porcentaje de presentados, de aprobados e incremento de las notas del examen final de ambas asignaturas). El segundo es la adquisición de competencias transversales (capacidad de trabajo en grupo y colaboración a distancia). El tercero es potenciar el interés de los participantes hacia la asignatura.

El artículo se estructura de la siguiente forma: en la segunda sección se revisan algunos trabajos relacionados. En la sección 3 se sitúa la experiencia en el contexto donde se ha desarrollado. La sección 4 presenta las tareas realizadas por los alumnos de cada asignatura y cómo se organizaron en el tiempo. La sección 5 explica brevemente qué espacio de trabajo se utilizó y por qué. La sección 6 incluye lo más destacable del desarrollo de la experiencia. La sección 7 expone y analiza los principales resultados obtenidos. El trabajo finaliza con una sección de conclusiones.

2. Trabajos relacionados

El teletrabajo trata de flexibilizar el lugar y horario de trabajo. Se desarrolla generalmente desde casa y el contacto con clientes y

empleadores se realiza mediante ordenador, teléfono y centros de teletrabajo [13, 3]. Estudios recientes identifican una clara tendencia al crecimiento de esta forma de trabajo [13, 7].

El aprendizaje colaborativo puede definirse como el obtenido a través de la interacción de un grupo de personas. Estas comparten experiencias, tomando roles asimétricos. Tratan de resolver una tarea común en un marco de total interdependencia [5, 8, 11].

La telecolaboración se define [12] como una actividad educativa que implica a personas de lugares geográficos distantes para trabajar juntos mediante herramientas y recursos de Internet. Este término une las ideas de teletrabajo y trabajo colaborativo.

Existen numerosos estudios sobre trabajos prácticos en el área de BD. Muchos de ellos usan métodos de aprendizaje basado en proyectos [9, 6, 14, 17]. También se han propuesto experiencias docentes interuniversitarias incluyendo alguna de telecolaboración entre alumnos [10, 2]. En este artículo se presenta un método de trabajo que aúne ambas facetas y su aplicación experimental.

3. Contexto de la experiencia

Las asignaturas sobre las que se realiza la experiencia constan de seis créditos. Son las primeras asignaturas de BD en sus respectivos planes de estudios, pero están situadas en segundo curso en la UPV/EHU y en primero en la UR. Esta diferencia puede suponer que los alumnos de la UPV/EHU han adquirido algo más de madurez al haber cursado más materias. Las tareas a realizar no precisan conocimientos previos específicos.

Las asignaturas comparten algunas competencias comunes: conocimiento de conceptos básicos de BD, del modelo relacional y lenguaje SQL y capacidad de implementar y explotar BD mediante dicho lenguaje. También

tienen características distintivas: en la UR se estudia el álgebra relacional y se profundiza más con SQL, mientras que en la asignatura de la UPV/EHU se estudian los diseños conceptual y lógico. Cabe destacar que en la UR se trabaja con BD ya diseñadas. En las dos universidades hay otras asignaturas posteriores con las que se completan las competencias sobre BD.

Aunque las asignaturas coinciden en el mismo cuatrimestre, cada institución cuenta con su propio calendario, de modo que existen pequeñas diferencias que hay que tener en cuenta a la hora de planificar los plazos de realización de las tareas. El segundo cuatrimestre del curso 2009-10 empezó una semana más tarde y concluyó dos semanas después en la UR que en la UPV/EHU.

4. Tareas a realizar

El trabajo colaborativo definido cubre los aspectos de diseño y explotación de BD. Sobre la misma BD se ponen en práctica la mayoría de competencias asociadas a nuestras asignaturas.

El planteamiento inicial es que cada alumno se responsabilice de los aspectos en los que se le ha preparado mejor en su asignatura y apoye al compañero de la otra universidad en los temas que domina menos. Tal intercambio de conocimientos pretende un mejor aprendizaje.

La tabla 1 muestra la lista de tareas a realizar. Para cada una se incluye la distribución de responsabilidades y los plazos de realización.

El diagrama de Gantt de la figura 1 muestra la distribución de las tareas en el tiempo. El grupo de tareas de la parte central del diagrama corresponde al trabajo colaborativo. Los grupos de tareas de la parte superior e inferior reflejan el desarrollo de los temas en la UPV/EHU y UR respectivamente. Se pueden apreciar las precedencias entre el desarrollo de los temas, en ambas asignaturas, y los plazos de realización de las tareas del trabajo.

Tarea	Especialista	Revisor	Plazo realización
0. Análisis requisitos datos inicial	UR	no hay	1 semana
1. Análisis requisitos datos	UR	EHU	3 días
2. Diseño	EHU	UR	4 días
3. Implementación y carga	ambos	ambos	1 semana
4. Consulta 1	EHU	UR	1 semana
5. Consulta 2	UR	EHU	1 semana
6. Documento final	ambos	ambos	4 días

Tabla 1. Tareas del trabajo

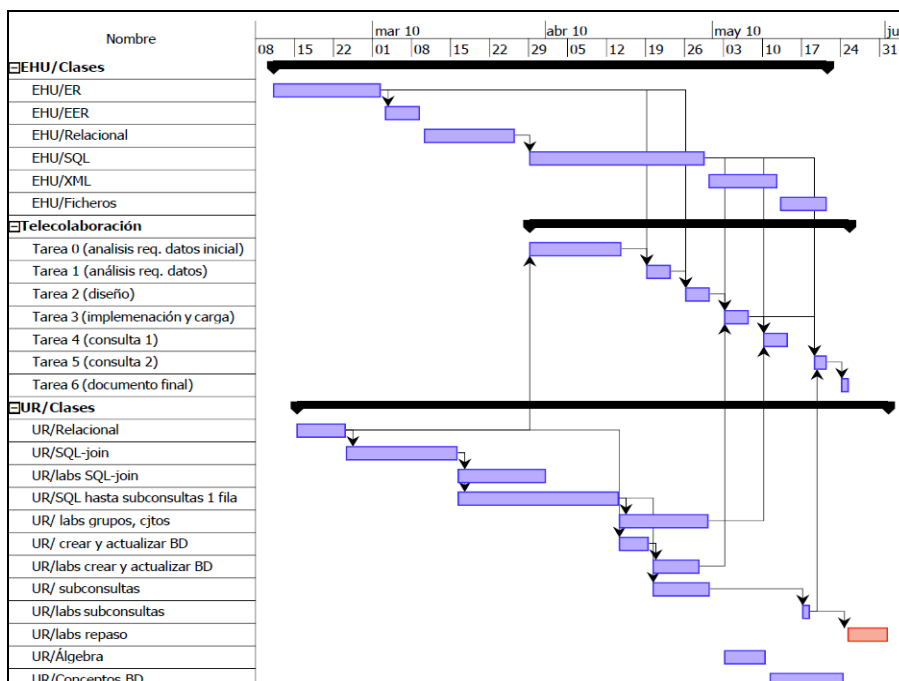


Figura 1. Diagrama de Gantt

Para realizar ejercicios de diseño conceptual se parte de un análisis de requisitos de datos. En ambas universidades se trabaja generalmente con casos artificiales descritos en lenguaje natural. Se suele obviar la realización de análisis de requisitos de datos por la dificultad de disponer de clientes, ficticios o no, con los que trabajar este aspecto.

Al estudiante de la UR se le asigna la tarea cero. Esta consiste, en primer término, en elegir un dominio. Una vez aceptado el dominio por los profesores, evitándose así que varios alumnos elijan el mismo, el estudiante debe idear requisitos de datos. Como deseamos equilibrar la complejidad de los esquemas realizados por los diferentes equipos, establecemos una serie de (meta) restricciones sobre el esquema de BD. Estas restricciones se presentan en términos del modelo relacional (el único conocido por estos estudiantes). Los profesores revisamos las descripciones de datos para garantizar que se cumplen las condiciones y que no se aportan detalles técnicos (por ejemplo, cómo ha de representarse determinada relación en el esquema

de BD o cuál de las posibles ha de ser la clave primaria de cierta tabla).

En la tarea 1 comienza la telecolaboración. El estudiante de la UPV/EHU debe entender los requisitos de datos proporcionados por su compañero de la UR. Para ello ha de actuar como diseñador de la BD y preguntar al estudiante de la UR aspectos que puedan faltar o que no estén recogidos claramente. Fruto de esta colaboración entregan un análisis de requisitos conjunto.

En la tarea 2 se realizan tanto el diseño conceptual como el lógico, presentándose un diagrama entidad-relación y un esquema relacional obtenido a partir de éste. El peso de esta tarea recae sobre el estudiante de la UPV/EHU. El estudiante de la UR debe entender primero los esquemas obtenidos, cuáles son las diferencias respecto a su idea inicial y, por último, validar los resultados o sugerir cambios. Tras este intercambio se entregará un resultado común.

En la tarea 3 hay que implementar la BD diseñada en la tarea anterior utilizando un SGBD común (MySQL). También se pide la inserción de algunas filas en cada tabla de la BD. Esto último

puede exigir cierto orden al insertar sobre las tablas (o deshabilitar restricciones y reactivarlas al final). Los estudiantes de ambas universidades deberían poder realizar estas actividades con similar nivel de competencia, por lo que ellos deciden cómo organizar esta tarea

Las tareas 4 y 5 consisten en idear un enunciado cuya resolución exija escribir una consulta SQL de una complejidad determinada. La consulta de la tarea 4 sería más simple y entraría en el nivel de competencia del alumno de la UPV/EHU. La consulta de la tarea 5, por su nivel de dificultad, está orientada al estudiante de la UR. El estudiante de la UPV/EHU debería entender en qué consiste el trabajo elaborado por su compañero y validarlo.

La tarea 6 consiste en la recopilación de las tareas anteriores y la elaboración de una breve memoria que recoja el trabajo completo.

El trabajo colaborativo que se acaba de describir se complementa con otros ejercicios a realizar en clases y laboratorios presenciales. También se han dedicado, en la UR, laboratorios a realizar algunas de estas tareas, como la implementación de la BD.

Cada tarea se ha presentado mediante una descripción de la actividad a realizar, de los contenidos y formato del entregable, del plazo de realización y de la identificación del alumno responsable de la tarea. En casi todos los casos se ha incluido un ejemplo de lo que se pedía.

Como en cualquier proyecto, la división en tareas y la planificación de entregas sirve para reducir la complejidad del trabajo y activar el inicio de su realización. En este caso hay un coste adicional en comunicación entre los integrantes del equipo, al tener que desarrollarse a distancia.

La asignación de responsabilidades en las tareas se ha hecho con objeto de activar el inicio de las tareas y orientar y repartir el peso de la comunicación. Como se muestra en la tabla 1, la mayoría de tareas tiene un responsable. El revisor de una tarea es también el encargado de su entrega.

La responsabilidad va alternándose entre los miembros del equipo según las competencias trabajadas en su asignatura. Hay dos excepciones donde los alumnos deciden cómo se organizan.

El punto de partida deseable en las propuestas de trabajos colaborativos es contar con estudiantes comprometidos, que se inicien en las técnicas necesarias para llevar a cabo cada tarea. Más aún

al tratarse de parejas de estudiantes que no se conocen y han de comunicarse fundamentalmente de forma escrita. Además cada estudiante ha de ser crítico en lo posible con el trabajo realizado por el otro y autocrítico con el propio. El hecho de escribir de forma sistemática las dudas a la hora de hacer una tarea o de ver cómo la ha resuelto el compañero, conlleva una mayor reflexión y mayor rigor tanto en comentarios como en resolución de tareas, por lo que, en hipótesis, ha de potenciar un mayor aprendizaje

Por último, hemos tratado de ubicar esta experiencia en la taxonomía de Bloom [1]. Es una de las clasificaciones de objetivos de aprendizaje más conocida, que también se ha usado en la enseñanza de BD [15, 16]. Las competencias a adquirir superarían los niveles bajos de la misma (como recordar y comprender conceptos) incidiendo en la aplicación de conocimientos adquiridos. Además hemos tratado de plantear actividades situadas en niveles altos como analizar, evaluar y crear nuevas actividades. Ejemplos de este tipo de actividades son la elaboración de un análisis de requisitos para una BD y la revisión crítica de las tareas realizadas por el compañero.

5. Espacio de trabajo

La naturaleza del trabajo exige abrir canales para la comunicación efectiva entre profesores y alumnos, y para establecer el contacto inicial entre los miembros de los equipos de trabajo.

Cada universidad dispone de un aula virtual institucional, accesible vía LDAP. Esto dificulta el acceso a alumnos y profesores que no pertenezcan a la universidad. Esta situación nos llevó a elegir una plataforma de uso abierto. Utilizamos un aula de moodle¹ desplegada en un servidor que permitía el acceso a los profesores y alumnos de ambas universidades. Llamamos al aula Ehureka (EHU-UR, Enseñanza-Aprendizaje).

Los alumnos participantes tuvieron que registrarse como usuarios del aula virtual, aportando un correo electrónico, su nombre y la ciudad donde estudiaban. A través de esta aula tenían acceso a la información de su compañero de equipo para poder iniciar el contacto. Los docentes participantes disponían de acceso al aula con rol de *profesor*.

¹ <http://lms.i2basque.es/course/>

En este espacio se fueron presentando los enunciados de las tareas conforme a lo descrito en la sección anterior. También sirvió como repositorio de entregables. Además de un foro general mediante el que se establecía la comunicación entre profesores y alumnos, se creó un foro para cada grupo, no visible por otros grupos. Su uso más frecuente fue la provisión de retroalimentación sobre los entregables.

Cada profesor mantuvo con sus estudiantes los canales habituales de comunicación: clase presencial, tutorías, correo electrónico o aula virtual institucional.

6. Desarrollo de la experiencia

Se presentó la experiencia a realizar en ambas asignaturas en la parte inicial del curso explicándose su fin experimental y voluntario.

Se interesaron por la actividad 23 alumnos de la UPV/EHU (46%) y 42 de la UR (69%). Se formaron 23 parejas para telecolaborar tomando un estudiante de la UPV/EHU y otro de la UR elegido aleatoriamente entre los disponibles.

El grupo de 19 alumnos de la UR sin emparejar realizó las mismas tareas que las realizadas como responsables por sus compañeros de la UR, según se explica en la sección 4, pero sobre un dominio de BD común. Trabajaron individualmente o formando equipo con otro estudiante de este mismo grupo. Los profesores proporcionaban y explicaban los diseños conceptual y lógico. Estos 19 alumnos han servido como grupo de control para un estudio comparativo de la experiencia.

Cinco de los equipos de telecolaboración resultaron *fallidos* (uno no llegó a establecer contacto y otros cuatro no llegaron a completar la tarea 1). Los alumnos de la UR que formaban parte de estos equipos fueron integrados en la experiencia que realizaba el grupo de control.

Participamos cuatro profesores en la experiencia, dos de cada universidad. Cada tarea tenía un responsable de la evaluación. Se mantuvo un contacto constante entre los profesores de ambas universidades y los criterios generales de corrección de las tareas eran conocidos por todos.

Los enunciados se presentaban con cierta antelación respecto al periodo de tiempo estimado para su realización. Junto a los enunciados, se publicaba el plazo de entrega y los enlaces para

depositar las tareas y realizar la declaración de tiempo dedicado a trabajo y comunicación.

El tiempo de dedicación a las tareas declarado por los alumnos de telecolaboración se aproxima a 10 horas de media (con desviación típica de más de 4.5 horas) y el tiempo asociado a la comunicación con el otro alumno superó las 6 horas (con desviación típica de más de 5 horas).

Hubo muchos retrasos en las entregas y nuestro objetivo de envío temprano de retroalimentación nos obligó a extender la actividad de corrección en el tiempo permaneciendo atentos a nuevas entregas. Estos retrasos no se tuvieron en cuenta en la evaluación.

La retroalimentación a los grupos exigió bastante disponibilidad de tiempo y se realizó, en general, de forma rápida (dos o tres días de plazo). Las tareas que más esfuerzo exigieron fueron las primeras. Por un lado, el análisis de requisitos de datos para controlar las condiciones establecidas sobre el mismo. Por otro lado, el diseño conceptual para controlar la coherencia entre los requisitos de datos y el esquema propuesto.

El reflejo de la experiencia en la calificación de la asignatura era diferente en cada universidad. En la UPV/EHU podía suponer un incremento de a lo sumo dos puntos, aunque ningún estudiante consiguió esta puntuación. En la UR formaba parte de las prácticas de la asignatura que incluían además la entrega de otros ejercicios realizados en clases prácticas. En total las prácticas constituían el 20% de la nota. Las notas del trabajo se publicaron junto a las del examen en ambos casos.

Desde un punto de vista puramente subjetivo detectamos mayor interacción entre profesores y alumnos que en cursos precedentes. Además de las dudas de tipo técnico, los alumnos también nos comentaban sus problemas en la comunicación con su compañero de equipo y su reflejo en los retrasos de las entregas

7. Resultados y discusión

Hemos considerado dos fuentes de datos. La primera son los resultados académicos, incluyendo el número de presentados, de aprobados y notas obtenidas en el examen final. También se cuenta con la asistencia de alumnos a laboratorios y la declaración del tiempo dedicado. La segunda fuente es una encuesta anónima, completada por los estudiantes tras la finalización

de la experiencia, que recoge su opinión sobre distintos aspectos de la actividad realizada.

Para estudiar los resultados académicos hemos distinguido varias categorías en cada universidad (ver tabla 2). En la UPV/EHU se distingue entre los estudiantes que participaron y los que no. En la UR se distingue entre los que trabajaron en un equipo de telecolaboración, el grupo de control con equipos formados por estudiantes solo de la UR y, por último, el resto de los alumnos. Los datos relativos a los grupos *fallidos* se presentan en fila aparte. La tabla 2 contiene también información sobre el número de alumnos de cada universidad, el porcentaje de presentados al examen de junio, la nota media sobre 10 puntos (y la desviación típica) y el porcentaje de aprobados

sobre presentados al examen, así como el porcentaje de asistencia a clases prácticas. Cabe destacar que los alumnos que hicieron el trabajo colaborativo a distancia en la UR obtienen mejores notas (aunque no sea estadísticamente significativo) que las de sus compañeros del grupo de control y por quienes no quisieron participar en la experiencia. Lo mismo sucedió en la UPV/EHU respecto a este último grupo. Los resultados académicos obtenidos por los alumnos de grupos fallidos son similares a los de aquellos que completaron el trabajo colaborativo a distancia de su universidad. Por lo tanto, no puede inducirse que el fracaso en la telecolaboración tenga una relación directa con los resultados académicos.

Universidad	Categoría	N (%)	% Presentados	Nota media (d.t.)	% Aprobados / presentados	% Asistencia a prácticas
UPV/EHU	<i>telecolabora</i>	18 (36)	88.9	5.28 (1.71)	43.8	
	<i>grupo fallido</i>	5 (10)	100	5.16 (2.35)	60	
	<i>no participa</i>	27 (54)	22.2	3.92 (1.88)	66.7	
UR	<i>telecolabora</i>	18 (29.5)	88.9	4.62 (2.40)	43.8	97.62
	<i>grupo fallido</i>	5 (8.2)	100	4.34 (2.42)	40	91.43
	<i>control</i>	19 (31.1)	84.2	3.47 (2.38)	25	94.36
	<i>no participa</i>	19 (31.1)	15.8	2.67 (1.32)	0	27.66

Tabla 2. Resultados académicos en los métodos a distancia (telecolabora) y presencial (control) por universidad

		Puntuación media (d.t.)	% bueno o excelente
La experiencia para aprender	A trabajar de forma colaborativa es	2.66 (0.71)	73.7
	A trabajar a distancia es	2.18 (0.80)	42.1
	A reforzar conocimientos de la asignatura es un medio	3.00 (0.70)	86.8
	A trabajar con estudiantes de otra universidad es una idea	2.78 (0.92)	62.1
Trabajo en equipo	Trabajo personal	3.08 (0.91)	73.7
	Trabajo del otro componente del equipo	2.41 (0.93)	48.6
	Valoración global del trabajo en equipo	2.57 (0.80)	59.5
Tareas	Distribución en el tiempo	2.32 (0.62)	34.6
	Claridad de los enunciados	2.79 (0.70)	63.2
	Dificultad	2.92 (0.59)	79
	El feedback (respuestas/correcciones) de los profesores	2.70 (0.88)	59.4
Evaluación	La forma de evaluar el trabajo (mediante corrección de tareas)	2.53 (0.61)	58.3
	Opinión sobre peso en la nota de la asignatura	2.22 (0.98)	43.2
Valoración del espacio de trabajo	Como herramienta de comunicación	2.39 (0.75)	39.5
	Como herramienta de presentación y recopilación de tareas	2.76 (0.67)	73.7
	Como herramienta de colaboración a distancia	2.50 (0.86)	52.6
Valoración global	La relación esfuerzo-aprendizaje ha sido	2.59 (0.72)	62.2
	Repetir una experiencia similar en otra asignatura es una idea	2.50 (0.92)	52.7
	Valoración global de la propuesta de trabajo	2.61 (0.75)	60.5

Tabla 3. Resultados de la encuesta

La encuesta constaba de una serie de preguntas que debían responderse en el rango de 1

a 4 representando a *malo*, *regular*, *bueno* o *excelente*. En dicha encuesta se incluyen

preguntas sobre la valoración de la experiencia para adquirir conocimientos en BD y también destrezas transversales, como trabajar en equipo, de forma colaborativa y a distancia. También se preguntaba sobre las propias características del trabajo a realizar y del espacio de trabajo (Ehureka). Por último se les solicitaba una valoración global de la experiencia y se ofrecían preguntas para indicar de forma abierta su opinión sobre la misma.

Se recogen en la Tabla 3 las preguntas realizadas y la puntuación media de las respuestas ofrecidas por los alumnos, así como el porcentaje de alumnos que respondieron *bueno* o *excelente*. Puede observarse que los alumnos tienen opiniones generalmente buenas (por encima de 2.5) en la mayoría de los ítems. Es destacable que más del 85% de los alumnos consideran que la experiencia es *bueno* o *excelente* para reforzar conocimientos de la asignatura y más del 73% la consideran *bueno* para trabajar de forma colaborativa. Más del 60% dan una valoración global positiva a la experiencia. Por otro lado, sólo el 42% la considera una *bueno* experiencia para aprender a trabajar a distancia, además, consideran que dentro del grupo ellos han trabajado más que sus compañeros (73.7% de valoración de trabajo personal, frente al 48.6% de sus compañeros) y que la distribución en el tiempo de las tareas no era del todo adecuada (sólo el 34.6% la valoran positivamente). Estos datos están en correspondencia con las opiniones ofrecidas a través de preguntas de respuesta abierta donde varios alumnos indicaban la dificultad para trabajar de forma sincrónica con sus compañeros y que los mensajes no eran respondidos satisfactoriamente en tiempo y forma.

Por último, hemos analizado si la experiencia ha tenido alguna influencia en la asignatura comparándola con lo sucedido en cursos académicos precedentes. Se ha observado una mejora en la opinión de los alumnos expresada a través de la encuesta institucional de los estudiantes hacia la asignatura mientras que los resultados académicos, cuyas medias son relativamente bajas, se mantienen estables. Creemos necesario un estudio a más largo plazo para poder obtener alguna conclusión sobre este aspecto.

8. Conclusiones

Hemos presentado una experiencia de trabajo colaborativo a distancia en el área de BD con equipos de trabajo heterogéneos formados por dos alumnos de dos universidades distintas.

Las tareas diseñadas han conseguido que los alumnos creasen elementos nuevos, enunciados y soluciones que siguieran unas pautas preestablecidas y validarlos sobre una BD implantada por ellos mismos. Cada equipo ha realizado un trabajo de las mismas características, pero diferente, ya que se ha elegido un dominio de BD particular en cada caso. La opinión de los alumnos sobre la experiencia como refuerzo de los conocimientos de las asignaturas es muy positiva.

La mejora del rendimiento académico de los participantes, que era nuestro primer objetivo, se ha alcanzado, si bien no de forma significativa. Aunque nos parece preciso repetir la experiencia durante más cursos para confirmar esta mejoría, pensamos que el hecho de que un estudiante tenga que colaborar con un compañero al que no conoce redundará en mejores resultados, ya que requiere un mayor interés y exige más implicación con la asignatura.

Los estudiantes han trabajado a distancia intercambiando conocimientos y realizando las tareas previstas. Por lo tanto, han puesto en práctica las competencias transversales propuestas como segundo objetivo. Además, según los datos de la encuesta, los estudiantes tienen una buena opinión de la experiencia de trabajo colaborativo. El planteamiento a distancia ha sido peor valorado. La razón puede estar relacionada con la necesidad de invertir tiempo adicional de trabajo para una buena coordinación con el compañero, por lo que creemos que su valoración mejoraría con la ampliación de los plazos y una modificación en la distribución de las tareas.

La disposición mostrada para participar en la experiencia y el número de presentados es acorde con el tercer objetivo, aumentar el interés por la asignatura. También se ha observado una mejora en las encuestas de satisfacción de las asignaturas al compararlas con los resultados obtenidos en cursos precedentes.

Para el profesorado participante la experiencia es positiva. Igual que los estudiantes, hemos realizado tareas adicionales de comunicación y coordinación. Trabajar con dominios distintos ha

dificultado la tarea de evaluación, sobre todo el seguimiento de las tareas de cada trabajo, pero ha asegurado el trabajo independiente de cada equipo. Es necesaria cierta ampliación de plazos para que los profesores podamos realizar una retroalimentación útil. Creemos que es aconsejable reorientar a algunos estudiantes hacia una BD bien diseñada para facilitar la continuación del trabajo sin arrastrar errores.

Por último, cabe señalar que las sinergias generadas mediante el trabajo colaborativo entre profesores de dos universidades constituyen un valor añadido importante y suponen un estímulo para seguir explorando esta línea de colaboración.

Agradecimientos

A las profesoras L. Muniozguen y C. Ocariz y a los alumnos participantes en la experiencia por su colaboración. A los revisores por sus valiosos comentarios, que ayudaron a mejorar la presentación de este trabajo. Al programa i2Basque que proporcionó la plataforma moodle. Este trabajo ha sido subvencionado parcialmente por la UR, proyecto API09/05.

Referencias

- [1] Anderson, L.W., Krathwohl, D.R. *A taxonomy for learning, teaching and assessing: A revision of Bloom's Taxonomy of educational objectives: Complete edition*. Longman 2001.
- [2] Bai, Y., Ma, K. *Interuniversity task-based cross-disciplinary collaborative learning model a case study of design and implementation of machine translation system*. Educational and Information Technology (ICEIT), 2010
- [3] Bermejo, M., Blanco, J.M., Fernández, C.R., Sánchez, A. *Simulación de teletrabajo con Software de Soporte a la Colaboración*. IX Internacional Symposium on Computers in Education-SIIE. 2007
- [4] BOE número 187 de 4/8/2009, páginas 66699 a 66710. *Acuerdo del consejo de universidades sobre recomendaciones para títulos oficiales de las Ingenierías Informática, Química y Técnica Informática*.
- [5] Chan, S.C.F., y otros. *Supporting real-time collaborative learning with web-based groupware*. Innovations in Education and Teaching International, 42(4), 2005.
- [6] Connolly, T.M., Begg, C.E. *A constructivist-based approach to teaching database analysis and design*. Journal of Information Systems Education, 17 (1), 2006
- [7] Cox, W. *Improving Quality of Life through Telecommuting*. The information Technology & innovation foundation. 2009
- [8] Dillenbourg, P. *Collaborative Learning: Cognitive and Computational Approaches*. *Advances in Learning and Instruction Series*. Elsevier Science, Inc. 1999.
- [9] Dominguez, C., Jaime, A. *Database design learning: A project-based approach organized through a course management system*. Computers & Education, 55(3), 2010.
- [10] Fainholc, B., Scagnoli, N. *Case Study: Blended Learning Through Interuniversity Collaborative Interaction*. 23rd conference on distance Teaching & Learning. 2007.
- [11] Francescato, D., y otros. *Evaluation of the efficacy of collaborative learning in face-to-face and computer-supported university contexts*. Computers in Human Behavior, 22, 2006.
- [12] Harris, J. *First steps in telecollaboration*. Learning & Leading with Technology, 27(3), 1999.
- [13] Lister, K., Harnish, T. *Undress for Success: The Naked Truth about Making Money at Home*. Wiley 2009.
- [14] Martínez, M., Duffing, G. *Teaching databases in compliance with the European dimension of higher education: Best practices for better competences*. Education Information Technology, 12, 211-228. 2007.
- [15] Mohtashami, M., Scher, J.M. *Application of Bloom's cognitive domain taxonomy to database design*. Information Systems Educators Conference 2000.
- [16] Neely, M.P. *Mastery level learning and the art of database design*. 30th Americas Conference on Information Systems 2007.
- [17] Soler, J., y otros. *Experiencia docente en diseño de bases de datos con ayuda de herramientas de e-learning*. JENU1 2009.

Sesión 3A:
Arquitectura de computadores
Sistemas operativos
Sistemas distribuidos y paralelos

El Abogado del Diablo como Técnica de Trabajo Cooperativo

Pablo Sánchez

Dpto. Matemáticas, Estadística y Computación
Universidad de Cantabria
Facultad de Ciencias, Avda. Los Castros S/N
39071 Santander (Cantabria)
p.sanchez@unican.es

Resumen

El Espacio Europeo de Educación Superior introduce la necesidad de formar a los alumnos en nuevas competencias profesionales les permitan afrontar su inserción en el mercado laboral con mejores garantías. Una de las competencias profesionales más demandadas por las empresas es la de trabajo cooperativo. Surge por tanto la necesidad de desarrollar metodologías para que los alumnos aprendan a trabajar de forma cooperativa eficazmente. Este artículo presenta una metodología de trabajo cooperativo, denominada *El abogado del diablo*, aplicable a las materias relacionadas con la programación de computadores. Un valor añadido de esta metodología es que forma sin esfuerzo adicional alguno a los alumnos en competencias relacionadas con la Ingeniería del Software.

Summary

Training students in new, and not strictly academic, professional skills has become practically mandatory with the emergence of the new European Space of Higher Education. The goal is to increase opportunities for employment of recently graduated students. One of the professional skills most demanded by companies is teamworking. Therefore, it becomes necessary to create new teaching methodologies which allow students to learn how to work as a team. This paper presents a methodology for teamworking, called "the devil's advocate", which is suitable for teaching subjects related to computer programming in the first courses of a Computer Science degree. Moreover, a positive side effect of our methodology is that students are also trained on Software Engineering skills without any additional effort.

Palabras clave

Trabajo Cooperativo, Competencias Profesionales, Programación, Metodologías Docentes

1. Introducción

El Espacio Europeo de Educación Superior, entre muchos otros objetivos, establece la necesidad de formar a los alumnos universitarios en una serie de competencias profesionales [7] que suelen ir más allá de las meramente académicas. Entre estas competencias se encuentran habilidades tales como la capacidad de iniciativa y de liderazgo, el aprendizaje autónomo o la capacidad de trabajar en equipo. Estas competencias plantean nuevos retos a los docentes universitarios, los cuales tienen en muchas ocasiones que formar a los alumnos en competencias en las cuales el propio docente no ha sido formado.

Surge por tanto la necesidad de crear nuevas metodologías docentes que asistan a los profesionales de la enseñanza en los procesos de formación en las nuevas competencias profesionales. Idealmente, estas nuevas metodologías deberían integrarse con las metodologías destinadas a la formación en competencias propias de cada titulación universitaria. Es decir, se trata de que el alumno adquiera las competencias profesionales a la vez que se adquiere las competencias propias de su titulación. La formación en competencias profesionales no sólo debería no interferir con la formación en competencias académicas, sino que debería favorecer la adquisición de estas últimas.

Una de las competencias profesionales más demandadas por la mayoría de los empleadores actualmente es la capacidad de trabajo en equipo. Esta competencia es particularmente difícil de trabajar en materias relacionadas con la Programación de Computadoras, sobre todo en los primeros cursos. El principal problema de desarrollar software en equipo es que, aparte de una sólida y eficaz dirección, nece-

sita de una infraestructura tecnológica, tales como repositorios de código o sistemas de control de versiones, que en muchas ocasiones no están fácilmente disponibles en el entorno universitario. Además, en los primeros cursos, dado el tamaño de las aplicaciones desarrolladas por los alumnos, resulta especialmente difícil dividir en desarrollo de una aplicación en partes o módulos independientes que permitan a los alumnos trabajar de forma eficaz como un equipo.

Este trabajo presenta una metodología basada en una técnica conocida de trabajo en equipo [5, 8] denominada *el abogado del diablo*, que resulta de especial utilidad para realizar trabajos prácticos de Programación en parejas. Dicha metodología se adecua a las necesidades de los primeros cursos de una titulación universitaria y tiene como ventajas adicionales:

- Ayuda a formar al alumno en otras competencias profesionales, como el aprendizaje autónomo;
- Predispone al alumno para el aprendizaje de competencias específicas relacionadas con la Ingeniería del Software, tales como la realización metodológica de pruebas o la gestión de versiones de un productos software;
- Mejora la percepción que el alumno tiene del trabajo cooperativo.

Dicha metodología ha sido aplicada con éxito en la asignatura de Estructura de Datos y Algoritmos perteneciente a la titulación de Ingeniería en Informática de la Universidad de Cantabria. El presente artículo describe dicha metodología así como los resultados obtenidos.

Tras esta introducción, este artículo se estructura de la siguiente forma: La sección 2 explica la motivación existente tras este trabajo. La sección 3 describe la metodología de trabajo en equipo basada en el abogado del diablo particularizada para materias de primeros cursos relacionadas con la Programación de Computadoras. La Sección 4 comenta los beneficios esperados, mientras que la Sección 5 discute los resultados obtenidos tras aplicar esta metodología. Finalmente, la Sección 6 resume las conclusiones principales extraídas de este trabajo.

2. Motivación

El principal problema que presenta la realización de trabajos en equipo en materias relacionadas con la Programación de Computadoras en los primeros cursos de una titulación relacionada con la Ingeniería en Informática es que el tamaño de las aplicaciones software desarrolladas por los alumnos no alcanza el mínimo necesario para que la aplicación pueda ser fácilmente dividida en partes o módulos independientes.

El principal inconveniente es que la metodología de trabajo cooperativo normalmente utilizada por los alumnos se suele basar en varios estudiantes sentados alrededor de una computadora observando como sólo uno de ellos programa. El resto, en el mejor de los casos, se limita a: (1) verificar que el que está programando no comete errores; (2) hacer sugerencias; y (3) proponer mejoras. En esta escenario, el hecho de trabajar en equipo se considera estéril e improductivo, y así lo perciben muchos alumnos.

Aunque esta situación pueda tener cierto parecido con técnicas de desarrollo software modernas, tales como la *programación por pares* [1], entendemos que las habilidades y capacidades de los estudiantes de los primeros cursos no están lo suficiente maduras como para poder sacar beneficio de tales técnicas. Por ejemplo, en el caso de la programación por pares, mientras un programador está codificando, el otro supervisa el proceso y propone mejoras en aras de una mayor facilidad de adaptación y evolución del producto software. Para que esta técnica funcione de forma efectiva, se presupone que ambos programadores son programadores experimentados y tienen amplio conocimiento de técnicas avanzadas de diseño software, como un sólido conocimiento sobre Patrones de Diseño [3]. Obviamente, las competencias adquiridas por los alumnos al comienzo de un segundo curso de una titulación como Ingeniería en Informática no son suficientes para alcanzar este nivel.

Cabe destacar que si pretendemos enseñar a nuestros alumnos a trabajar en equipo, el primer paso es hacer que comprendan el valor del trabajo cooperativo. Según los datos recopilados entre el alumnado antes de aplicar nuestra metodología, la mayoría de los alumnos consideraban que el hecho de trabajar en equipo no les aportaba ningún beneficio. Argumentaban que podían concluir las tareas asignadas

con mayor rapidez si las realizaban de forma individual, en lugar de realizarlas en equipo. Además, en el caso de las tareas a realizar fuera del aula, el hecho de trabajar en equipo les obligaba a reunirse fuera del periodo de docencia presencial, con el consiguiente esfuerzo de coordinación. Todo esto aumenta el tiempo dedicado al trabajo en equipo y su descrédito. Por tanto, la realidad actual era que los estudiantes percibían el trabajo en equipo como algo que les perjudicaba más que beneficiarles. Si los alumnos no perciben las ventajas de trabajar de forma cooperativa, difícilmente podrán aprender a trabajar cooperativamente. Se hace por tanto patente la necesidad de crear metodologías que permitan a los alumnos valorar el trabajo cooperativo. Es decir, se trata de crear técnicas de trabajo en equipo que incrementen la productividad en lugar de decrementarla.

La siguiente sección describe la metodología del abogado del diablo aplicada al desarrollo de pequeñas aplicaciones software por parejas de estudiantes. Una de las ventajas de esta metodología es que permite a los alumnos trabajar de forma individualizada cooperativamente. Se benefician del hecho de pertenecer a un grupo sin que ello les suponga un gran esfuerzo adicional, viendo aumentada su productividad.

3. Descripción de la Metodología

La figura del *abogado del diablo* tiene un origen eclesiástico. Su función es la de verificar si un hecho considerado como milagroso es realmente un milagro o, por el contrario, es un hecho extraordinario pero basado en una explicación perfectamente racional. En este segundo caso, se descarta la intervención divina y no se le atribuye carácter milagroso. La persona que desempeña esta tarea, por tanto, parece empeñada en negar la existencia de Dios, de ahí que se le considere un mensajero o un abogado del diablo.

El papel de abogado del diablo se introdujo en las técnicas de trabajo en equipo hace ya varias décadas [8, 1]. Fue empleada incluso por el presidente americano John F. Kennedy. La idea de Kennedy era la de disponer de un asesor que criticara enérgicamente las decisiones de su gabinete. Si Kennedy lograba rebatir con contundencia las críticas de este asesor, las decisiones se consideraban acertadas. Además, el presidente adquiriría la destreza necesaria

para rebatir tales críticas si éstas eran planteadas por algún miembro de la oposición o de la prensa.

La metodología que proponemos en este artículo está basada en crear parejas de estudiantes en las cuales uno realiza una primera parte del trabajo asignado, y el otro ejerce el papel de abogado del diablo. El abogado del diablo evalúa a su propio compañero de forma exhaustiva. A continuación, los alumnos intercambian los papeles. De esta forma, el que ha ejercido de abogado del diablo continúa con el trabajo asignado; y el que había realizado el trabajo, pasa a ejercer el papel de abogado del diablo.

A continuación, exponemos como se ha aplicado esta metodología a la implementación de un Tipo Abstracto de Datos dentro de la asignatura de Estructuras de Datos y Algoritmos de la titulación de Ingeniería Informática. La asignatura pertenecía al segundo curso y tenía un total de 42 alumnos.

La tarea a realizar por los alumnos era implementar, en un lenguaje que nunca hubiesen utilizado anteriormente y de su elección, un Tipo Abstracto de Datos (por ejemplo, un conjunto implementado sobre listas enlazadas). El objetivo último del trabajo era fomentar el aprendizaje autónomo de nuevos lenguajes de programación, competencia fundamental para todo Ingeniero Informático. Dicha tarea se debía realizar por parejas, que podían ser libremente formadas por los alumnos de acuerdo a sus propias afinidades. Cada grupo debía elegir, según sus capacidades y expectativas, un Tipo Abstracto de Datos y una técnica de implementación (por ejemplo, conjuntos sobre tablas de dispersión).

Una vez elegido el Tipo Abstracto de Datos a implementar, la metodología de trabajo propone como siguiente etapa dividir la interfaz del mismo en grupos de operaciones relativamente independientes que se puedan implementar de forma secuencial. Lo mismo se aplica a la implementación del soporte para dicho Tipo Abstracto de Datos. Por ejemplo, suponemos que una pareja de alumnos escogen implementar un caso de conjunto implementado sobre tablas de dispersión. Habría que implementar primero una tabla de dispersión y a continuación, usando dicha tabla de dispersión, implementar un conjunto.

Las operaciones a implementar para este TAD podrían dividirse en los siguientes bloques:

1. Implementación de las operaciones de inserción, búsqueda y eliminación en la tabla de dis-

130 Sesión 3A: Arquitectura de computadores / Sistemas operativos / Sistemas distribuidos y paralelos

- persión.
2. Implementación de las operación de creación, inserción, eliminación y búsqueda en un conjunto (implementado sobre una tabla de dispersión).
 3. Implementación de las operaciones de unión, intersección y diferencia entre conjuntos.
 4. Implementación de las operaciones cardinal, subconjunto e igualdad entre conjuntos.

Una vez divididas estas operaciones en bloque, la idea principal de la metodología es que un estudiante implemente sin la presencia de su compañero. Por tanto, no se requieren esfuerzos de coordinación. Su compañero va a ejercer el papel de abogado del diablo sobre la implementación creada. El fundamento y motivación para el alumnado reside en que resulta mejor que los fallos cometidos por un compañero los descubra otro compañero antes de que el trabajo sea calificado por el docente, momento en el cual dichos fallos ya no serían reparables.

Es decir, la metodología de trabajo se basa en que un alumno implementa y el otro prueba y *certifica* el programa. Usamos el verbo certificar porque las tareas del abogado del diablo van más allá de la realización de pruebas exhaustivas.

Concretamente, las tareas del abogado del diablo son verificar que:

1. El programa funciona para todos los casos posibles. Ello implica que el abogado del diablo debe diseñar un conjunto exhaustivo de pruebas y ejecutarlas, con objeto de comprobar que el código creado por su compañero funciona para todos los casos posibles.
2. La documentación de los métodos es adecuada, es decir, es suficiente, precisa y concisa.
3. Las precondiciones para los métodos declarados son correctas y suficientes y que siempre que se satisfacen ningún método lanza una excepción (se sigue en la asignatura un proceso de aprendizaje basado en *Diseño por Contrato* [6]).
4. Siempre que se invoca un método se satisfacen las precondiciones, (de nuevo de acuerdo con los principios del *Diseño por Contrato*).
5. En los comentarios y documentación de los métodos no existen errores tipográficos u ortográficos.
6. No se incumplen las normas de estilo de pro-

gramación contenidas en la *Guía de Buenas Prácticas en Programación* proporcionada al comienzo de la asignatura.¹

7. El código creado se entiende bien, ya sea porque es trivial o porque su funcionalidad se deduce fácilmente de la documentación de los métodos o de los comentarios insertados en el código.
8. La información que se muestra al usuario por pantalla es clara, precisa, concisa y está libre de errores tipográficos u ortográficos.

Es decir, el abogado del diablo trabaja con una *rúbrica* idéntica a la que usaría el profesor para calificar dicho trabajo. De esta forma, el trabajo producido por un estudiante es precalificado por su compañero de trabajo. Si éste descubriese fallos, éstos se podrían corregir antes de que fuese demasiado tarde, es decir, antes de que el trabajo fuese definitivamente calificado por el profesor.

En caso de detección de fallos, el propio abogado del diablo podría optar por corregirlos él, o solicitar a su compañero que lleve a cabo las acciones correctivas pertinentes.

Una vez que el trabajo realizado recibe la aprobación del abogado del diablo, se pasaría a la siguiente iteración en la realización del trabajo. En la siguiente iteración, los estudiantes se intercambian los papeles, siendo el abogado del diablo el que implementa el siguiente bloque de código; y el que implementó anteriormente pasa a ejercer ahora el papel de abogado del diablo.

Destacar que si el número de bloques en los que se ha dividido el trabajo fuese impar no habría problema alguno dado que la carga de trabajo por bloque asignada al que implementa y al abogado del diablo es aproximadamente la misma. Es decir, aunque al principio a los estudiantes le pueda parecer lo contrario, el abogado del diablo trabaja igual, o algo más incluso, que el que implementa, siendo por tanto la distribución del trabajo equitativa.

Para que esta metodología tenga éxito, el abogado del diablo debe criticar, de forma racional y constructiva, pero con firmeza y exhaustividad, el trabajo de su compañero. El compañero debe tomarse estas críticas no como un ataque personal, sino como un trabajo cuyo único objetivo es mejorar el resul-

¹ Disponible en <http://personales.unican.es/sanchezbp/teaching/faqs/programming.html>

tado global del equipo. Si el compañero que es inspeccionado por el abogado del diablo detectase que después del trabajo del abogado quedasen aún existiesen defectos que deban ser reparados, deberá instar al abogado a repetir su trabajo, pero esta vez con mayor diligencia y profesionalidad.

La siguiente sección describe los beneficios obtenidos al aplicar esta metodología.

4. Beneficios Esperados

Los beneficios derivados de la aplicación de la metodología del abogado del diablo se resumen en los siguientes puntos, los cuales desarrollaremos a lo largo de esta sección:

1. Los alumnos pueden trabajar de forma individual a la vez que realizan un trabajo cooperativo.
2. Permite la colaboración no presencial entre alumnos.
3. Los estudiantes aprenden de los errores de sus compañeros, es decir, se produce en parte un *aprendizaje por observación y experimentación*.
4. Fomenta el espíritu crítico del alumno, que aprende a autoevaluarse, competencia necesaria para alcanzar el aprendizaje autónomo.
5. Los alumnos, mediante la experimentación, descubren la existencia de ciertos problemas relacionados con la Ingeniería del Software, tales como el control de versiones o el diseño de pruebas.
6. La percepción y valoración del estudiante del trabajo cooperativo mejora.

4.1. Trabajo individual y cooperativo

De acuerdo con los principios de nuestra metodología, los alumnos trabajan de forma individual pero coordinada. Por su propia construcción, la metodología evita el que haya dos alumnos sentados frente a un mismo computador, uno trabajando y el otro mirando. En la metodología del abogado del diablo, ambos estudiantes se convierten en elementos productivos del grupo.

Mencionar que esta forma de trabajar en equipo es la que se corresponde con la práctica diaria de las empresas de desarrollo software. En tales empresas,

un director de proyecto coordina el trabajo de varios desarrolladores. Estos desarrolladores trabajan de forma individual sin que exista excesiva comunicación entre ellos. Es decir, se trata de un trabajo cooperativo pero autónomo.

Podría también plantearse como solución para convertir a los dos estudiantes en elementos productivos que, tras dividir el código a desarrollar en bloques, cada alumno implementase un bloque de forma individual y luego los integrasen. Debido a las dependencias normalmente existentes entre diferentes bloques, este desarrollo independiente se hace francamente complejo. Incluso en los casos donde es posible alcanzar cierto grado de independencia, los esfuerzos requeridos para integrar el código desarrollado de forma aislada por los estudiantes suelen ser bastante grandes y tediosos. Por tanto, el estudiante vuelve a tener la sensación de que podría haber completado el trabajo en menor tiempo si hubiese realizado la tarea de manera individual en lugar de colectiva.

Por tanto, consideramos que la metodología del abogado del diablo ayuda a mejorar percepción que poseen los alumnos sobre los beneficios del trabajo en equipo.

4.2. Coordinación no necesariamente presencial

Como consecuencia directa de que la metodología obligue a que los alumnos trabajen de forma individual pero coordinada, se elimina la necesidad de que los alumnos tengan que reunirse en un momento concreto en un espacio físico determinado para realizar el trabajo asignado.

Los alumnos sólo necesitan reunirse para decidir el tema del trabajo (si se les da la opción de elegir), dividir el trabajo en diferentes bloques y decidir quién implementa y quién es abogado del diablo en la primera iteración. Es conveniente también que los alumnos acuerden un calendario de trabajo, así como que se comprometan a cumplirlo. Tras esta reunión inicial, y dado los avances tecnológicos existentes hoy en día, los alumnos no tienen necesidad alguna de volver a reunirse.

En cada iteración, el alumno que implementa sólo necesita enviar el código creado al abogado del diablo una vez que ha terminado su trabajo. El abogado del diablo elabora entonces su informe de fallos,

corrige aquellos que considere triviales, y le vuelve a reenviar el código, si lo hubiese modificado, junto con el informe de errores a subsanar. El alumno que creó el código subsana los errores notificados y vuelve a enviar el código al abogado del diablo. Este último repite el proceso hasta que ambos estudiantes den el visto bueno al trabajo.

Por tanto, tener acceso al correo electrónico es suficiente para poder trabajar de forma cooperativa pero individualizada. Ello evita el que los alumnos se tengan que reunir para trabajar. Este hecho, según las opiniones, suele ser el primer obstáculo que los alumnos encuentran para trabajar en equipo.

4.3. Aprendizaje mediante la evaluación por pares

Se dice que el hombre es el único animal capaz de tropezar dos veces con la misma piedra. A pesar de ello, es conveniente permitir que los alumnos tropiecen. De esta forma, tras varios tropiezos, acaban por darse cuenta de dónde están las piedras y aprenden a evitarlas.

La metodología del abogado del diablo fomenta que los alumnos descubran dichas piedras y aprendan a evitarlas. Usando esta metodología, los alumnos descubren dónde fallan las aplicaciones de sus compañeros y cómo se solucionan esos fallos. A partir de ese momento, toman conciencia de la existencia de tales fallos y comienzan a adoptar las actitudes necesarias para evitarlos.

Un ejemplo muy claro es la operación de eliminación en listas enlazadas con punteros al principio y al final de dicha lista. La mayoría de los alumnos prueban el código desarrollado para listas vacías y listas con varios elementos, pero casi nunca para listas con un único elemento, donde dicho elemento sea además el elemento que hay que eliminar.

La metodología del abogado del diablo ayuda a que los alumnos descubran estos casos particulares. Esto es debido en gran parte al hecho de que el abogado del diablo sólo tiene que diseñar los casos de prueba, lo cual ayuda a que el alumno que desempeña el papel del abogado del diablo sea más exhaustivo diseñando casos de prueba, dado que es su única función y no se encuentra agotado o cansado por el esfuerzo de la implementación. Este cansancio o hastío es la razón principal que esgrimen los alumnos para no diseñar casos de prueba exhaustivos.

4.4. Aprender a autoevaluarse

El hecho de ejercer de abogado del diablo convierte temporalmente al alumno que desempeña dicho papel en evaluador de su compañero. Ello hace que para la siguiente iteración, el abogado del diablo asuma que va a ser evaluado de la misma forma que él ha sido evaluado. Por tanto, desarrolla el código evitando cometer los mismos errores que él ha detectado en el trabajo de su compañero. Es decir, el alumno aprende a autoevaluarse, a detectar sus propios errores y a corregirlos. Por tanto mejora sus habilidades y competencias de manera autónoma. Es decir, se fomenta el *aprendizaje autónomo* y el alumno aprende a aprender.

4.5. Competencias relacionadas con la Ingeniería del Software

La aplicación de la metodología del abogado del diablo prepara al alumno para una mejor adquisición de competencias tradicionalmente relacionadas con la Ingeniería del Software, materia que suele impartirse en un tercer o cuarto curso. Esto se debe a que el alumno experimenta y sufre mediante la aplicación de la metodología presentada algunos de los problemas que la Ingeniería del Software trata de solucionar. Una mejor comprensión del problema, y sobre todo el haberlo sufrido, permite al alumno entender mejor la necesidad de aprender y comprender las soluciones existentes a esos problemas. Es decir, el haber experimentado ciertos problemas y dificultades motiva al alumno a aprender y comprender cómo solucionarlos.

Dos claros ejemplos son la *gestión de versiones* [2] y la necesidad de disponer de *metodologías para la realización de pruebas sistemáticas* [9].

Uno de los primeros problemas con los que se enfrenta el abogado del diablo es el diseño de casos de prueba sistemáticos. Sin ayuda de una metodología adecuada, el abogado del diablo se encuentra la mayoría de las veces desamparado. Por tanto, el alumno toma conciencia de la existencia una rama de conocimiento dedicada al desarrollo sistemático de pruebas software. Consecuentemente, el alumno no sólo entiende la necesidad de su existencia, sino que además se encuentra motivado hacia su aprendizaje, al ser plenamente consciente de su utilidad práctica.

El mismo fenómeno acontece con la gestión de versiones. Al tener que compartir y modificar el mismo código fuente la pareja de estudiantes, aparecerán los inevitables problemas de versiones. Es por ello que el alumno, al sufrir y experimentar el problema, entenderá y valorará mejor la necesidad de disponer de metodologías para la gestión del cambio y la configuración, así como la necesidad de disponer de herramientas como *Subversion* que sirvan para automatizar parte de estas tareas.

4.6. Valoración del trabajo en equipo

Al ser los dos miembros del grupo elementos productivos con escasa necesidad de coordinación, los alumnos perciben que el hecho de trabajar en equipo aumenta la productividad. Como consecuencia, los alumnos valoran los resultados obtenidos como de mayor calidad que los que podrían haber obtenido trabajando a nivel individual, y por tanto descubren las ventajas de trabajar en equipo.

La siguiente sección analiza las argumentaciones de esta sección a partir de los resultados obtenidos, las opiniones recogidas entre el alumnado y una comparación con los cinco elementos clave que debe poseer todo aprendizaje cooperativo.

5. Discusión de los Resultados Obtenidos

Esta sección discute los resultados obtenidos a partir de las opiniones y sensaciones recopiladas entre el alumnado acerca de su percepción de trabajo cooperativo antes y después de aplicar la metodología propuesta en este artículo. Se analiza también cuándo la metodología propuesta satisface los 5 elementos clave del aprendizaje cooperativo [4].

En general, la mayoría de los alumnos comentaban al principio del cuatrimestre que el trabajo en equipo era a veces positivo, pues en ocasiones podían aprender algo de sus compañeros. Pero, por otra parte, les suponía normalmente un esfuerzo extra considerable debido a la dificultad de encontrar un horario y espacio determinados donde poder reunirse para trabajar juntos. Además, algunos alumnos, normalmente los que mejores calificaciones tenían, eran de la opinión de que el trabajo en equipo les perjudicaba más que les beneficiaba. Estos alumnos consideraban que podían llegar a los mismos resultados y de forma más rápida trabajando de forma in-

dividual que cooperativamente.

Al final del cuatrimestre las opiniones del alumnado acerca de la metodología del abogado del diablo eran mayoritariamente positivas. En especial, porque les permitía trabajar de forma coordinada, pero sin estar físicamente juntos ni trabajando al mismo tiempo. La percepción de los alumnos acerca del trabajo en equipo parecía haber mejorado notoriamente al final del cuatrimestre. Casi la totalidad de ellos, en mayor o menor grado, eran de la opinión de que el hecho de trabajar en equipo les había ayudado a obtener resultados de mejor calidad que los que hubiesen obtenido trabajando individualmente, incluyendo a los alumnos con mejores calificaciones. Este tipo de alumno comentaba que la metodología les ayudaba a mejorar su trabajo, dado que sus compañeros detectaban errores que a ellos se les habían inicialmente escapado. Por otra parte, cuando ejercían de abogados del diablo se sentían mucho más cómodos a la hora de criticar, siempre de forma constructiva, el trabajo de sus compañeros. Comentaban que, estas mismas críticas, cuando se hacían usando otra metodología, no eran siempre recibidas con agrado por el resto del equipo, mientras que ahora, las mismas críticas, si eran bien recibidas.

Como aspectos negativos de la metodología, la mayoría de los alumnos comentaban que les resultaba difícil diseñar casos de prueba exhaustivos, principalmente porque no sabían que pasos debían seguir, qué casos considerar, o cuántos casos eran suficientes. Por tanto, nos proponemos como mejora a esta metodología formar a los alumnos de los próximos cursos en el diseño sistemático de casos de prueba. De este modo el alumno que ejerza el papel de abogado del diablo, podrá hacerlo de manera más efectiva, sistemática y científica.

Por último, analizamos como la metodología del abogado del diablo cumple con los 5 aspectos clave del trabajo cooperativo: (1) interdependencia positiva; (2) interacción positiva cara a cara; (3) exigibilidad individual; (4) habilidades cooperativas; y (5) autoanálisis del grupo.

Interdependencia positiva A tenor de las opiniones recogidas entre el alumnado, el alumno entiende que el hecho de pertenecer a un equipo le beneficia, por lo que considera el hecho de trabajar de forma cooperativa como algo positivo. El fomentar la interdependencia positiva era uno de los objetivos prin-

cipales que perseguíamos con la introducción de esta metodología.

Interacción positiva personal En este aspecto la metodología del abogado del diablo ayuda a mejorar la percepción que los alumnos tienen de sus compañeros. Obsérvese por ejemplo que los alumnos con mejores calificaciones mejoraron la percepción que tenían de sus compañeros menos capaces. De la misma forma, los menos capaces aprendieron a aceptar las críticas de los alumnos más aventajados.

Exigibilidad individual Tal como planteamos la metodología, cada alumno tiene una tarea bien definida y ha de responder con su trabajo ante su compañero. Si alguno de los dos miembros de un grupo no trabaja, queda patente y manifiesto quién es el que no trabaja. No obstante, la metodología no incluye ningún punto específico acerca de cómo resolver los conflictos, en caso de que se produzcan, o de cómo evitar que se produzca ningún tipo de *parasitismo*.

Habilidades cooperativas La metodología del abogado del diablo es en sí una técnica de trabajo cooperativo que los alumnos aprenden mediante su aplicación. Por tanto, mejoran su competencia de trabajo cooperativo mediante la incorporación a su saber y habilidades de una nueva técnica de trabajo cooperativo, en mi humilde opinión, de gran utilidad en el ámbito del desarrollo software. La metodología además no es incompatible con la formación en otras habilidades cooperativas, como la toma de decisiones o la gestión de conflictos.

Autoanálisis del grupo En nuestro caso particular, no ha existido ningún tipo de autoanálisis de los grupos, a excepción de la propia de final del cuatrimestre, con objeto de conocer las opiniones del alumnado acerca de la técnica aplicada. No obstante, la técnica no es incompatible con el autoanálisis. Es más, al estar las funciones de cada miembro del grupo claramente identificadas, es más fácil comprobar si cada uno está cumpliendo su función y si dicha función es eficaz para alcanzar un objetivo determinado.

6. Conclusiones

El presente artículo ha presentado una metodología de trabajo en equipo que resulta especialmente adecuada para las materias relacionadas con la Programación de Computadoras que se imparten en los primeros cursos de las titulaciones universitarias. Las principales ventajas de esta metodología es que mejora la visión que los alumnos tienen del trabajo cooperativo, al servir para incrementar su productividad y desarrollar productos de mayor calidad.

Referencias

- [1] Kent Beck. *Extreme Programming Explained: Embrace Change*. Addison-Wesley Professional, October 1999.
- [2] Stephen P. Berczuk. *Software Configuration Management Patterns: Effective Teamwork, Practical Integration*. Addison-Wesley Professional, November 2003.
- [3] Erich Gamma, Ralph Johnson Richard Helm and, and John M. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, November 1994.
- [4] David W. Johnson, Roger T. Johnson, and Karl A. Smith. *Active Learning: Cooperation in the College Classroom*. Interaction Book, December 1991.
- [5] Thomas Kelley and Jonathan Littman. *The Ten Faces of Innovation: IDEO's Strategies for Defeating the Devil's Advocate and Driving Creativity Throughout Your Organization*. Doubleday, Diciembre 2005.
- [6] Bertrand Meyer. *Touch of Class: Learning to Program Well with Objects and Contracts*. Springer, 2009.
- [7] Mercedes Marzo Navarro, Marta Pedraja Iglesias, and Pilar Rivera Torres. Las competencias profesionales demandadas por las empresas: el caso de los ingenieros. *Revista de Educación*, 341:643–661, Septiembre-Diciembre 2006.
- [8] Michael A. West. *Effective Teamwork: Practical Lessons from Organizational Research*. Wiley-Blackwell, December 2003.
- [9] Colin Willcock, Thomas Deiss, Stephan Tobias, Stefan Keil, Federico Engler, and Stephan Schulz. *An Introduction to TTCN-3*. Wiley, July 2005.

Entrenamiento de la creatividad y la innovación en la Ingeniería de Computadores basándose en la metodología de aprendizaje por proyectos.

Gabriel Jiménez Moreno, José Luis Sevillano Ramos, Angel Jiménez Fernández, Rafael Paz Vicente, Alejandro Linares Barranco, Manuel Domínguez Morales, Elena Cerezuela Escudero, Lourdes Miró Amarante
Dpto. Arquitectura y Tecnología de Computadores
Universidad de Sevilla
Escuela Técnica Superior de Ing. Informática, Avda. Reina Mercedes, s/n
41012 Sevilla
gaji@atc.us.es

Resumen

En los estudios de Ingeniería normalmente se instruye en competencias sobre diseño, en las que es fundamental conocer las tecnologías y herramientas correspondientes a la materia en que el alumno se especializa. Los problemas y proyectos que el alumno resuelve a lo largo de su carrera suelen estar orientados por el profesor, de manera que la idea básica forma parte de la propuesta o enunciado. El alumno no hace más que aplicar la tecnología para resolver el problema que ha ideado otro, en raras ocasiones se plantean problemas o proyectos abiertos que requieran aplicar la creatividad. En este trabajo se muestra una experiencia realizada durante el curso 2009/10, y financiada por el Plan de Renovación de Metodologías Docentes Universidad Sevilla, en la Escuela Técnica Superior de Ingeniería Informática (ETSII), en la que los alumnos debían hacer propuestas de proyectos a resolver con las herramientas propias de las asignaturas implicadas.

Summary

In engineering studies typically instructed in responsibilities for design, which is essential to know the technologies and tools for the field in which the student specializes. Problems and projects that the student meets along his career tend to be guided by the teacher, so the basic idea is part of the proposal or statement, the student merely applying technology to solve the problem that devised another, rarely are problems or open

projects that require creativity applied. This paper shows an experiment carried out during the 2009/10 course and funded by the Plan de Renovación de Metodologías Docentes Universidad Sevilla in the Escuela Técnica Superior de Ingeniería Informática, where students had to make project proposals to resolve the tools of the subjects involved.

Palabras clave

Entrenamiento creatividad, Innovación docente, innovación tecnológica, aprendizaje por proyectos, ingeniería computadores.

1. Introducción.

La necesidad de que los Ingenieros sean capaces de innovar y ser creativos ha sido una constante universal a lo largo del tiempo, pero en la actualidad, y en nuestro entorno, dicha necesidad es aún más perentoria. Los medios de comunicación constantemente nos recuerdan que nuestro país se encuentra inmerso en una crisis, de la que probablemente sólo saldremos si se crea nuevas estructuras productivas mediante ideas innovadoras. Nuestra sociedad demanda de forma urgente que los titulados universitarios sean cada vez más innovadores y emprendedores. Sin embargo, en los estudios de Ingeniería normalmente se instruye en competencias sobre diseño en las que es fundamental conocer las tecnologías y herramientas correspondientes a la materia en que el alumno se especializa. Por otra parte, los problemas y proyectos que el alumno

realiza a lo largo de su carrera suelen estar orientados y propuestos por el profesor, de manera que la idea básica forma parte del enunciado, el alumno no hace más que aplicar la tecnología. Incluso a la hora de la realización de los proyectos fin de carrera al alumno se le suele plantear qué debe desarrollar, con lo que la creatividad apenas suele ejercitarse durante los estudios.

Hay que reconocer que esto no es así en todas las enseñanzas técnicas. Si tomamos como ejemplo la Arquitectura encontramos que una buena parte de las materias que componen dicha titulación intenta entrenar en crear y tener nuevas ideas. Incluso en las competencias de dicha titulación figura expresamente: “capacidad de generar creativamente ideas y formas”. Este tipo de competencias son muy difíciles de entrenar, a un alumno de Bellas Artes se le pueden enseñar las herramientas para dibujar y las técnicas de dibujo, pero “crear” requiere algo más, entre otras cosas conocer los precedentes, y también el presente desde todos los puntos de vista, al fin y al cabo el artista debe conocer la historia del arte y la sociedad en la que desarrolla su trabajo. Sabemos que todo esto no basta, pero aun así las asignaturas en este tipo de estudios están orientadas a dicho entrenamiento. ¿Es aplicable esto mismo a los estudios de Ingeniería? Evidentemente hay diferencias, pero innovar o crear algo en el campo de la ingeniería supone también conocer, además de la tecnología actual, el entorno social, económico y empresarial en el que se desarrolla la actividad. Además, es necesario diferenciar claramente qué son las herramientas tecnológicas, su utilización, fundamento etc., y el proceso de creación de nuevas ideas y productos, basándose en dichas tecnologías.

Una impresión particular en algunos docentes es que, además de no entrenar adecuadamente la creatividad del alumno, los profesores por el contrario cada vez deben ser más creativos para motivar a un alumnado, que en cierta forma, llega más inmaduro a los estudios superiores. En este sentido podemos pensar en dos posibles fallos, por una parte podemos estar dando demasiado “mascados o digeridos” los contenidos de las asignaturas, y por otra, puede que sean excesivamente específicas las materias que impartimos, cuando el mundo de la ingeniería claramente es multidisciplinar. Cabe plantearse si

no puede ser un elemento de motivación para el alumno que ellos desarrollen trabajos más creativos y que el profesor modere precisamente su creatividad.

En la Universidad de Sevilla, en el curso 2007, se puso en marcha el Plan de Renovación de las Metodologías Docentes 2007 (PRMD) [7]. Este plan estaba dividido en varias líneas, la cinco se dedicaba a los proyectos de Innovación Docente, en concreto, a sufragar la implantación de nuevas metodologías docentes. Este trabajo muestra una experiencia que parte de las premisas enunciadas anteriormente, la cual se desarrolló en el departamento de Arquitectura y Tecnología de Computadores durante el curso 2009/10 financiada, entre otros, por dicho plan, y que en el actual curso 2010/11 se sigue aplicando.

2. Objetivos

En el proyecto de innovación docente para el entrenamiento de la creatividad se plantea que el alumno proponga nuevas ideas, a ser posible con aplicación comercial, en el ámbito de las asignaturas implicadas, y las lleve a cabo aplicando las tecnologías que en las asignaturas se le enseña, pero relacionándolas con el resto de las asignaturas de la carrera y su conocimiento del entorno y la sociedad. Se trata, no sólo de proponer proyectos y problemas para que el alumno los desarrolle y le sirvan de ejemplo (enseñanza basada en proyectos o problemas), sino también de que el alumno proponga nuevas ideas en las que utilizar lo que se enseña. Creemos que este es un aspecto doblemente innovador, por una parte pretendemos utilizar las metodologías activas a las que hace referencia la convocatoria de proyectos de innovación docente de la Universidad de Sevilla, pero por otra vamos más allá: pretendemos modificarlas para que sea una forma de hacer que el alumno, no sólo aplique los conocimientos, sino que intente decir en qué se puede aplicar dichos conocimientos y buscar ideas comerciales al respecto. Sabemos que muchos “reinventarán la rueda”, pero eso no importa, lo fundamental es que empiecen a “mover esos resortes mentales en los que se basa la creatividad”. Aunque las ideas que se planteen por parte de los alumnos no sean innovadoras, hay que tener en cuenta que el simple hecho de intentar

copiarla es un ejercicio de creatividad en sí mismo.

En lo que respecta a la docencia en el departamento de Arquitectura y Tecnología de Computadores hay una serie de materias que claramente tratan sistemas abiertos adaptables a las más diversas aplicaciones (microcontroladores, FPGAs, software de procesamiento de imagen, etc.). Aunque el departamento en su conjunto está implicado en este proyecto de innovación docente, hemos hecho una selección de las asignaturas del departamento que mejor se pueden adaptar a este tipo de entrenamiento en creatividad, basándonos sobre todo en su especialización y en que el número de alumnos no impida una buena ejecución de la metodología propuesta, las asignaturas seleccionadas son: Informática Industrial (II), Diseño e Implementación de Computadores (DIC), Tecnología de Microcontroladores (TM), Arquitectura de Sistemas Tiempo Real (ASTR) y Tecnología para la Imagen Documental (TID). Estas cinco asignaturas corresponden a cuatro titulaciones diferentes, tres de Informática y una de Ingeniería Tec. Industrial. Durante el curso 2010/11 se le ha sumado, a las asignaturas anteriormente mencionadas, Diseño con Microcontroladores, correspondiente al Master en Ingeniería de Computadores y Redes que se imparte en la ETSII.

3. Metodología

Las metodologías básicas que se han utilizado son las de aprendizaje por proyectos y/o problemas. Este tipo de metodología está ampliamente tratado en diversos foros y publicaciones [8][1][2][3][4]. Además, estas metodologías ya habían sido utilizadas por algunas de las asignaturas implicadas en el presente proyecto de entrenamiento de la creatividad [5][6]. En este caso se trata de adaptar el resto de asignaturas y de modificar este tipo de metodologías docentes activas, de forma que los alumnos, después de realizar el/los proyectos/problemas propuestos por el profesor, propongan a su vez otros proyectos innovadores por su cuenta y resuelvan dichos proyectos, al menos parcialmente, teniendo en cuenta su viabilidad comercial.

El procedimiento inicialmente propuesto consistía en dividir el periodo de docencia de las asignaturas en tres tramos:

- Tramo 1: En un primer periodo se imparten los fundamentos de las asignaturas, la teoría y las prácticas de laboratorio básicas para entender la tecnología y herramientas propias de las diversas materias.
- Tramo 2: Se les proponen a los alumnos una serie de proyectos o problemas que deben desarrollar. En las asignaturas que ya utilizan esta metodología lo normal es que los alumnos elijan voluntariamente los proyectos de una serie propuesta. Los alumnos desarrollan esta fase en su mayor parte durante las horas no presenciales, pero otra parte la realiza en horas de prácticas de laboratorio con supervisión de los profesores.
- Tramo 3: Los alumnos proponen un proyecto innovador, la idea de dicho proyecto debe desarrollarse lo más posible y entregarse al profesor para su consideración, debe contener un breve análisis sobre “mercado y sociedad”. Si dicho proyecto recibe el visto bueno, el alumno debe desarrollarlo en más profundidad. Hay asignaturas que disponen de material que puede prestarse al alumno (elaborado en pasadas convocatorias del PRMD), otras deben elaborar nuevo material. A los alumnos que presenten proyectos verdaderamente innovadores y de suficiente envergadura, no sólo se les premiará con la correspondiente calificación en la asignatura, sino que se les planteará que desarrollen en profundidad la idea como proyecto fin de carrera. Si además se ven posibilidades comerciales, se intentaría, mediante los mecanismos de que dispone el departamento y la Universidad, buscar patrocinadores empresariales. Esta fase se desarrolla fundamentalmente en horas no presenciales, a los alumnos se les presta el material necesario para desarrollar la actividad. En el caso de que dicho material no pueda prestarse se les permite el uso de laboratorios no supervisados, pero con control de acceso mediante tarjetas y vídeo-vigilancia a través de internet (en el departamento se ha instalado un laboratorio de este tipo para ser usado tanto en proyectos fin de carrera como en experiencia del tipo que describimos). Los

alumnos deben presentar los trabajos desarrollados en el laboratorio al profesor para que sean evaluados.

Sin embargo, esta idea original de dividir la docencia en tres tramos fue imposible de llevar a la práctica por falta de medios o por falta de tiempo (evidentemente no se puede mantener los contenidos y ampliar aún más las actividades). En su lugar las asignaturas implicadas optaron por modelos intermedios:

- En las asignaturas ASTR y TM se exigía a los alumnos entregar una idea original, en realidad un producto comercial, que tuviese que ver, aunque fuera parcialmente, con la asignatura. Todos los alumnos de estas asignaturas debían elaborar un documento en el que se plasmase el desarrollo de un producto, la entrega de dicho documento era obligatoria para aprobar. Al mismo tiempo que entregaban sus trabajos los alumnos debían indicar su intención de desarrollarlo en la realidad o no. Si el profesor admitía el desarrollo del producto por parte del alumno, y el alumno lo hacía, a éste se le compensaba el tiempo dedicado a este trabajo con la exención parcial de la obligación de asistencia a las prácticas de la asignatura. En realidad llevar a cabo el desarrollo del producto era decisión del alumno, pero el profesor tenía la última palabra de admitir o no el proyecto. La compensación de las prácticas por dicho esfuerzo de desarrollo está más que justificada, pues generalmente requiere más horas el desarrollo de un producto real que unas prácticas de laboratorio.
- En las asignaturas DIC e II, las cuales tienen un número de alumno más elevado que las anteriores, los profesores decidieron dejar el sistema como optativo. Es decir, los alumnos no tenían obligación de presentar una memoria con una idea innovadora, eso sí, aquellos que la presentase y la desarrollase tendría ciertas horas de prácticas compensadas.
- Por último la asignatura TID, también con un elevado número de alumnos, utilizó la experiencia como método de evaluación alternativa. Se planteó a los estudiantes la posibilidad de proponer un proyecto, desarrollando la idea con detalle y

entregándola al profesor para su consideración. Si dicho proyecto recibía el visto bueno, el alumno/a debía desarrollarlo en más profundidad y realizar una presentación en clase a final de curso.

En todas las asignaturas la propuesta que los alumnos realizasen debía incluir al menos los siguientes apartados: Introducción (con un breve análisis sobre la situación social y tecnológica actual en relación con el sistema a desarrollar), Objetivos, Plan de Trabajo y Resultados Esperables.

4. Resultados de la experiencia

Por asignaturas estos fueron los resultados:

- ASTR (15 alumnos): Todos los alumnos aprobados entregaron proyecto de innovación (era requisito indispensable para aprobar), sin embargo, sólo dos optaron por desarrollarlo.
- TM (33 alumnos): Al igual que la anterior asignatura, los 21 alumnos aprobados en primera convocatoria entregaron el proyecto de innovación, desarrollaron el proyecto 8 alumnos, y 3 continúan en la actualidad realizando dicho trabajo como proyecto fin de carrera.
- TID (51 alumnos): Se recibieron un total de 14 propuestas, de las que 3 fueron inicialmente desestimadas, ya que la documentación del proyecto no cumplía los requisitos mínimos. De las 11 propuestas inicialmente aceptadas, finalmente sólo 5 fueron desarrolladas, entregadas y defendidas a final de curso. Es necesario tener en cuenta que este método de evaluación era opcional para esta asignatura. Los estudiantes tenían la alternativa de aprobar a través de un examen, por lo que la realización y posterior evaluación pública del trabajo requería un esfuerzo mucho mayor. Lo que hizo que algunos estudiantes desistieran. A pesar de todo, los resultados fueron muy satisfactorios, pues los 4 alumnos que finalmente presentaron y defendieron el proyecto realizaron un gran trabajo, obteniendo las más altas calificaciones.
- DIC (45 alumnos): Es una asignatura del primer cuatrimestre y cuando se intentó implementar el mecanismo estaba próxima su

finalización. A pesar de todo, se presentaron 6 propuestas y ninguna se llegó a desarrollar.

- II (80 alumnos): Es una asignatura anual, la metodología alternativa se propuso a mitad de curso, cuando los alumnos ya habían realizado numerosas prácticas y tenían planificado el resto de la asignatura. Se presentaron 11 propuestas y ninguna llegó a realizarse.

Por último, a todos los alumnos de las asignaturas implicadas (hubieran o no participado en la experiencia) se les planteó una encuesta voluntaria con sólo tres preguntas:

1. Indicar si en la correspondiente carrera se ejercita la creatividad o innovación de los alumnos: 1 nada, 2 poco...5 fuertemente. *La media fue 1,4.*
2. Indicar como de necesaria es dicha ejercitación para el alumno y su futuro profesional: 1 no necesario, 2 poco necesario, 3 necesario, 4 muy necesario, 5 imprescindible. *La media fue 3,5.*
3. Indicar si el método empleado le parece bien para ejercitar la creatividad y la innovación: 1 no adecuado, 2 poco adecuado, 3 adecuado, 4 bastante adecuado, 5 el más apropiado. *La media fue 3,2.*

Además había un apartado de observaciones para que los alumnos indicaran su opinión. La mayoría no lo rellenaron, pero la observación más repetida fue que no había tiempo para tantos trabajos de clases y encuestas, uno incluso llegó a indicar textualmente “que la carrera ya estaba demasiado sobrecargada para que los profesores encima hiciéramos nuevos inventos”.

En la encuesta participaron un total de 55 alumnos, lo cual representa una baja participación con respecto a los más de 200 que podrían haber opinado.

Asignatura	Alumnos matriculados	Ideas presentadas	Ideas desarrolladas
ASTR	15	15	2
TM	33	21	8
TID	51	14	5
DIC	45	6	0
II	80	11	0
Totales	224	67	15

Tabla 1. Resumen de la participación de los estudiantes en la experiencia.

5. Variaciones de la metodología empleada.

La mayor parte de las asignaturas ha continuado en el siguiente curso (2010/11) con la misma metodología que emplearon en el curso anterior. Pero se ha introducido una importante novedad, la auto-evaluación cruzada (co-evaluación). Este método de evaluación consiste en que los alumnos intervienen como evaluadores de sus propios compañeros junto con el profesor, en lo que se refiere a los proyectos creativos de innovación. La principal ventaja, desde el punto de vista docente, con este tipo de autoevaluación cruzada, es que entrena aspectos como el espíritu crítico y el de análisis entre los alumnos. Es cierto que el análisis es una faceta que en las carreras de ingeniería siempre se ha entrenado, pero enseñar a tener “espíritu crítico” es más propio de otro tipo de titulaciones. Sin embargo, al igual que es importante tener creatividad e ideas innovadoras también es de vital importancia para el “ingeniero emprendedor” saber reconocer las buenas ideas y valorarla en su justa medida. Es por ello que las asignaturas implicadas se han planteado, además de entrenar la creatividad, que el alumno adquiriera experiencia como evaluador de ideas.

El evaluar las ideas de los demás, reconociendo aciertos y fallos en las mismas, así como su exposición, permite mejorar las ideas propias y como las debemos presentar. Este último aspecto es bastante importante, pues muchas ideas brillantes no son tomadas en consideración por no estar adecuadamente presentadas. Creemos que el descubrir dichos fallos en los trabajos de los demás debe servir al alumno para que él mismo no los cometa.

La autoevaluación cruzada de las ideas innovadoras se está llevando a cabo con las siguientes particularidades:

- El profesor hace público antes de nada los criterios de evaluación que se aplicarán a las ideas innovadoras, así como los formularios que rellenarán los evaluadores.
- Se realiza un sorteo para ver qué alumnos evalúan a otros, no evaluando cada alumno más de tres ideas y siendo evaluado cada alumno por al menos dos compañeros (el

anonimato es casi imposible de lograr en este tipo de asignaturas lo cual resta bastante posibilidades a la metodología).

- Los alumnos evaluadores deben razonar y motivar sus puntuaciones.
- El profesor evalúa también las ideas innovadoras, siendo la puntuación final que obtiene el alumno una media ponderada de las propuestas por los diferentes evaluadores, incluidos los profesores (la ponderación depende de la asignatura).
- Los informes de los evaluadores son revisados por los profesores y a su vez evaluados.

Esta metodología supone una nueva complicación y añadir de nuevo más tareas al ya apretado calendario escolar. De todas formas su aplicación está en marcha actualmente, y su viabilidad y resultados los tendremos al final del presente curso 2010/11.

6. Conclusiones

Hay un gran número de alumnos que prefieren que les digan qué tienen que hacer, y operar además con reglas y mecanismos automáticos, de forma que la resolución de los problemas que se plantean realmente se lleva a cabo por la simple aplicación de dichas reglas automáticas. Ese tipo de aprendizaje es el más cómodo, pues el alumno sabe que estudiando siempre obtiene los resultados deseados. Cuando se trata de aplicar la creatividad entra en juego la inspiración y, con ella, limitaciones de tipo personales. Parafraseando a un alumno “sometido” a esta metodología docente: “hay veces que no es cuestión de tiempo, simplemente es que no se te ocurre ninguna idea”. Esto crea cierta ansiedad en los alumnos, los profesores implicados, para rebajarla, han hecho en general que la nota final de sus respectivas asignaturas no sea demasiado dependientes de los resultados de las ideas innovadoras, de forma que se premia a quien las tiene, pero no se castiga al que no las tiene. Esto supone un relajamiento y una pérdida de eficacia de esta metodología, pues al final casi lo único con lo que contamos los docentes para lograr que el alumno haga el esfuerzo es la evaluación (además de la motivación).

Podría pensarse que la motivación que supone la satisfacción del proceso creativo es otro mecanismo para que los alumnos hicieran dicho esfuerzo. Al principio de este trabajo nos preguntábamos si este tipo de metodología de entrenar la creatividad es motivadora, nuestra experiencia es que motiva a unos pocos alumnos, pero la mayoría lo ven como una dura prueba que les aumenta el estrés, la encuesta y las quejas personales de los alumnos por falta de tiempo así lo reflejan.

Por otra parte, es evidente que no todas las asignaturas se adaptan por igual a esta metodología. Hay asignaturas como Tecnología de Microcontroladores o Informática Industrial que parecen ser idóneas para entrenar la creatividad, puesto que es fácil plantear proyectos relacionados con su materia, al fin y al cabo los microcontroladores se parecen a una paleta de colores con los que plantear un “cuadro”. Las asignaturas de desarrollo software, aunque sean específicas, como Tecnología para la Imagen Documental, también parecen adecuadas. Sin embargo, en otras, como en Diseño e Implementación de Computadores, resulta muy difícil que el alumno pueda plantear un proyecto medianamente brillante.

En general, todos los actores que intervienen en el proceso educativo entienden la necesidad de ejercitar la creatividad y el aprendizaje. La mayoría de los alumnos, según reflejan nuestra breve encuesta, ven una carencia en ese sentido y una buena idea el cambio metodológico para incluir dicho entrenamiento. Pero la realidad es que, cuando se deja a voluntad del alumno dicho ejercicio, la mayoría optan por no realizarlo. Lo cual es normal en el caso que nos ocupa, pues supone gastar tiempo en una actividad añadida, la cual no ha sido suficientemente valorada.

Por otra parte, los alumnos que han llegado hasta el final han obtenido muy buenos resultados, mostrando además una gran satisfacción con la metodología usada, por lo menos es lo que subjetivamente perciben los profesores. Todas las asignaturas que se han implicado en este proyecto de innovación docente tienen previsto seguir con la experiencia en cursos sucesivos, de manera que el éxito de los alumnos participantes pueda servir de acicate para ampliar el número de estudiantes que eligen esta opción.

Por último, un aspecto no tratado en este trabajo, y que resulta muy difícil de juzgar, es si realmente hemos conseguido entrenar la creatividad en los alumnos implicados y en qué medida. Evidentemente un seguimiento comparativo a largo plazo podría ser útil, cuantificando, por ejemplo, el número de patentes, creación de empresas, premios por ideas, etc. que los alumnos en su futuro laboral consiguen. Pero por ahora carecemos de recursos para este tipo de comprobaciones. Sólo tenemos la evaluación que de las ideas presentadas hacemos los profesores. Con esta limitación, y desde un punto de vista siempre subjetivo, creemos que la iniciativa sí consigue su objetivo de entrenar en creatividad, o por lo menos sirve para concienciar al alumno de la necesidad que hay de tener nuevas ideas e imaginación.

Agradecimientos

La experiencia que se muestra en este trabajo ha sido financiada por el proyecto VULCANO (TEC2009-10639-C04-02) y por el Plan de Renovación de las Metodologías Docentes 2007 (PRMD) de la Universidad de Sevilla.

También queremos agradecer a los alumnos que han colaborado en esta experiencia su participación, sobre todo a aquellos que nos han animado a seguir en cursos sucesivos con este proyecto (también los profesores necesitamos que nuestros alumnos nos motiven).

Referencias

- [1] Markham T. (2003) Project Based Learning, a guide to Standard-focused Project based learning for middle and high school teachers. Buck Institute for Education.
- [2] Solomon G. (2003) Project Based Learning: a Primer, Technology a Learning, vol. 23, num. 6.
- [3] Dolamns H. y otros. (2005) Problem-based learning: future challenges for educational practice and research. Blackwell Publishing. Medical Education.
- [4] Hedley M. Barrie S. (1998) An undergraduate microcontroller systems laboratory. IEEE Transactions on Education, Volume: 41, Issue: 4.
- [5] Proyecto innovación docente 698 / 2007. Universidad de Sevilla. (2007) Plan de Renovación Metodologías Docentes 2007. www.institucional.us.es/plan_renovación.
- [6] Jimenez, G. Jimenez, A. Senhadji R. y otros (2008) Aprendizaje por proyectos en la enseñanza de los microcontroladores. XVI Congreso Universitario de Innovación Educativa en las Enseñanzas Técnicas (CUIEET), Cádiz.
- [7] *Universidad de Sevilla*. Plan de Renovación de Metodologías Docentes 2007. http://www.institucional.us.es/plan_renovación.
- [8] Valero-García, M. Navarro, J. FAQ sobre la docencia centrada en el aprendizaje. <http://epsc.upc.edu/~miguel%20valero/contingut3.html>.

Sistemas Operativos Avanzados: de la clase magistral al entorno colaborativo.

Joan Navarro, Xavi Canaleta, Andreu Sancho-Asensio

Departamento de Informática

La Salle - Universitat Ramon Llull

C/ Quatre Camins 2

08022 Barcelona

{jnavarro, xavic, andreus}@salle.url.edu

Resumen

La implantación del Espacio Europeo de Educación Superior (EEES) apuesta, entre otras cosas, por un nuevo modelo educativo en el que el alumno es el actor principal y el profesor queda relegado a un segundo plano. Esto conlleva el replanteamiento de la metodología docente del profesorado así como la reestructuración integral de ciertas asignaturas. En este trabajo se presenta el caso particular de la asignatura Sistemas Operativos Avanzados, impartida en el tercer curso de Ingeniería Técnica en Informática de Sistemas, y se expone cómo dicha asignatura, hasta ahora impartida según el clásico modelo magistral, ha evolucionado a un modelo pragmático basado en la filosofía *learning by doing*. Este proceso ha requerido una adaptación de las técnicas pedagógicas y del sistema de evaluación, cuyos resultados indican la aceptación del nuevo modelo por parte del alumnado y la mejora en la asimilación de sus conocimientos. Finalmente, el modelo expuesto para esta asignatura podría aplicarse fácilmente a asignaturas de características similares.

Summary

The European Higher Education Area (EHEA) demands a new teaching model where the student is the main actor and teachers are pushed into the background. This forces practitioners to review the current teaching methodology and the structure of many subjects. This work describes the case of Advanced Operating Systems, a subject taught in the third course of Computer Engineering, which has evolved from a master class paradigm to the learning by doing philosophy. It also shows how the assessment system and teaching methods have been adjusted, proving the acceptance of the new model by the students and improving their assimilation of

knowledge. Finally, the aforementioned model used in this subject might be easily applied to other subjects with similar features.

Palabras clave

Administración de sistemas operativos, ABP, EEES, aprendizaje activo.

1. Apertura

La Universidad La Salle, como entidad pionera en el fomento e implantación de las metodologías en innovación docente, siempre ha apostado por la mejora continua del profesorado y de las asignaturas que imparte.

Dentro de este marco de trabajo y con la entrada en vigor del EEES, se está dando la situación idónea para la reestructuración integral de las asignaturas. Muestra de ello son las numerosas publicaciones y propuestas que están emergiendo en estos últimos tiempos [5, 7, 8], la mayoría de ellas focalizadas en el Aprendizaje Basado en Proyectos (ABP), cuyo rendimiento satisfactorio ha sido demostrado ampliamente en la literatura docente, como por ejemplo en [1].

El ABP es una aplicación concreta de la filosofía *Learning by Doing*. El pilar de su metodología es la interacción con los alumnos, los cuales tienen que trabajar en grupo [6] y de forma colaborativa, con el objetivo común de solucionar interesantes retos. Dicha metodología también ofrece la oportunidad a cada miembro del grupo de aportar sus habilidades personales para el bien común. Los conocimientos adquiridos mediante el ABP son de un valor incalculable, puesto que están conectados con una realidad palpable al alcance del alumno. Por último, el ABP sugiere que la forma de evaluar estos conocimientos adquiridos es mediante la comunicación y

la (auto)evaluación. En este entorno, el profesor se convierte en guía y mentor de sus alumnos hacia el conocimiento.

En este trabajo, además de constatar la eficacia del ABP, se describe el proceso de transición pedagógica que se ha llevado a cabo en la asignatura Sistemas Operativos Avanzados (SOA), en la cual el modelo magistral ha dejado paso a un modelo colaborativo [2]. Evidentemente, el motor de este cambio no podía centrarse en otro punto que no fuera el alumno.

Actualmente, SOA (bajo el nombre de ASSOO) sólo se imparte como asignatura troncal en el tercer curso de Ingeniería Técnica en Informática de Sistemas. Con la entrada en vigor de los nuevos grados del EEES, no solamente los graduados en Informática, sino también los graduados en Telemática tendrán que cursar esta asignatura. Esto implica que el volumen de alumnos en el futuro más próximo se incrementará sustancialmente. Según el *roadmap* de la universidad, el curso académico 2010 – 2011 es el último curso en el que se impartirá Ingeniería Técnica en Informática de Sistemas, iniciando totalmente los nuevos grados en el curso 2011 – 2012. Con todo esto, lo que se quiere destacar es que este es el año perfecto para implementar, a modo de prueba piloto, la filosofía del aprendizaje activo en SOA.

El resto del trabajo se organiza como sigue. En la sección 2 se describe el funcionamiento de la asignatura según el modelo magistral. En la sección 3 se evalúa el modelo descrito en la sección anterior y se presentan los cambios necesarios para la transición al modelo colaborativo. La sección 4 describe el detalle de la aplicación actual del curso piloto. En la sección 5 se muestran los resultados obtenidos de aplicar este cambio radical y sus beneficios. Por último, en la sección 6 se discuten las conclusiones y el trabajo futuro.

2. Preludio

En esta sección se describe el funcionamiento de SOA (en realidad ASSOO) tal y como estaba diseñada antes del cambio que aquí proponemos (curso 2009 – 2010). Con estos datos se podrá seguir y comprender la evolución de la asignatura hacia el nuevo modelo. Después de esta sección se expondrán los puntos fuertes y débiles, en los que se tiene que apoyar la nueva SOA.

En sus inicios, SOA se llamaba Administración de

Administración de sistemas operativos	Horas lectivas
1. Introducción	3
2. El sistema operativo UNIX	4
3. Instalación del sistema UNIX	8
4. Inicialización y cierre	10
5. Usuarios y grupos	10
6. Seguridad	7
7. Automatización de tareas	24
8. Copias de seguridad	6
9. Administración de red	8
Total	80

Cuadro 1: Contenido de la asignatura ASSOO

Sistemas Operativos (ASSOO). Como hasta ahora, esta era una asignatura troncal en la Ingeniería Técnica en Informática de Sistemas que tras las múltiples modificaciones de los planes de estudios, seguía fiel a sus principios sin apenas modificar sus contenidos ni su filosofía.

ASSOO era una asignatura anual con una carga para el alumno de 2 horas semanales de clase más una práctica (que se realiza a lo largo de todo el curso). Estas clases se impartían de forma magistral mediante diapositivas, en las que el profesor iba mezclando los contenidos teóricos de la asignatura con experiencias personales vividas en el campo de la administración de sistemas. En el Cuadro 1 se exponen los contenidos generales de la asignatura tal y como figuran en el plan de estudios junto a las horas lectivas que se dedicaban antes de llevar a cabo la propuesta aquí descrita. Obsérvese que la unidad didáctica de automatización de tareas (*scripts*) tenía una carga muy superior (31 % de las horas lectivas) a las demás. Para evaluar la parte teórica de la asignatura, se sometía al alumno a un único examen por cuatrimestre.

La parte práctica de la asignatura se realizaba en parejas, y consistía en la configuración y administración de un sistema operativo Linux. Debido a la envergadura y complejidad de esta única práctica, el alumno tenía la posibilidad de ir entregándola por módulos, es decir, cada una de las cuatro fases en que se dividía la práctica. La entrega de la práctica, tanto la final como la parcial, se realizaba mediante una entrevista con el profesor, en la que el alumno debía demostrarle que, además del correcto funcio-

namiento de la práctica, tenía unos conocimientos profundos sobre el funcionamiento de la misma.

Aunque se usara el modelo magistral, ASSOO era considerada una asignatura eminentemente práctica, por lo que en la nota final del alumno se ponderaba por igual la práctica y la teoría, tal y como se muestra en la Ecuación 1.

$$N_{Final} = 0,5 \cdot N_{Teoría} + 0,5 \cdot N_{Práctica} \quad (1)$$

Sobre el papel, el objetivo principal de la asignatura era claro: formar administradores de sistemas operativos. A la realidad, debido a restricciones logísticas y temporales, la asignatura concentraba el 95% de su tiempo en el sistema operativo Linux.

El planteamiento de este modelo docente derivaba en un notable absentismo en las aulas y unas calificaciones finales bajas tal y como se demuestra en la Figura 1. Desafortunadamente, a medida que iban pasando los años, esta tendencia se acentuaba. Según las encuestas realizadas a los alumnos, algunos de éstos achacaban su absentismo a la excesiva carga de la práctica. Otros, afirmaban que el hecho de tener las diapositivas en el campus virtual, les daba la posibilidad de leerlas en casa sin tener que asistir a clase.

Dichos comentarios plantearon algunas dudas al equipo docente sobre el correcto funcionamiento de la asignatura sugiriendo la posibilidad de valorar un cambio en su impartición.

3. Paráfrasis

Mediante las encuestas de calidad docente que rellenan los alumnos de todos los cursos, se llegó a la conclusión que los dos principales factores que amenazaban el éxito de la asignatura basada en el modelo tradicional eran el (1) desinterés por parte del alumno y (2) su bajo rendimiento. El objetivo de esta sección es enfatizar las debilidades y las fortalezas resultantes de aplicar dicho modelo en ASSOO.

A continuación se puntualizan las sensaciones más relevantes recogidas por el equipo docente cuando la asignatura se impartía según el modelo clásico:

- No se conseguía que la práctica de la asignatura fuera al unísono con la teoría. Esto implicaba que en las actas de Junio, se llegara a tener

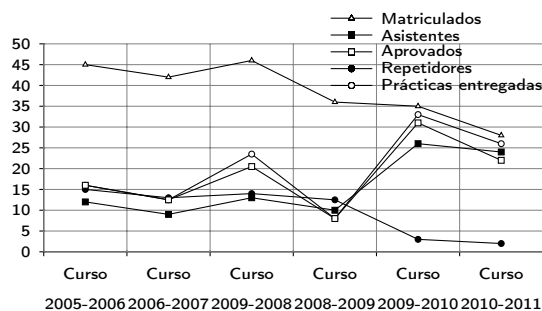


Figura 1: Estadísticas de la convocatoria ordinaria

más de un 80% de alumnos no presentados (al no presentar la práctica no podían optar a tener nota de la asignatura). Este porcentaje bajaba al 30% en la convocatoria de Septiembre. En otras palabras, los alumnos desarrollaban la práctica durante el verano.

- El porcentaje de alumnos que asistían asiduamente (más de un 90% de las sesiones) a clase rozaba el 30% de los matriculados.
- Un año después de cursar la asignatura, los alumnos afirmaban “haberse olvidado” de los conocimientos supuestamente adquiridos.
- Los alumnos no se sentían a gusto con la asignatura: Aun valorando positivamente su importancia, la encontraban demasiado teórica.
- A pesar de tener un elevado número de alumnos suspendidos, los alumnos no encontraban que la asignatura fuera especialmente difícil; según ellos las malas estadísticas se debían a la falta de tiempo.

En la Figura 1 se muestra el número de alumnos (1) matriculados en la asignatura, (2) que asisten asiduamente a clase, (3) aprobados en la convocatoria ordinaria, (4) que han aprobado la práctica en la convocatoria ordinaria y (5) repetidores. Idealmente, las 4 primeras curvas deberían ir juntas, mientras que la última debería ser 0. Nótese que los resultados hasta el curso académico 2008 – 2009 son alarmantes.

A partir de estos puntos, el equipo docente decidió plantearse qué ventajas e inconvenientes presentaba la clase magistral. En líneas generales, esta manera de trabajar tenía las siguientes ventajas:

- Por parte del profesor, no se requería de demasiado tiempo para preparar las clases. A lo largo

de los años se había recopilado material suficiente.

- El marco que ofrecía el modelo magistral encorsetaba el hilo argumental de la asignatura quedando muy acotado el discurso del profesor y sesgando la parte más creativa de la asignatura.
- El hecho de basar las explicaciones magistrales en diapositivas digitales facilitaba la planificación de la asignatura, así como el reajuste de los contenidos en función de eventualidades en el calendario.

Estas ventajas quedan reducidas a su mínima expresión cuando se comparan con su único pero gran inconveniente: **los alumnos no aprendían como se deseaba.**

Humildemente, el equipo docente se dio cuenta de que la persona más beneficiada del modelo magistral, en la asignatura de ASSOO era el profesor. En un entorno universitario esto no se podía permitir, por lo que era necesario hallar la manera de revertir esta situación adversa. Como no podía ser de otra forma, la solución pasaba por el *Learning by Doing* y el ABP.

En la siguiente sección se describen el conjunto de medidas que se han adoptado para migrar a un entorno colaborativo.

4. Interludio

Después de analizar todos los factores negativos que conllevaba la clase magistral, se llegó a la conclusión de que la raíz del problema residía en (1) el entorno físico y (2) en algunos de los contenidos. Efectivamente, lo primero que debía cambiar para fomentar el aprendizaje activo, era el aula. Una asignatura como ASSOO, no se podía impartir sin que el alumno tuviera un ordenador con el que poder practicar todos los conceptos que iban apareciendo. Además, debido a la distribución de contenidos y tal y como refleja el Cuadro 1, el alumno tenía la falsa impresión que la administración de sistemas operativos se limitaba a la automatización de tareas (*scripts*).

En el curso 2009 – 2010, primer año en el que el equipo docente decide tomar cartas en el asunto, se llega a la conclusión de que la clase tiene que ser impartida en el laboratorio. La verdad es que hay que apreciar el matiz de esta frase; tenemos la firme

creencia de que no es lo mismo impartir una clase **en** el laboratorio, que impartir una clase **de** laboratorio. Una clase de laboratorio, suele ser asociada por los alumnos a un entorno distendido en el que van a reforzar los conocimientos adquiridos durante las sesiones teóricas. En cambio, una clase en el laboratorio implica que los alumnos tendrán que adquirir los conocimientos durante esta sesión, es decir, el alumno tendrá que aprovechar al máximo la sesión y los recursos que temporalmente tiene a su alcance para consolidar los conceptos.

Impartir la clase en el laboratorio tiene la ventaja de que la relación entre el alumno y el profesor es más próxima. Por un lado, el alumno se siente a la misma altura que el docente, lo que le facilita el diálogo y el entendimiento entre ambos. Por otro lado, el docente está más próximo al alumno, lo que le facilita la evaluación integral del alumno, pudiendo seguir su evolución durante el transcurso de la asignatura.

Una vez solucionado el entorno de trabajo es momento de revisar los contenidos. Llámese ASSOO (en los planes de estudios de Ingeniería Informática) o llámese SOA (en los planes de estudios de los nuevos Grados), el objetivo de la asignatura es el mismo: formar administradores de sistemas. ¿Es cierto que todos los administradores de sistemas trabajan únicamente con sistemas Linux? Probablemente no, en realidad, aunque fuera cierto, siempre es peligroso focalizar una asignatura entorno a un único concepto, sea Linux, sea Windows, sea el que sea. Nadie sabe qué sistemas tendrán que administrar nuestros ingenieros del mañana. En conclusión, se tienen que equilibrar los conceptos. A tal efecto, se ha decidido dividir la materia del curso en tres partes:

- Sistemas operativos Linux: 50% de las horas lectivas del curso.
- Sistemas *embedded*: 25% de las horas lectivas del curso.
- Sistemas operativos Windows Server: 25% de las horas lectivas del curso.

Entonces, los contenidos de la asignatura tienen que quedar uniformemente en estas tres áreas temáticas. Así, se balancean las horas lectivas del Cuadro 1 de modo que se consigue una distribución más uniforme. Por ejemplo, se dedican más horas lectivas a la instalación del sistema operativo, puesto que se instala en múltiples plataformas, y se reducen las

horas lectivas dedicadas a la automatización de tareas.

Tal y como se comenta más adelante, ofreciendo un abanico más amplio de plataformas se incrementa la satisfacción del alumno. Además, se elimina la percepción de que la asignatura se limita a programar *scripts*.

Hasta ahora, se ha presentado una reforma únicamente a nivel estructural de la asignatura; la cual facilitará la implementación del *Learning by Doing* mediante el ABP. A continuación se aborda el tema de la práctica.

Según el plan de estudios del nuevo grado en Informática [3], SOA es una asignatura que tiene 6 créditos ECTS. En el caso de que sea una asignatura anual, como lo es en el curso académico 2009 – 2010, esto significa que el alumno tiene que dedicar alrededor de 5 horas de trabajo¹ a la semana para aprobar la asignatura.

En realidad, la idea de que los alumnos deban realizar una práctica para consolidar sus conocimientos es francamente buena. Lo que no nos parece correcto es que la práctica no esté sincronizada temporalmente con la teoría. Entonces, se tiene que rediseñar la práctica. Aunque sigue siendo una práctica para realizar en parejas, la hemos rediseñado tanto a nivel conceptual como a nivel logístico.

A nivel conceptual, la práctica tiene que coincidir con los conceptos expuestos en la teoría, pues tendrá que tocar explícitamente cada una de las tres plataformas descritas anteriormente (Linux, embedded y Windows Server). Además, la carga de trabajo también tendrá que ir acorde con la teoría (50%, 25% y 25% respectivamente). Con esta distribución conceptual y temporal de la práctica, se consigue que el alumno la siga a tiempo, ya sea con el objetivo de reforzar los conceptos trabajados en la parte experimental de la asignatura (anteriormente, en ASSOO, llamada parte teórica) o por el temor a olvidarlos y no ser capaz de seguir el ritmo de las clases.

A nivel logístico, la práctica se dividirá en cuatro puntos de control (tal y como se hacía con el modelo antiguo) pero esta vez serán obligatorios y se variará el formato de la entrega.

Para cada una de las cuatro fases de la práctica, todas las parejas tendrán que depositar la memoria,

de la que obtendrán una nota N_M , en el campus virtual. Una vez ahí y después de una entrevista personal (N_{EP}), el equipo docente analizará las soluciones propuestas por cada pareja y hará grupos de parejas, de forma que dentro de un mismo grupo aparezcan soluciones contradictorias. En el laboratorio, los grupos de parejas tendrán que reunirse y discutir una única solución para presentar al resto de la clase, obteniendo así una nota N_{EG} . Mientras tanto, el docente puede ir recopilando datos empíricos y de primera mano sobre las discusiones de sus alumnos y recogiendo sus impresiones.

No es suficiente con hacer una declaración de intenciones sobre los objetivos y el funcionamiento de la práctica. Para poder aplicar el ABP se tiene redactar una práctica suficientemente abierta como para que invite a la discusión y el debate y suficientemente cerrada como para que el tema quede acotado a la asignatura.

Llegados a este punto, se ha descrito la remodelación de la parte práctica de la asignatura así como su arquitectura básica. Únicamente queda pendiente la parte teórica, el núcleo del conocimiento, el eje alrededor del cual gira la asignatura.

Tal y como se ha expuesto anteriormente, cuando un alumno está en el laboratorio, éste debe encontrar la forma óptima de aprovechar los recursos de los que dispone: laboratorio, compañeros y personal docente. Entonces, leer diapositivas, leer libros, leer o cualquier cosa que implique una actividad individual, se considera malbaratar los recursos que el alumno tiene disponibles en un espacio acotado de tiempo. El alumno ha de ir al laboratorio a practicar, a hacer y evidentemente, a aprender a hacer. Nótese que todas estas premisas están en contraposición con el clásico modelo magistral.

Entonces, hace falta una reestructuración completa de la forma de impartir conocimiento hacia los alumnos. Los alumnos pueden y deben trabajar en casa, por lo tanto, hay que buscar qué tareas relacionadas con la asignatura pueden realizar fuera del laboratorio. Efectivamente, la respuesta está en la lectura. Entonces, la estructura de la parte teórica de la asignatura queda como sigue.

En primer lugar, es fundamental cambiar el paradigma: en SOA no hay clases o sesiones teóricas, hay sesiones experimentales. Este sutil cambio de nomenclatura, refleja las intenciones del párrafo anterior al mismo tiempo que se ajusta más a la filoso-

¹Se toma como referencia que (a) 1 crédito ECTS se corresponde con 27,5 horas de trabajo para el alumno y (b) un curso tiene 33 semanas.

fía ABP.

Antes de cada sesión experimental, se les exige a los alumnos que lean un capítulo de la teoría (que evidentemente estará relacionado con el contenido de la sesión). Además, para apoyar el proceso de lectura, se les proporciona un foro en el campus virtual para que expongan dudas conceptuales relacionadas con la teoría. De este modo son los mismos compañeros los que se responden las dudas o, en última instancia el docente. Cuando el alumno llega al laboratorio, el docente le entrega una serie de mini-proyectos o miniprácticas relacionadas con la teoría que ha leído, los cuales el alumno tiene que resolver individualmente y entregar al finalizar la sesión. Éste método de trabajo ofrece las siguientes ventajas:

- El alumno es capaz de autoevaluar hasta qué punto ha asimilado los conceptos teóricos.
- El docente es capaz de evaluar las aptitudes y habilidades reales del alumno.
- El ambiente es propicio para una comunicación constructiva entre los alumnos y el profesor.

Otra vez, hay que ir con cuidado. Preparar las miniprácticas no es trivial. Para la satisfactoria implementación de esta metodología, los miniprácticas deben satisfacer las siguientes restricciones:

- Estar orientadas a reforzar los conceptos expuestos en el capítulo de la teoría.
- Ser lo más cerradas posible para que el alumno sea capaz de resolverlos en el laboratorio.
- Por cada tema, estas deben estar ordenadas por dificultad, de forma que los alumnos más avanzados sean capaces de llegar a resolverlas todas y los menos avanzados puedan consolidar sus conocimientos.
- Tienen que ser desafiantes para que puedan llegar a estimular el ingenio de los alumnos y con ello despertar su motivación.

Además, para fomentar la asistencia a clase y las ganas del alumnado, cada día de clase los alumnos podrán realizar un examen que, en caso de aprobarlo liberará el capítulo para el examen de cuatrimestre. De nuevo, hay que reparar especial atención a este examen: debe ser un examen rápido y conciso que sea capaz de discriminar si el alumno ha adquirido los conocimientos sobre el capítulo. En caso de tener una duración excesiva, el alumno se quedaría sin tiempo para resolver los miniproyectos.

Por último, queda recalcularse la nota final (Ecuación 2) de los alumnos, teniendo en cuenta la reestructuración de la asignatura.

$$\begin{aligned} & \text{iii}(N_{Teoría} \geq 5 \text{ y } N_{Práctica} \geq 5), \\ N_{Final} &= 0,6 \cdot N_{Teoría} + 0,4 \cdot N_{Práctica} \\ N_{Teoría} &= 0,8 \cdot N_{Examen} + 0,3 \cdot N_{EC} \\ N_{Práctica} &= 0,4 \cdot (N_{EP} + N_{EG}) + 0,2 \cdot N_M \end{aligned} \quad (2)$$

Analizando la Ecuación 2 se puede ver que el nuevo modelo pondera la nota que obtiene el alumno en función de las horas de trabajo real que ha dedicado para la asignatura. Nótese que la nota de teoría² está evaluada sobre un 110%, de modo que el 30% de la nota (N_{EC}) se corresponde a la evaluación continua del alumno, que incluye su participación activa en la clase así como en el campus virtual.

De este modo, se consigue incentivar al alumno para que huya del cada vez más típico “conformismo del suficiente” (informalmente, hemos observado que hay un volumen preocupante de alumnos que no luchan por sacar más nota que un aprobado) y que vea la máxima calificación (10) al alcance de su mano. Efectivamente, a los alumnos que saquen una nota superior a 10, se les deja la nota en 10.

Después de exponer cómo se ha modificado la asignatura para dejar paso a la metodología *Learning by Doing*, describiendo el papel del alumno y del profesor en este nuevo paradigma, es momento de apuntar las ventajas e inconvenientes de esta filosofía así como exponer los resultados obtenidos de la prueba piloto.

5. Post-ludio

Todo empezó en Septiembre de 2009; fue entonces cuando el equipo docente de la sección de Arquitectura de Computadores del Departamento de Informática decidió enterrar el modelo magistral para la asignatura ASSOO. Era una apuesta arriesgada puesto que implicaba un cambio muy significativo y había poco tiempo para prepararlo. El objetivo de esta sección es por un lado exponer las impresiones recogidas a lo largo de este curso por parte del personal docente y por el otro exponer los resultados que han obtenido los alumnos en este curso académico

²Con el nuevo modelo, la “nota de teoría” se corresponde con la nota obtenida en las sesiones experimentales. Se ha mantenido el concepto $N_{Teoría}$ para que las ecuaciones 1 y 2 sean comparables.

así como algunas de sus impresiones más relevantes.

Recordemos que esta asignatura se impartió como prueba piloto en el curso académico 2009 – 2010, con 35 alumnos matriculados de los cuales 26 asistieron asiduamente a clase (74% de asistencia). En el curso académico 2010 – 2011, se está perfeccionando el piloto con las reflexiones aquí expuestas y se está trabajando con 28 alumnos matriculados de los cuales 24 asisten asiduamente a clase (85% de asistencia).

Ciertamente, el primer contacto que tuvieron los alumnos con la asignatura y su metodología fue áspero. Durante las dos primeras semanas los alumnos se sentían incómodos en clase, no estaban acostumbrados a esta forma de trabajar, no estaban acostumbrados a ser los protagonistas; probablemente tampoco lo esperaban. Esto se notaba en que se les veía desubicados en el aula, no participaban, estaban callados esperando a que alguien les dijera qué hacer. Como por arte de magia, quizás por la persistencia del equipo docente, a partir de la tercera semana los alumnos entendieron perfectamente cuál era su rol dentro de este nuevo esquema didáctico.

Es a partir de este punto cuando se observa que los alumnos empiezan a venir antes de hora a clase y aún con la clase terminada se quedan ahí para seguir “jugando”. Los alumnos vienen motivados a clase, se disculpan por sus faltas de asistencia aún sin exigirles ningún mínimo de asistencia. El sistema parece que funciona.

No todo es coser y cantar, hay otras iniciativas que no acaban de funcionar como se esperaba: todas ellas relacionadas con el campus virtual. Los alumnos apenas utilizan los foros, aún prometiéndoles una bonificación en su nota. Se ha observado que los alumnos prefieren venir a consultar al despacho o en última instancia enviar un correo electrónico. Esto puede ser debido a la alta disponibilidad del personal docente y al trato directo y amigable que se respira en clase entre alumnos y docentes.

Desde el punto de vista del docente, las cosas tampoco han sido fáciles: en primer lugar, para impartir una asignatura según esta metodología, se requiere de los siguientes factores:

- Disponer de como mínimo 2 horas seguidas de clase. En caso de no ser así, los alumnos no tendrían suficiente tiempo para resolver el examen

(en el caso que nos ocupa no solía durar más de 30 minutos) y entregar las miniprácticas.

- Disponer de un campus virtual (*moodle* o similar) que permita resolver toda la logística asociada a los exámenes de forma eficiente.
- Disponer de un laboratorio bien equipado y con espacio suficiente para que el personal docente pueda recolectar fácilmente información sobre todos los alumnos.

Aun cumpliendo con estos factores imprescindibles, la mayor dificultad con la que se han encontrado los profesores es el tiempo. Como norma general, para preparar 2 horas de clase, se tenían que invertir unas 20 horas, de las cuales 16 se dedicaban a la redacción de la teoría, unas 2 a la redacción de las miniprácticas y las otras 2 restantes a la redacción del examen. Afortunadamente, el grueso del trabajo de redacción de la teoría, orientada al ABP, se llevó a cabo en la prueba piloto del curso 2009 – 2010. No obstante, en el curso actual, por lo que a contenidos teóricos se refiere, el gasto de tiempo ha sido mínimo ya que sólo se han tenido que corregir algunas erratas y actualizar puntualmente algunos contenidos.

Dentro de este cómputo, faltaría añadir las horas de trabajo asociadas a la preparación y diseño de la práctica (que cada año es distinta); lo cual ha implicado un sobrecoste añadido de 40 horas. Estas, incluyen diseño de la práctica, corrección de cada una de las cuatro fases y gestión de grupos.

Efectivamente, el sobreesfuerzo asociado a la prueba piloto fue una clara apuesta de futuro, pues toda la documentación (miniprácticas y teoría) generada, orientada al nuevo paradigma docente, ha sido reutilizable, en gran parte, para el curso académico actual. Sin lugar a duda, este sobreesfuerzo se ha visto sobradamente gratificado por la satisfacción aparente que se aprecia en los alumnos cuando están en clase.

Por último y no por ello menos importante, es necesario reparar en el número de alumnos por profesor que hay en el aula. Mientras que en el modelo magistral un único profesor puede hacer frente a prácticamente un número ilimitado de alumnos, en un entorno *Learning by Doing* esto no es posible. En el caso que nos ocupa, para un conjunto de 30 alumnos se ha requerido de un profesor titular, un becario y un colaborador.

Con esta jerarquía en el aula, el alumno percibe

una atención casi personalizada por parte del docente, aumentando así su satisfacción y sus ganas de trabajar.

Tal y como muestra en la Figura 1, los resultados son esperanzadores y se está marcando una tendencia de mejora:

- Se ha dejado atrás la cifra del 25% de alumnos que asistían a clase (en la Figura 1, cursos académicos anteriores al piloto: 2005 – 2006, 2006 – 2007 y 2008 – 2009) llegando a casi el 85% de los alumnos matriculados (en la Figura 1, cursos académicos anteriores al piloto: 2009 – 2010 y 2010 – 2011).
- Todos los alumnos, sin excepción, quieren hacer los exámenes de cada tema, teniendo un 80% de aprobados de media por tema. Esto se traduce en un aumento considerable del número de aprobados en la convocatoria ordinaria tal y como se muestra en la Figura 1.
- Los alumnos valoran muy positivamente la “nueva” manera de entregar la práctica, alegrándose de ver otras soluciones y aprendiendo de ellas. Esta motivación se aprecia en el incremento del número de prácticas entregadas en la convocatoria ordinaria (Figura 1).
- De la misma forma que en los exámenes parciales, todos los alumnos participan activamente en el desarrollo de las prácticas, incitándose entre ellos a trabajar y a mejorar día a día. Esto puede percibirse (1) de los comentarios que el personal docente escucha mientras ellos discuten sobre la práctica y (2) de las encuestas de calidad docente.
- Se ha disminuido dramáticamente el número de repetidores de la asignatura. Tal y como se aprecia en la Figura 1, se ha dejado atrás el desafortunadamente tradicional 30% para dejarlo por debajo del 8%.
- Ahora sí, **los alumnos aprenden como se desea**: de forma progresiva y adaptativa a sus capacidades. Más allá de los resultados numéricos que nos avalan, hay detalles sintomáticos: muchos alumnos, clásicos usuarios de Windows, tras terminar la asignatura toman Linux como su sistema operativo para uso *doméstico*.

Después de todo lo aquí descrito, ya sólo queda una única cosa pendiente: hacer balance del trabajo y analizar qué es lo que queda por hacer.

6. Coda

En definitiva, este artículo plasma todo el trabajo desarrollado, así como los resultados obtenidos, para migrar de un modelo magistral a un modelo basado en el aprendizaje activo, aplicado concretamente en la asignatura de SOA.

Tal y como se describe en la sección 5, los resultados de migrar de un modelo magistral a un modelo basado en el aprendizaje activo están siendo satisfactorios: se ha revertido claramente la tendencia negativa de los últimos años haciendo que los alumnos realmente disfruten tanto o más que los profesores de la asignatura. Además, este nuevo paradigma supone un reto diario para el docente ya que requiere mucha preparación y dedicación a cada sesión.

No todo el trabajo está hecho. Por un lado falta ajustar el formato de la asignatura a las últimas tendencias de la universidad, eso es: el modelo semipresencial. Por otro lado falta relacionar las competencias descritas en [4] con las actividades desarrolladas en la asignatura.

Referencias

- [1] Alcober, J.; Ruiz, S.; Valero, M. *Evaluación de la implantación del aprendizaje basado en proyectos en la EPSC (2001-2003)*, XI Congreso Universitario de Innovación Educativa en las Enseñanzas Técnicas, 2003.
- [2] Hurtado Simón, M.A.; et al. *Análisis de la incidencia de las metodologías docentes activas en los estudiantes*, XV Congreso Universitario de Innovación Educativa en las Enseñanzas Técnicas, 2007.
- [3] ANECA. *Libro blanco del título de grado en Ingeniería Informática*, Madrid 2004.
- [4] Golobardes, E.; et al., *Guía para la evaluación de competencias en el área de Ingeniería y Arquitectura*. AQU Catalunya, B-27.204-2009, 2009.
- [5] Santa, J.; Zamora, M.A.; Toledo, R. *Proceso adaptativo para la implantación de ABP en materias de sistemas embebidos*, XV JENUI, 2009.
- [6] Solomon, G. *Project-Based Learning: A primer*. Technology and Learning, 2003.
- [7] Vivaracho Pascual, C.E. *Aplicación del ABP a la parte de laboratorio (MINIX) de una asignatura de Sistemas Operativos*, XV JENUI, 2009.
- [8] Zamora, M.A.; Santa, J.; Villalba, G. *Una experiencia de transición hacia el EEES en la enseñanza de tecnología electrónica mediante nuevas técnicas docentes*, III Jornadas Sobre el Espacio Europeo de Educación Superior: 'Avanzando Hacia Bolonia', 2008.

Integración de elementos visuales y animaciones en las prácticas de programación paralela

José Miguel Mantas Daniel Guerrero Sergio Rodríguez
Departamento de Lenguajes y Sistemas Informáticos
Univ. Granada. Daniel Saucedo s/n, 18071 Granada
jmmantas@ugr.es danigrmartinez@gmail.com lumley256@gmail.com

Resumen

Dada la importancia actual del procesamiento paralelo y su complejidad, este trabajo pretende integrar el elemento visual y la animación dentro de la enseñanza y aprendizaje de la programación paralela. Con dicha orientación, se presentan las principales aportaciones de un proyecto de innovación docente que extiende el material web existente en las prácticas de una asignatura de programación paralela, dotándolo de elementos visuales dinámicos y cierto grado de interactividad para favorecer el aprendizaje. Asimismo, se introduce en las prácticas, el uso de herramientas de visualización de procesos y trazado gráfico de la ejecución que facilitan la comprensión y depuración de los programas paralelos. Las ideas presentadas pueden ser fácilmente aplicables a cualquier asignatura de programación que integre contenidos de concurrencia.

Palabras clave

Programación paralela, Visualización y animación de algoritmos paralelos, análisis y depuración de programas paralelos, Paso de mensajes, MPI.

1. Introducción

La programación paralela ha cobrado una gran importancia en los últimos años debido a la difusión y uso extendido de los procesadores multinúcleo, la computación en red y los procesadores gráficos programables. Dado que hoy en día, casi todos los sistemas de cómputo son paralelos, se le debe prestar una mayor atención a la enseñanza de la programación paralela y los algoritmos paralelos en los grados de Ingeniería informática [4]. Por otro lado, la programación paralela es mucho más compleja que la programación secuencial y este hecho se agudiza para el caso, de gran importancia, de la programación de plataformas distribuidas usando paso de mensajes [2, 14]. La dificultad de la programación paralela

se debe principalmente al comportamiento dinámico de los programas, en los que múltiples entidades de ejecución (procesos o hebras) interactúan entre sí para lograr un propósito común. La sincronización y comunicación entre estas entidades hace que la comprensión del funcionamiento de los programas paralelos y la detección de errores sean especialmente complicados.

Es un hecho aceptado que la representación gráfica favorece la comprensión de sistemas complejos, ya que el sistema visual humano está más preparado para asimilar información en formato gráfico que en formato textual. En particular, para el aprendizaje de la programación y la algoritmia, el uso de enfoques basados en la visualización y animación del comportamiento han mostrado ser muy efectivos [9, 12]. Dada la mayor complejidad que exhiben los algoritmos paralelos, la adopción de enfoques de enseñanza basados en la visualización y animación [6] resulta aún más necesaria en este ámbito que en el ámbito de la programación secuencial. La comprensión de los algoritmos paralelos puede mejorarse tremendamente usando representaciones gráficas animadas, como se pone de manifiesto en algunos trabajos previos [8, 13]. Estas representaciones abstraen los aspectos fundamentales del algoritmo (forma en que se ha descompuesto el problema, sincronización entre las unidades de trabajo, transferencia de datos, etc.) y los presentan de una forma amigable al estudiante, permitiéndole captar más fácilmente el proceso de resolución paralela del problema.

En este trabajo, se presentan las principales aportaciones de un proyecto de innovación para mejorar el material web docente existente en las prácticas de una asignatura de programación paralela, dotándolo de elementos visuales dinámicos y cierto grado de interactividad.

La asignatura objeto de mejora pertenece al segundo ciclo de la Ingeniería Informática y aborda el diseño, análisis e implementación de algoritmos paralelos y distribuidos. En las prácticas, los estu-

diantes aprenden a programar y analizar algoritmos paralelos y distribuidos usando la interfaz de paso de mensajes MPI [2, 14] y en menor medida programan algoritmos paralelos basados en memoria compartida usando directivas y funciones de OpenMP [2, 5]. Se plantean varios guiones de prácticas como material de prácticas. En algunos guiones, el alumno trabaja con un tutorial web que sirve de introducción a la notación particular a usar (MPI u OpenMP). En estos tutoriales, se plantean problemas sencillos de programación paralela (ordenación, cuadratura numérica, cálculos de diferencias finitas, etc.) para que el alumno intente resolverlos, y más adelante se muestra el programa solución. El resto de guiones plantean proyectos de implementación paralela de pequeña envergadura (por ejemplo, implementación paralela de algoritmos de grafos). Estos guiones describen de forma esquemática un algoritmo paralelo para resolver el problema, y el alumno debe completar el desarrollo del proyecto incluyendo el análisis del rendimiento de su implementación paralela.

A lo largo de varios cursos, se han ido detectando problemas de aprendizaje en las prácticas, debido tanto a la dificultad para entender las funciones de paso de mensajes y la dinámica de los algoritmos distribuidos, como al arduo proceso de detección y corrección de errores en los programas paralelos. Para mitigar estos problemas, se propuso el proyecto de innovación docente cuyas aportaciones se presentan en este trabajo.

La sección 2 presenta el enfoque general seguido para mejorar las prácticas de programación paralela, mientras que los aspectos más relevantes de la implementación particular que se ha hecho de las ideas generales se describirá en la sección 3.

2. Enfoque general para introducir elementos visuales en asignaturas relacionadas con la programación paralela

El enfoque que se ha seguido para mejorar las prácticas (véase la Figura 1) está basado en las tres líneas de actuación que se presentan a continuación.

- Creación en la web de tutoriales, ayudas y guiones de prácticas que incluyan gráficos y animaciones interactivas. Estas animaciones describen los pasos principales de los algoritmos paralelos y la funcionalidad de las funciones sub-

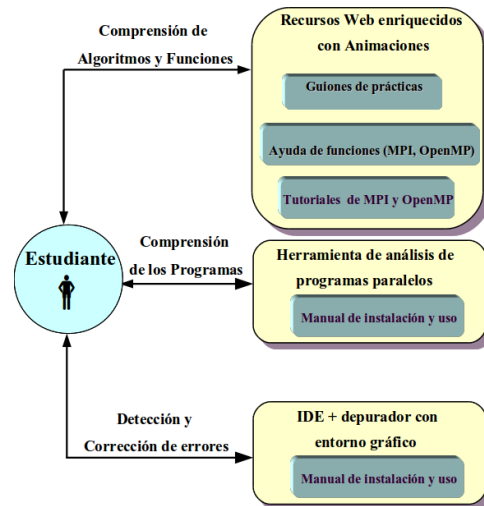


Figura 1: Visión general del enfoque propuesto

yacentes.

- Fomentar y apoyar en las prácticas el uso de herramientas de análisis y visualización de procesos que permitan comprender la interacción entre procesos en programas distribuidos.
- Adoptar el uso en prácticas de un entorno de desarrollo que integre una herramienta de depuración paralela con una alta componente visual.

2.1. Inserción de Animaciones en Tutoriales y guiones de prácticas

Entender cómo funciona un algoritmo puede ser complicado, si además este debe funcionar parcial o totalmente de forma distribuida, la situación se vuelve aún más difícil. Generalmente el problema se encuentra asociado a la falta de entendimiento sobre qué parte del trabajo realiza cada proceso participante y, especialmente, en qué puntos se comparten los datos y qué datos son estrictamente necesarios para el correcto funcionamiento del algoritmo. En ocasiones, dichos datos se transmiten usando operaciones de comunicación colectivas (transferencia de uno a muchos, reparto de uno a muchos, recolección de muchos a uno, etc.) pensadas específicamente para simplificar la programación sin mermar el rendimiento del programa. Explicar este tipo de problemas claramente usando sólo un formato tex-

tual es complicado y no asegura el adecuado aprendizaje del estudiante.

Desde nuestro punto de vista, un desarrollador nuevo en el mundo del paralelismo carece de una estructura firme sobre la que recrear el funcionamiento de algoritmos distribuidos. Para solventar las dificultades de comprensión planteadas, recurrimos al dicho "más vale una imagen que mil palabras", y si son imágenes en movimiento que describan la naturaleza dinámica de un algoritmo paralelo, mejor.

En el enfoque propuesto se combinan dos aproximaciones diferentes, usando imágenes estáticas introductorias junto con animaciones para dar una descripción más detallada. La imagen representativa puede describir de forma abstracta el funcionamiento del algoritmo u operación relacionándolo con un entorno cotidiano. Usar una imagen de estas características permite reducir el tiempo dedicado al aprendizaje, y además, una vez aprendida la finalidad, el usuario tan solo deberá contemplar de nuevo la imagen para recordarla en un futuro.

Las animaciones, en cambio, tienen como finalidad mostrar los puntos clave en el tiempo de cada algoritmo u operación, así como los datos más relevantes que participan. Para facilitar el aprendizaje a largo plazo, todas las animaciones deberían usar el mismo modelo. De esta forma un usuario inexperto, tras visualizar unas pocas animaciones, ya se encontraría familiarizado con los conceptos básicos de cada una de ellas y esto también le ayudaría para entender situaciones futuras.

Un ejemplo sencillo de explicación para el modelo gráfico es el caso de una operación de difusión de datos entre procesos. En esta operación, un único proceso envía exactamente los mismos datos a todos los procesos de un grupo determinado. La Figura 2 presenta una imagen que podría representar esta operación y la Figura 3 muestra algunos fragmentos de una animación descriptiva de la misma.

2.2. Herramientas de visualización de procesos en programas basados en paso de mensajes

Cuando nos proponemos validar, analizar o comprender un programa paralelo, la tarea se vuelve más complicada que en un programa secuencial. Este análisis no es tan simple como hacer un seguimiento de la traza de una ejecución secuencial. En el caso de un programa paralelo de paso de mensajes, la

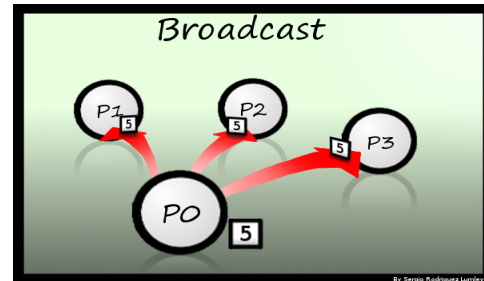


Figura 2: Imagen representativa de una difusión

existencia de múltiples procesos interactuando entre sí con operaciones de transferencia de datos, algunas de ellas bastante complejas, hace que analizar dichos programas sea una tarea ardua.

Una forma de abordar el análisis de programas basados en paso de mensajes consiste en usar herramientas de análisis del rendimiento de programas distribuidos [10, 15]. Entre otras posibilidades, estas herramientas permiten generar la información necesaria durante la ejecución de los programas para poder procesarla e interpretarla después de forma gráfica. Para visualizar la información recopilada por la herramienta, se hace uso de aplicaciones específicas para visualizar el comportamiento de un programa distribuido [3, 17]. Estas aplicaciones generan representaciones gráficas de los procesos donde las flechas denotan el paso de mensajes.

Los análisis visuales que permiten este tipo de herramientas favorecen la comprensión de los programas paralelos a los estudiantes que se inician en esta materia.

2.3. Uso de entornos de desarrollo con extensiones para depuración multiproceso

Para cualquier principiante en programación, disponer de la ayuda de un Entorno de Desarrollo Integrado (IDE) sobre el que poder trabajar de forma centralizada es una gran ventaja. Con estas herramientas a disposición del programador, el desarrollo de programas paralelos se facilita sustancialmente. El usuario puede dedicarse enteramente al desarrollo de aplicaciones sin preocuparse por detalles sobre compilación, dependencias y, en algunos casos, pueden recibir ayuda sobre errores sintácticos. A la hora de encontrar errores en la ejecución, los

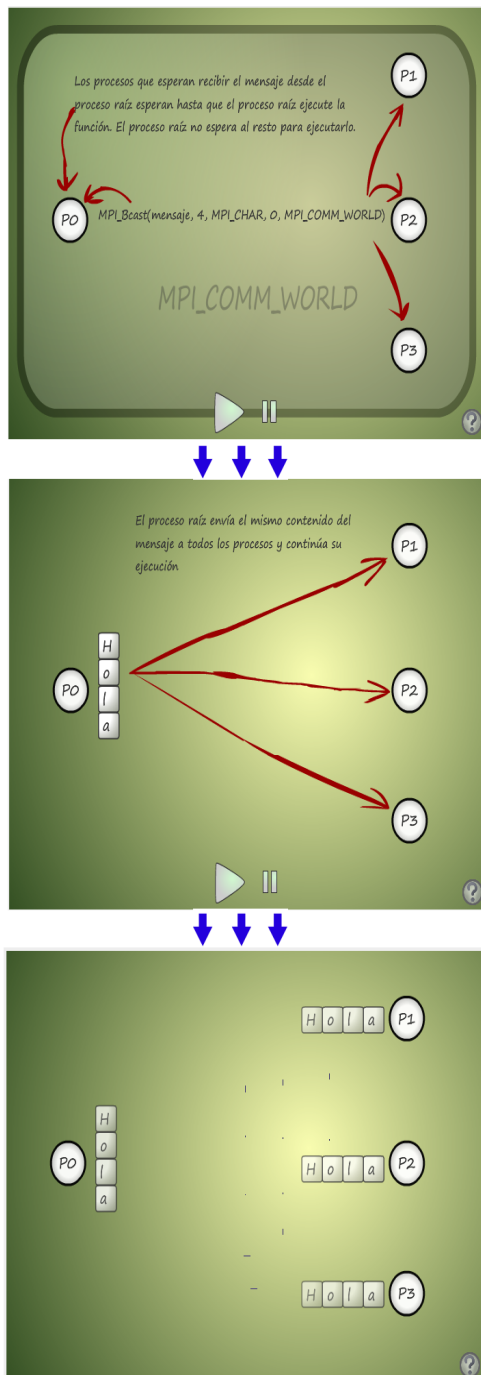


Figura 3: Animación para la difusión de uno a muchos

depuradores se convierten en una herramienta primordial, y cuando hablamos de depurar programas paralelos, más aún. Es muy común entre los programadores utilizar la salida por consola como método de depuración simple. Sin embargo esto no es lo más aconsejable en programas paralelos, puesto que intervienen múltiples procesos o hebras que avanzan de forma simultánea. Sin embargo, la mayoría de los depuradores tienen un problema añadido para principiantes, ya que suelen funcionar por consola y el usuario debe aprender a manejarlo. Por consiguiente, para facilitar lo máximo posible la tarea de depuración de programas paralelos, disponer de un depurador con representación gráfica es esencial.

Para que entorno de desarrollo y el depurador integrado sean útiles para nuestros propósitos, el estudiante debe poder visualizar algún tipo de representación gráfica de las unidades de ejecución, ya sean hebras y/o procesos, durante el trazado de la ejecución del programa paralelo, pudiendo consultar el estado de cada unidad de forma independiente en diferentes puntos de la ejecución.

3. Implementación particular en una plataforma web de una asignatura de programación paralela

El enfoque propuesto en la sección anterior ha sido implementado dentro de la página web <http://lsi.ugr.es/~jmantas/pdp>, dirigida a alumnos con un nivel considerable de programación secuencial, pero con poca experiencia en programación paralela. La web se encuentra centrada principalmente en la enseñanza práctica de la programación paralela mediante los métodos anteriormente descritos, incluyendo los siguientes recursos

- Animaciones dentro de los guiones prácticos.
- Un completo tutorial de MPI que describe los problemas planteados con ayuda de animaciones, así como un tutorial de OpenMP.
- Documentación web sobre MPI y OpenMP necesaria para abordar las prácticas, incluyendo animaciones para describir funciones de MPI.
- Guías de instalación en Linux y manuales de uso de las principales herramientas usadas. Dentro de las herramientas se incluyen:

1. Una Herramienta de análisis y visualización de programas paralelos, denominada

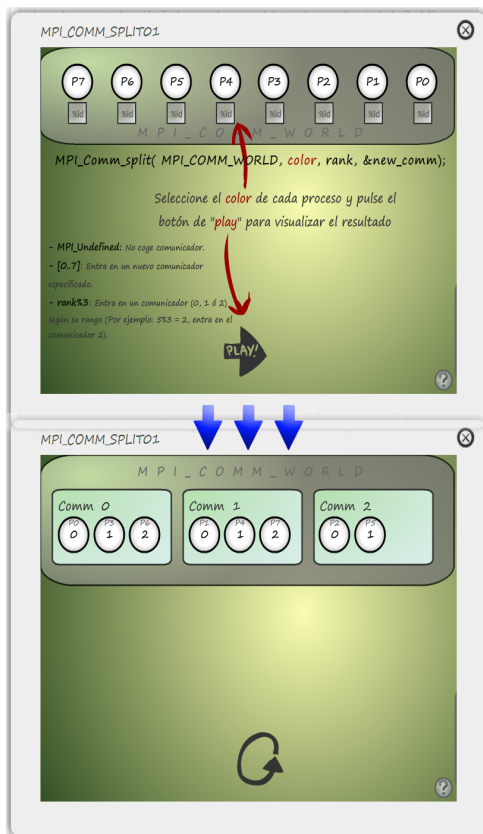


Figura 4: Ejemplo de animación Flash interactiva

TAU [15].

2. El IDE *Eclipse* y su depurador para programas paralelos.

3.1. Diseño de las animaciones

Siguiendo la filosofía propuesta, todas las animaciones se introducen mediante una imagen que ilustra del funcionamiento descrito (véase un ejemplo en la Figura 2). Si el usuario necesita aún más información, pulsando sobre la imagen se desplegará la animación (véase la Figura 3). Esta forma de funcionar permite que un mayor número de dispositivos vean la web correctamente y también permite evitar la carga de la animación cuando no se dispone de conexiones rápidas. Para el diseño de animaciones hemos recurrido al ampliamente utilizado *Flash* [1, 11], lo que permite que sean visualizadas por prácticamen-

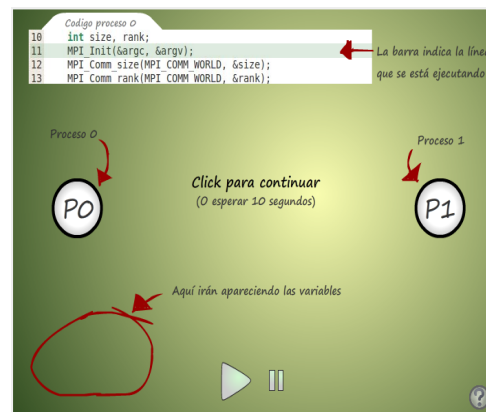


Figura 5: Ejemplo de animación Flash cinemática

te cualquier navegador actual, además de poder dotar a las animaciones de cierto grado de interactividad. Todas las imágenes y animaciones siguen el mismo modelo para lograr una mayor adaptación del usuario y permitirle centrarse antes en el contenido que en el funcionamiento de la propia animación. Para el caso de las animaciones, siempre siguen la misma secuencia: una secuencia de introducción animada, presentación del cuadro de ayuda y ocultación del mismo, y reproducción del resto de la animación (véase la Figura 4).

Cuando se presenta texto en pantalla, se muestran controles del flujo de la presentación y las opciones disponibles son: pulsar sobre la pantalla, para continuar con la siguiente secuencia; pulsar sobre pausa para detener la animación; o no hacer nada, en cuyo caso la animación continuará cuando pase tiempo suficiente (véase la Figura 5).

Las animaciones para las funciones muestran su funcionamiento en un orden temporal, destacando el uso de los parámetros (véase la Figura 3). Las animaciones para los tutoriales muestran qué debe hacer el programa que resuelve el problema planteado y algunos detalles sobre cómo lograrlo. Como muestra, la Figura 6 esboza un algoritmo distribuido de ordenación Mergesort. En las animaciones para prácticas, se explica cuál es el comportamiento que se debe lograr, utilizando representaciones similares a las ya presentadas, para que el usuario pueda reconocerlas fácilmente. Así, en la Figura 7 se muestra el marco de una animación que describe una implementación paralela del algoritmo de Floyd para bus-

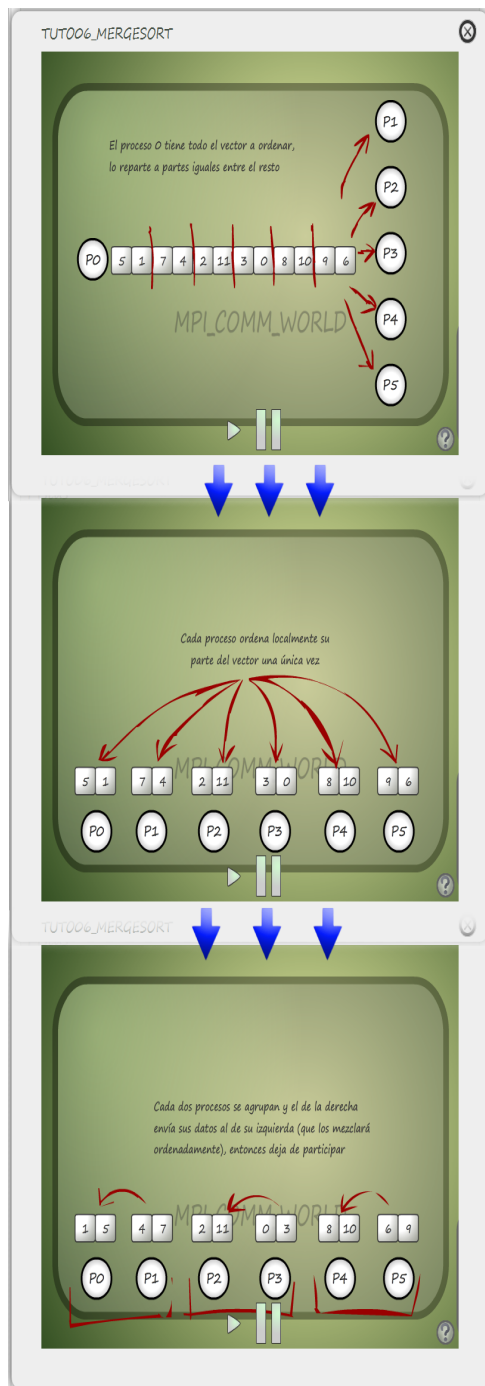


Figura 6: Ejemplo de animación en tutorial

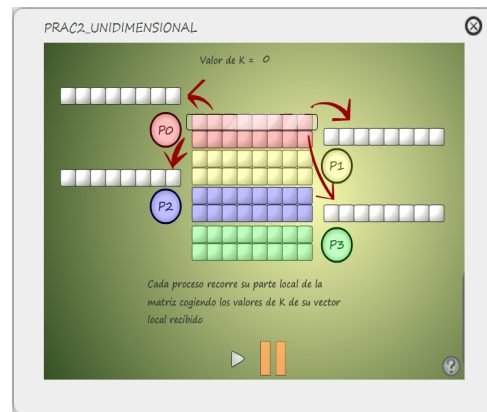


Figura 7: Ejemplo de animación en guión de prácticas

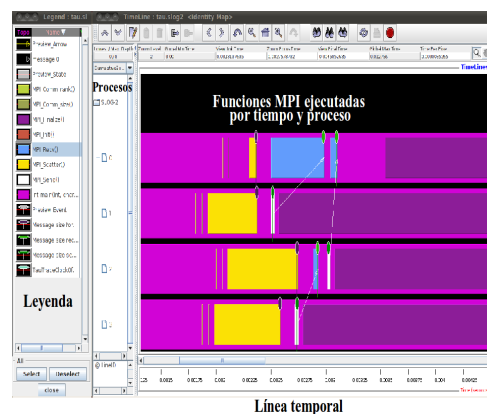


Figura 8: Salida de la herramienta TAU

car los caminos más cortos en un grafo.

3.2. Integración de una herramienta de análisis de programas de paso de mensajes

Para facilitar su uso a los estudiantes, se ha optado por implantar un software gratuito de análisis de programas MPI denominado TAU (Tuning and Analysis Utilities). TAU es un conjunto de utilidades que nos permiten analizar programas paralelos escritos en lenguaje Fortran, C, C++, Java y Python. TAU nos permite hacer un análisis en profundidad de los programas paralelos de forma parecida a los analizadores convencionales, además de realizar un análisis del trazado de las comunicaciones entre procesos. Los resultados de dichos análisis pueden ser

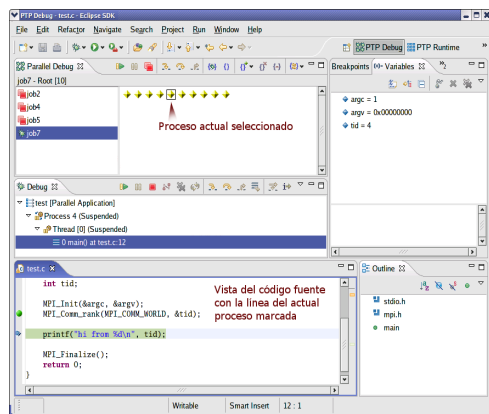


Figura 9: Depuración con Eclipse y PTP

visualizados de forma gráfica gracias a herramientas como *Paraprof* [17], destinada a realizar estadísticas del análisis, y *Jumpshot* [3] que muestra el trazado de las funciones de comunicación entre los procesos en una línea temporal.

El uso de TAU para visualización es sencillo. Se usa un script específico para compilación y enlace del programa MPI que genera un ejecutable extendido. Al lanzar este ejecutable, además de la salida normal del programa, se guardan datos en ficheros durante su ejecución. Estos ficheros pueden ser unificados y visualizados con *Jumpshot*. La Figura 8 muestra la visualización obtenida para un programa MPI sencillo. En la parte inferior se muestra una línea temporal y los diferentes procesos aparecen como barras horizontales divididas en franjas de diferentes colores que pueden hacer referencia a la ejecución de diferentes funciones de MPI (tal como indica la leyenda que aparece en el lado izquierdo). Los puntos de comunicación entre procesos se indican claramente con elipses y flechas que unen diferentes instantes de tiempo en la representación gráfica de los procesos.

3.3. IDE con depurador paralelo

Se ha optado por un IDE lo suficientemente sencillo de usar, extensible y gratuito, por lo cual el popular IDE *Eclipse* [7] cumplía las características buscadas para nuestros propósitos, además de ser de código abierto. A través del sistema de extensiones del que dispone, Eclipse cuenta en su repertorio

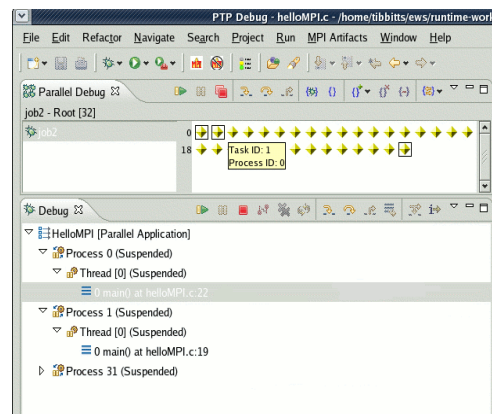


Figura 10: Control de hebras con un IDE

con la mejor de las opciones para un desarrollador de programas distribuidos, un depurador gratuito, libre, gráfico y con la capacidad de controlar programas distribuidos, hablamos de la extensión *Parallel Tools Platform (PTP)* [16]. Esta herramienta permite depurar programas divididos en varios procesos. En tiempo de ejecución, el programa permite seleccionar cualquiera de los procesos y controlar su ejecución mostrando el depurador en qué línea de código se encuentra el proceso, pausar el resto de procesos o hacer que avancen hasta el mismo punto que el proceso seleccionado. La Figura 9 muestra el IDE durante el proceso de depuración, distinguiéndose el proceso seleccionado y la línea de código que está ejecutando.

La herramienta es incluso capaz de controlar hebras dentro de cada proceso, por lo que está indicada también para trazar programas paralelos que usan OpenMP, como se muestra en la Figura 10.

3.4. Evaluación

Aunque no se ha tenido tiempo de realizar una evaluación rigurosa, se ha pasado a los alumnos un cuestionario para detectar posibles deficiencias del sistema implantado. La Tabla 3.4 resume los resultados obtenidos (la máxima valoración es 5) y refleja en general una buena aceptación del uso de las animaciones. La implantación de las herramientas de análisis y depuración ha sido parcial ya que se pretende ir introduciendo los nuevos elementos paulatinamente en dos cursos académicos. No obstante, los estudiantes que han optado por usar Eclipse con el

Pregunta	Med.
Calidad y efectividad de las guías de ayuda	4,2
Comprensión del texto en ayuda de MPI	4,2
Efectividad de imágenes en la ayuda de MPI	4,4
Efectividad de animaciones en la ayuda de MPI	4,5
Comprensión del texto en la ayuda de OpenMP	3,7
Comprensión del texto en tutoriales y prácticas	4,2
Efectividad de imágenes en tutoriales y prácticas	4,4
Efectividad de animaciones en tutoriales y prácticas	4,5

Cuadro 1: Resumen de resultados del cuestionario

plugin PTP se han mostrado muy satisfechos con los beneficios obtenidos tanto en la detección de errores como en la comprensión de la ejecución paralela.

4. Conclusiones y Trabajos Futuros

En este trabajo se proponen tres tipos de actuaciones para mejorar el aprendizaje de la programación paralela. Estas ideas están inspiradas en enfoques basados en la animación y visualización y se han implementado en la página web de una asignatura de programación paralela. La combinación de animaciones para describir el comportamiento de los algoritmos paralelos, junto con el uso de herramientas de análisis, visualización y depuración gráfica de programas paralelos, se muestra como una vía adecuada para mejorar el aprendizaje de los aspectos más complejos de la programación paralela. Además estas ideas se pueden implantar fácilmente en asignaturas de fundamentos de programación y algoritmia en las que se decida impartir contenidos sobre computación y algoritmos paralelos.

Actualmente, se está trabajando en la extensión del trabajo con las siguientes mejoras:

- Insertar animaciones para enriquecer el material web dedicado a OpenMP y extender el material web con recursos para el aprendizaje de la programación de GPUs.
- Desarrollar una aplicación web interactiva que permita la edición de imágenes y animaciones, siguiendo las líneas guía ya introducidas.

Agradecimientos

El trabajo se ha financiado con los proyectos 09-162 de la Univ. de Granada (Innovación Docente) y MTM2008-06349-C03-03 (MEC).

Referencias

- [1] Adobe Flash CS4 Professional. www.adobe.com/es/products/flash.
- [2] Almeida F., Giménez D., Mantas J. M., Vidal A. M. *Introducción a la programación paralela*. Paraninfo Cengage Learning, 2008.
- [3] Bell R., Malony A. D., Shende S. *ParaProf: A Portable, Extensible, and Scalable Tool for Parallel Performance Profile Analysis*. Europar 2003, LNCS 2790 pp.17-26, 2003.
- [4] Giménez D., Almeida F., Mantas J. M., Vidal A. M. *Sobre la situación del paralelismo y la programación paralela en los Grados de Ingeniería Informática*. ReVisión, Vol 3, N 1, 2010.
- [5] Chapman B., Jost G., van der Pas R. *Using OpenMP. Portable Shared Memory Parallel Programming*. The MIT Press, 2007.
- [6] Gómez-Albarrán M. *Una revisión de métodos pedagógicos innovadores para la enseñanza de la programación*. JENUJ 2003 pp.: 363–370.
- [7] IDE Eclipse. <http://www.eclipse.org/>.
- [8] Johnson D., Kotz D., Makedon F. *Teaching Parallel Computing to Freshmen*. Conference on Parallel Computing for Undergraduates 1994.
- [9] Mandow L., Villalba F., Coego J. *Uso de animaciones para la enseñanza de algoritmos de búsqueda en Inteligencia Artificial*. JENUJ 2010 pp.: 461–468.
- [10] MPE. www.mcs.anl.gov.
- [11] Mook C. *ActionScript : The Definitive Guide*. O' Reilly Media 2001.
- [12] Pérez A., Velázquez J. A. *Animación automatizada de técnicas de diseño de algoritmos*. JENUJ 2009 pp.: 115–122.
- [13] Rantakokko J. *Algorithm Visualization through Animation and Role Plays*. Proc. of 3rd Progr. Visualization Workshop, 2004. pp.: 76–81.
- [14] Snir M., W. Gropp. *MPI. The Complete Reference. 2nd edition*. The MIT Press, 1998.
- [15] TAU. Tuning and Analysis Utilities. www.cs.uoregon.edu/research/tau.
- [16] Watson G., Rasmussen C., Tibbitts. B. *An integrated approach to improving the parallel application development process*. IEEE ISDPDP 2009, pp.1-8, 2009.
- [17] Wu C. E., et. al. *From Trace Generation to Visualization: A Performance Framework for Distributed Parallel Systems*. Proc. of SC2000.

Sesión 3B:
Organización curricular y planes de estudio
Informática en otras carreras

Organización y gestión de una titulación del EEES

Fermín Sánchez
Dept. de Arquitectura de
Computadores
U. Politècnica de Catalunya
Campus Nord, Mòdul D6
08034 Barcelona
fermin@fib.upc.edu

Maria-Ribera Sancho
Dept. de Ingeniería de Servicios y
Sistemas de Información
U. Politècnica de Catalunya
Campus Nord, Edif. Omega
08034 Barcelona
ribera@fib.upc.edu

José R. Herrero
Dept. de Arquitectura de
Computadores
U. Politècnica de Catalunya
Campus Nord, Mòdul C6
08034 Barcelona
josepr@fib.upc.edu

Resumen

La organización de los nuevos grados y másteres del EEES debe garantizar que el estudiante adquiere las competencias técnicas y transversales definidas por la titulación. Para ello, deben existir mecanismos muy precisos de coordinación entre asignaturas que aseguren la adquisición de estas competencias, y que garanticen que la organización de los estudios permite la evaluación continuada y el aprendizaje activo. Para alcanzar estos objetivos, es imprescindible ofrecer al profesorado las herramientas apropiadas para que diseñe y ponga en marcha las asignaturas siguiendo los objetivos del centro.

Summary

The organization of new bachelor and master degree studies within EHEA must guarantee the student's achievement of hard and professional skills defined in each degree. Consequently, coordination mechanisms must exist. Such mechanisms must ensure the achievement of skills and warrant that both, active learning and continuous assessment are possible. In order to achieve these goals, it is necessary to offer appropriate tools to the teachers so that they can design and implement courses according to the objectives of the institution.

Palabras clave

Organización curricular, coordinación de titulación, coordinación de asignaturas, gestión de una titulación, diseño de un plan de estudios, adquisición de competencias, herramientas de coordinación.

1. Introducción

La puesta en marcha de las nuevas titulaciones de máster e ingeniería del EEES requiere un gran nivel organizativo. Estas titulaciones deben haber sido diseñadas a partir de competencias profesionales y han de estar organizadas para que el estudiante adquiera dichas competencias. A diferencia de las titulaciones anteriores al EEES, en las que el estudiante cursaba un conjunto de asignaturas más o menos relacionadas entre sí que constituían su formación, las nuevas titulaciones deben haber sido diseñadas de forma integral a partir de una lista de competencias que el estudiante debe haber adquirido al finalizar sus estudios. La organización de los nuevos grados y másteres debe, por lo tanto, garantizar que el estudiante adquiere estas competencias. Para ello, es preciso diseñar la titulación desde un punto de vista global, y no como un conjunto de asignaturas independientes más o menos relacionadas.

En [3,4,7,8] se describe una metodología de diseño de un plan de estudios que parte de una lista de competencias de la titulación y un conjunto de restricciones y criterios de diseño. El plan de estudios se concibe como una matriz de "cajas negras" (las futuras asignaturas) de distintos créditos, concebida a partir de la lista de restricciones y criterios. Sobre los elementos de esta matriz se distribuye la lista de competencias, teniendo en cuenta el conjunto de restricciones y criterios [6] e indicando el nivel al que debe trabajarse cada competencia dentro de cada asignatura. A continuación, se asignan las asignaturas a los responsables de diseñarlas y ponerles nombre.

Para que este diseño funcione, es preciso disponer de mecanismos de coordinación muy

precisos que garanticen la cooperación entre los diseñadores de las diferentes asignaturas. Esta coordinación debe permitir que las competencias de la titulación sean convenientemente tratadas en las asignaturas a lo largo del plan de estudios, de forma que no haya repeticiones (más allá de las necesarias) ni lagunas en la formación de los alumnos [6]. El diseño del plan de estudios garantiza, de este modo, la correcta distribución de las competencias técnicas y transversales definidas por la titulación. Para que los estudiantes adquieran las competencias de una forma razonable es imprescindible definir una cierta precedencia entre las asignaturas, de forma que los estudiantes puedan matricular en cada momento únicamente aquellas asignaturas que les permitan adquirir nuevas competencias o mejorar en las competencias ya adquiridas. Una forma de conseguirlo es definir el plan de estudios como un grafo dirigido en el que las asignaturas son los nodos y las aristas las relaciones que definen el orden de matrícula (pre-requisitos). En cada momento, un estudiante sólo puede matricular aquellas asignaturas en las que todas las aristas entrantes llegan de asignaturas que el estudiante ya ha cursado (se puede exigir que las haya aprobado). Otro mecanismo, aunque más restrictivo, consiste en agrupar las asignaturas en bloques y definir las conexiones a nivel de bloque. El sistema de avanzar por curso es un ejemplo de este esquema (un bloque es un curso, un alumno matricula un curso completo y no pasa al siguiente curso hasta haber aprobado el curso anterior).

Las asignaturas deben estar orientadas al aprendizaje activo, de forma que se realice una evaluación continuada. Para alcanzar estos objetivos, es imprescindible ofrecer al profesorado las herramientas apropiadas para que diseñe y ponga en marcha las asignaturas siguiendo los objetivos del centro, y es necesario que los profesores tengan formación en estrategias docentes de aprendizaje activo. Las asignaturas no pueden diseñarse de forma independiente unas de otras. El estudiante cursará varias asignaturas simultáneamente, y la carga de trabajo de las mismas debe estar distribuida de forma que el conjunto sea razonable y no se pida al alumno una dedicación excesiva en momentos puntuales (como pasa con las actuales titulaciones cuando se acerca el final del curso).

Para poner en marcha el plan de estudios es necesario mantener e incluso mejorar todos los

mecanismos de coordinación previamente descritos. Es necesario hacer un seguimiento exhaustivo de la puesta en marcha del plan de estudios, especialmente en los primeros años, para detectar y corregir posibles desajustes, midiendo el grado de consecución de las competencias en cada asignatura y la carga real de trabajo del estudiante. Muy probablemente, las asignaturas necesitarán pequeños ajustes para que estos parámetros se acerquen lo más posible a los previstos en el diseño inicial. También es probable que el diseño inicial del plan de estudios no sea perfecto y sea necesario modificarlo a medida que el centro adquiere experiencia en su puesta en marcha. Cuando los primeros egresados salgan al mercado laboral, es imprescindible valorar en qué medida han adquirido las competencias de la titulación y, a partir de los resultados, realizar los cambios oportunos en el plan de estudios.

Finalmente, los planes de estudios no son estáticos, y mucho menos en el mundo de las TIC. La tecnología avanza muy rápido y aparecerán nuevas competencias que deben ser incorporadas a los planes de estudios. El centro y el plan de estudios deben disponer de mecanismos que permitan modificar ágilmente las asignaturas para incorporar estas nuevas competencias.

La Facultat d'Informàtica de Barcelona (FIB) realizó en el año 2003 una revisión del plan de estudios, actualizándolo y adaptándolo al EEES. Fruto de la experiencia adquirida durante ese proceso, el plan de estudios de Grado en Ingeniería Informática se ha diseñado siguiendo el modelo descrito en esta sección. Las secciones posteriores muestran, paso a paso, las decisiones de diseño que se han tomado y la forma en que se han coordinado. La Sección 2 presenta el esquema del plan de estudios diseñado y algunas de las restricciones y criterios definidos, la Sección 3 detalla la distribución temporal de trabajo del estudiante, la Sección 4 explica los mecanismos de coordinación, la Sección 5 describe la guía docente que vehicula el diseño de las asignaturas y la Sección 6 presenta las conclusiones.

2. Estructura del plan de estudios

La estructura final del plan de estudios estuvo marcada por un conjunto de decisiones iniciales ligadas, fundamentalmente, a nuestra visión de la profesión de ingeniero informático y a la

estrategia a seguir como centro de referencia en la impartición de estos estudios en Catalunya.

En primer lugar, el Grado en Ingeniería Informática se desarrolló a partir de las recomendaciones elaboradas por el Consejo de Universidades (CU) [1] para situar a nuestros titulados en la mejor posición de cara a la obtención de las correspondientes atribuciones profesionales que pudieran ligarse a este grado en el futuro.

La segunda decisión de partida fue diseñar una sola titulación con cinco especialidades, motivada por criterios de eficiencia (viabilidad, oportunidad y estrategia) y también por los acuerdos tomados en el seno de la CODDI-cat y de la CODDI.

Estas dos decisiones iniciales restringieron las posibles alternativas en cuanto a la estructura y contenido del plan de estudios. Aún así, existían condicionantes adicionales que provenían del marco de referencia definido por nuestra universidad y otros que se derivaron de los trabajos de las comisiones de nuestro propio centro.

La estructura final del plan de estudios de Grado en Ingeniería Informática se muestra en la Figura 1. Q_x representa el cuatrimestre x .

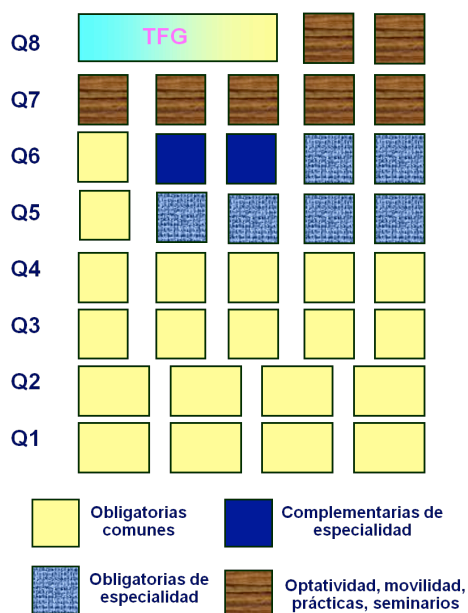


Figura 1: Estructura del plan de estudios

Como puede observarse, la obligatoriedad se organiza en tres bloques más el Trabajo Final de Grado (TFG). Un primer bloque de 60 créditos, que se cursan en el Q1 y el Q2, responde a los requisitos de formación básica impuestos por las recomendaciones del CU. Estos créditos constituyen la Fase Inicial del grado y los estudiantes deben superarla en un tiempo máximo, establecido por nuestra universidad. Durante el primer año de estudios se programan solamente cuatro asignaturas por cuatrimestre. Esta decisión formaba parte de los criterios de diseño, y el objetivo es reducir la dispersión del estudiante en los primeros cursos.

El segundo bloque lo constituyen 72 créditos que dan respuesta a las exigencias de formación común para la ingeniería informática. De estos créditos, 60 se cursan entre el Q3 y el Q4, 6 créditos en el Q5 y otros 6 en el Q6.

El tercer bloque, de 48 créditos, responde a las exigencias en cuanto a Tecnología Específica (especialidades) de las recomendaciones del CU. Se ha organizado cada especialidad en 36 créditos básicos, que cursan todos los alumnos de la especialidad, y 12 créditos complementarios que, aún siendo obligatorios, permiten al alumno escoger en cuáles de las competencias propias de su especialidad desea profundizar, seleccionando dos asignaturas de entre un reducido conjunto. En los 36 créditos comunes se incluye una asignatura de proyecto enfocada a las competencias propias de la especialidad. Los créditos de especialidad se cursan entre los cuatrimestres Q5 y Q6.

El último bloque obligatorio lo constituye el TFG, al que se han asignado 18 créditos. La decisión relativa al número de créditos dedicada al TFG no fue fácil. Se buscó un compromiso entre la importancia indiscutible de este trabajo en la formación de un ingeniero y la compatibilidad del mismo con la movilidad internacional y las prácticas en empresa. Detallaremos estos aspectos más adelante, en esta misma sección. Está todavía pendiente el decidir si se dedica una parte de estos 18 créditos a formación específica en gestión de proyectos. El formato y evaluación del TFG se está discutiendo en estos momentos siguiendo las directrices de la Guía para la Evaluación de competencias en los Trabajos de Fin de Estudios de las Ingenierías de la AQU [2] y el MICINN [5, 9,10].

La optatividad del plan de estudios se estructura en 42 créditos, que se cursan entre los cuatrimestres Q7 y Q8. La distribución de estos créditos responde a dos objetivos que se consideraron prioritarios: la internacionalización y las prácticas empresariales.

Nuestro centro considera imprescindible facilitar y promover los intercambios internacionales de nuestros alumnos y su participación en acuerdos de doble titulación. Disfrutamos de una larga tradición en este aspecto y la consideramos estratégica. En consecuencia, el plan de estudios debía dar cabida, en forma de créditos optativos, a esta opción internacional. Nuestros alumnos podrán cursar en universidades extranjeras uno o dos cuatrimestres, dependiendo de si la movilidad incluye o no el TFG y de si van a obtener una doble titulación.

Otra opción que debía ser facilitada a los estudiantes y convenientemente organizada son las prácticas en empresa. El alumno puede cursar hasta 12 créditos optativos de prácticas en empresa. Se permiten tres posibilidades, dependiendo de la intensidad con que el alumno quiera integrarse en el entorno empresarial: dedicar los 12 créditos optativos (20 semanas, 3 horas al día), desarrollar el TFG en una empresa (20 semanas, 5 horas al día), o ambas cosas (dedicación completa durante un cuatrimestre). Como puede observarse, esta estructura es mucho más ambiciosa que los actuales convenios de cooperación educativa, se integra perfectamente en el currículo del estudiante y resulta mucho más atractiva para las empresas que participan. También permite un mejor control y nivel de exigencia en las prácticas y proyectos realizados.

Finalmente, un estudiante que no participe ni en la opción internacional ni en la empresarial puede elegir entre un conjunto de asignaturas optativas de interés general, que se han estructurado en itinerarios temáticos de unos 18 créditos cada uno. También tiene a su disposición como optatividad las asignaturas de las especialidades que no son la suya propia y las complementarias de su especialidad que no ha cursado. El plan de estudios también contempla la posibilidad de programar seminarios, asignaturas optativas de 1-2 créditos que pueden impartirse también de forma intensiva al principio o al final del cuatrimestre.

Las competencias de la titulación, tanto técnicas como transversales, se han asignado a las

diferentes asignaturas de forma que para cada asignatura sabíamos, antes de comenzar su diseño, qué competencias debía trabajar y a qué nivel. Se definieron 3 niveles de profundidad para cada competencia. Con respecto a las competencias transversales, se decidió que no habría asignaturas específicas. Cada asignatura del plan de estudios tiene la misión de trabajar y evaluar una o dos competencias transversales, que han sido asignadas a las asignaturas de forma que se establezcan itinerarios transversales que permitan al estudiante trabajar cada competencia durante toda la carrera. Finalmente, para visualizar la importancia que se concede a las competencias transversales en el plan de estudios, cada asignatura tiene que poner, además de la nota final de la asignatura, una nota a cada una de las competencias transversales que trabaja, en el rango discreto A-D (A indica que se ha adquirido con excelencia, B con el nivel deseado, C de forma suficiente y D que no se ha adquirido).

3. Organización temporal

En el Grado en Ingeniería Informática de la FIB, el número de créditos asignados a las asignaturas de los dos primeros cuatrimestres, que llamamos Fase Inicial, es de 7,5 ECTS. En las asignaturas del resto de cuatrimestres, fuera de la Fase Inicial, es de 6 ECTS. En el marco de los estudios de grado de la FIB esto corresponde a un máximo de 5 y 4 horas de clase por semana respectivamente. El resto del tiempo dedicado por el alumno a la asignatura corresponde a otras actividades dirigidas y actividades de aprendizaje autónomo. Nuestros estudios de grado están pensados para que un estudiante curse 4 asignaturas mientras está en la Fase Inicial, y 5 una vez superada la misma, como se describe en la Sección 2. Por tanto, un estudiante tiene en cualquier caso un máximo de 20 horas de clase por semana.

Atendiendo al criterio de cuándo se puede aprobar una asignatura, podemos clasificar las asignaturas en dos tipos: las que permiten aprobar en la semana 15 mediante el seguimiento satisfactorio de una evaluación continua durante el cuatrimestre y las que, con independencia de la existencia de otras pruebas adicionales de evaluación, sólo se pueden aprobar tras realizar un examen final una vez acabado el periodo de clases (las semanas 16 a 18 están dedicadas a exámenes finales). Para repartir el esfuerzo de los

estudiantes a lo largo del cuatrimestre, se ha intentado que haya asignaturas de los dos tipos en cada cuatrimestre. Para evitar que un estudiante tenga que concentrar toda su dedicación en 15 semanas, se ha fijado un máximo número de asignaturas que se pueden aprobar en 15 semanas: dos (de cuatro) en Fase Inicial, o tres (de cinco) fuera de la Fase Inicial.

En los estudios de grado la evaluación continua ha cobrado mucho protagonismo. La mayoría de asignaturas tiene múltiples actos de evaluación. Para coordinarlas, se ha ideado un modelo que facilita la realización de pruebas fuera del horario de clase (FHC).

Con el objetivo de facilitar a las asignaturas la realización de actos de evaluación, se han reservado franjas horarias fuera del horario de clase. Denominamos pruebas FHC a los actos de evaluación realizados en estas franjas horarias. El hecho de no planificar clases de ninguna asignatura en esas franjas horarias permite convocar a los alumnos de diversos grupos de una asignatura, o incluso a todos los grupos, a realizar simultáneamente una prueba de la asignatura.

En el contexto de un centro con muchos estudiantes y muchos grupos por asignatura, la realización de actos de evaluación en horas de clase supone la preparación de muchas variantes de examen. Una ventaja potencial de la realización de pruebas FHC es la reducción del número de versiones de cada examen. Esto no sólo reduce el trabajo del profesor en cuanto a la preparación de las pruebas sino que, además, aumenta la equitatividad al evitar la posible desigualdad en el nivel de dificultad de diferentes exámenes preparados para diferentes grupos.

Las franjas horarias FHC pueden ser usadas por diversas asignaturas para realizar pequeños exámenes a lo largo del cuatrimestre. La compartición de estos espacios temporales conlleva la necesidad de sincronización entre diferentes asignaturas. Es imprescindible coordinar las pruebas FHC que los estudiantes pueden realizar en un cuatrimestre determinado para evitar solapamientos entre pruebas de asignaturas que un estudiante pueda cursar simultáneamente y repartir adecuadamente las pruebas a lo largo del cuatrimestre. Por ejemplo, en la planificación de pruebas FHC de la semana 15, que es la última de clases, se han priorizado

aquellas asignaturas que se pueden aprobar en 15 semanas mediante evaluación continua.

No obstante, la coordinación entre las pruebas de evaluación de las asignaturas no es exclusiva de las pruebas FHC. La coordinación puede y debe realizarse también con las pruebas de evaluación realizadas en clase siguiendo los mismos criterios que con las pruebas FHC.

La existencia de pre-requisitos entre asignaturas facilita la planificación de pruebas en las franjas FHC, ya que un estudiante no puede estar cursando simultáneamente dos asignaturas que tengan establecido un pre-requisito.

Con 20 horas de clase en turno de mañana o de tarde, quedan pocas horas disponibles para reservar franjas horarias para pruebas FHC. La solución que hemos implementado consiste en reservar una franja horaria al mediodía, de 12 a 4, dos días a la semana. La ventaja de esta solución es que con sólo dos horas del turno de la mañana reservadas y dos del de la tarde, más la hora de la comida, tenemos dos franjas de tres horas cada semana. En nuestro caso, hemos reservado una hora adicional una de las mañanas, con lo cual tenemos una franja de 4 horas y otra de 3. Algunas semanas, sin embargo, es necesaria una franja adicional, porque coinciden las pruebas de muchas asignaturas. Por ello, algunas semanas aprovechamos excepcionalmente unas horas que en nuestro centro están reservadas para actividades culturales los miércoles de 11 a 14 horas. Con estas franjas hemos podido planificar un total de 42 pruebas FHC de 17 asignaturas, realizadas durante las 15 semanas de clase, asegurando que un estudiante no tiene dos pruebas en un mismo día y puede tener un máximo de dos pruebas por semana.

El inconveniente de esta solución es que los estudiantes tienen exámenes a la hora de la comida. Por ello, hay que dejar tiempo de descanso entre la finalización de las clases y el inicio de las pruebas de evaluación.

4. Coordinación

Para diseñar y gestionar el plan de estudios se requieren diferentes modelos de coordinación a diferentes niveles. Para empezar, es necesario un coordinador para cada asignatura. Además, es conveniente que las asignaturas de la misma materia que trabajan el mismo conjunto de

competencias se coordinen entre sí, por lo que consideramos necesaria la figura de un coordinador de materia. En el caso de asignaturas obligatorias lo denominamos coordinación vertical (o de materia); en el caso de los créditos de tecnología específica lo denominamos coordinación de la especialidad. Consideramos también fundamental gestionar que la carga de las distintas asignaturas que puede matricular un estudiante esté repartida, en conjunto, de forma lo más uniformemente posible a lo largo del curso. A esto lo llamamos coordinación horizontal. Finalmente, las competencias transversales merecen una coordinación aparte. Los profesores estamos poco acostumbrados a trabajar estas competencias y por ello es importante que tengan una coordinación específica, que llamamos coordinación transversal.

En la FIB hemos definido cinco tipos de coordinadores para realizar las coordinaciones previamente descritas: coordinadores de asignatura, verticales (de materia), de especialidad y transversales. La coordinación horizontal de los dos primeros cursos la realizan los jefes de estudios (tenemos dos, uno para la Fase Inicial y otro para el resto de cursos) y la coordinación horizontal de las especialidades la realizan los coordinadores de especialidad.

Se describen a continuación las principales funciones de los diferentes coordinadores, a excepción de los coordinadores de asignatura, dado que son una figura bien conocida y muy usada en la mayoría de centros.

La función de los coordinadores horizontales es coordinar la carga semanal de trabajo del estudiante en las asignaturas de un determinado cuatrimestre, de forma que la carga semanal del conjunto esté distribuida lo más uniformemente posible.

Las funciones de los coordinadores verticales (de materia) son las siguientes:

- Garantizar que las asignaturas coordinadas se diseñan e imparten de forma que los estudiantes adquieran las competencias asignadas a cada asignatura con el nivel de competencia indicado en la guía docente.
- Coordinar los objetivos, contenidos y actividades de las asignaturas coordinadas de forma que no se produzcan excesivos solapamientos de objetivos/contenidos y no haya lagunas de conocimiento.

- Indicar al centro si consideran que es preciso realizar alguna modificación de las competencias asignadas a las asignaturas coordinadas.
- Reunirse al menos una vez por semestre con los coordinadores de las asignaturas coordinadas para realizar un seguimiento de las asignaturas.
- Garantizar que la evaluación de las asignaturas permite evaluar la adquisición de las competencias asignadas.
- Emitir un informe al principio de cada semestre sobre las asignaturas coordinadas, indicando los cambios realizados en cada asignatura respecto al semestre anterior y cómo influyen estos cambios en la adquisición y evaluación de competencias. Entregar dicho informe al jefe de estudios.

Las funciones de los coordinadores de especialidad son, además de las anteriores:

- Coordinar la realización de las diferentes pruebas de evaluación que se realizan en las asignaturas asignadas a un determinado semestre (Q5 o Q6) para que estén distribuidas de la forma más uniforme posible a lo largo de todo el curso. Tratar de conseguir que no se realice más de una prueba cada semana en el conjunto de las asignaturas de la especialidad en un determinado cuatrimestre (en el caso de las asignaturas complementarias no es preciso gestionar esta coordinación, ya que no es posible saber con antelación cuáles matriculará un estudiante).
- Coordinar la carga semanal de trabajo para el estudiante de las asignaturas asignadas a un determinado cuatrimestre, de forma que la carga semanal final del conjunto esté distribuida lo más uniformemente posible.
- Coordinar la carga semanal de trabajo para el estudiante de las asignaturas complementarias, de forma que la carga semanal final de cada una de ellas esté distribuida lo más uniformemente posible.

Las funciones de los coordinadores transversales son las siguientes:

- Definir claramente la competencia en el contexto del Grado en Ingeniería Informática, estableciendo tres niveles de competencia y definiendo indicadores para cada nivel.

- Colaborar con los coordinadores de las asignaturas que tienen asignada la competencia transversal en el diseño de un conjunto adecuado de actividades para trabajar y evaluar la competencia.
- Definir el mecanismo de evaluación que permitirá decidir la nota final del estudiante en la competencia una vez acabados sus estudios

En la FIB consideramos fundamental el trabajo de estos coordinadores, hasta el punto que hemos definido un coordinador independiente para cada competencia transversal (hemos definido 9 competencias).

5. Guía docente

Los profesores de un centro tienen una formación docente muy diversa, desde los que han hecho multitud de cursos (relacionados con docencia) hasta los que no han hecho ninguno. Por ello, es preciso definir un procedimiento muy pautado para diseñar las asignaturas de forma que el diseño se realice siguiendo los criterios que se usaron en la definición del plan de estudios. La guía docente es el mecanismo ideal para hacerlo.

En primer lugar, es importante facilitar la tarea del profesor para que pueda dedicarse a lo importante. El diseño de una asignatura requiere tiempo, mucho tiempo, y hay que evitar que se gaste en tareas mecánicas. Por ello, es conveniente que las competencias, tanto técnicas como transversales, estén preasignadas a la asignatura según la distribución realizada en el plan de estudios. Es importante, no obstante, darle al coordinador la oportunidad de añadir (pero no de eliminar) la posibilidad de trabajar nuevas competencias en la asignatura. No obstante, recomendamos que éstas se escojan a partir de una lista predefinida para evitar tener un conjunto de competencias definidas con distintos niveles de granularidad.

A partir de las competencias, el coordinador debe definir los objetivos de la asignatura, indicando a qué competencia/s corresponde cada objetivo. Los objetivos pueden definirse a un único nivel o a más de uno (no recomendamos más de dos niveles).

Lo ideal sería definir a continuación las actividades de la asignatura. Lamentablemente, la mayoría de los profesores piensan todavía en los

contenidos como el vehículo adecuado para “transmitir conocimientos”, en lugar de plantear actividades como mecanismo de “aprendizaje” del estudiante. Por ello, creemos necesario que después de los objetivos el coordinador defina los contenidos de la asignatura, de forma que le sirvan de ancla para diseñar las actividades. No obstante, no es importante relacionar los contenidos con los objetivos ni con las competencias previamente definidas, para conseguir que el diseño de la asignatura se centre en las actividades.

El diseño de las actividades puede hacerse de muy distintas formas. Los profesores mejor formados en aprendizaje activo diseñarán sin duda las actividades a partir de los objetivos de la asignatura, usando los contenidos únicamente como una referencia lateral. Los profesores más “tradicionales”, por el contrario, diseñarán las actividades desde el punto de vista de los contenidos, y les será más sencillo pensar las actividades en unidades como “clases” o “semanas”, que les permitan organizar fácilmente los contenidos. Consideramos importante que la introducción de actividades sea sencilla para todos los profesores, y por lo tanto es conveniente que los dos modelos descritos convivan. Cada actividad debe contener una descripción de las tareas a realizar por el profesor y el estudiante, tanto dentro como fuera de clase, así como una estimación razonable del tiempo de dedicación del estudiante, indicando las diferentes tareas a que debe dedicarlo. También es preciso indicar con qué objetivos y contenidos está relacionada cada actividad. De esta forma, es posible saber qué actividades se están haciendo para desarrollar cada objetivo (y por lo tanto, cada competencia), y es posible incluso cuantificar el tiempo dedicado por el estudiante a trabajar cada competencia. Esta información puede ser muy valiosa a la hora de evaluar el resultado del plan de estudios o descubrir por qué alguna cosa no funciona.

A continuación debe describirse, en líneas generales, la metodología docente de la asignatura (en general, ya que en diferentes actividades se pueden usar metodologías distintas).

Finalmente, hay que definir el método de evaluación. La evaluación es un aspecto fundamental del diseño de la asignatura, ya que los estudiantes trabajan generalmente guiados por el método de evaluación. Su objetivo es aprobar, mientras que el del profesor es que el estudiante

aprenda. La única forma de hacer que ambos objetivos coincidan es diseñar un método de evaluación que obligue al estudiante a trabajar regularmente en la asignatura. El método de evaluación debe definir la forma de calcular la nota final de la asignatura y las distintas pruebas y entregas que realizará el estudiante. Para cada prueba hay que especificar los objetivos de los que se va a evaluar al estudiante y la forma en que se realizará esta evaluación (test, oral, examen escrito, prácticas de laboratorio, etc.). En el caso de que la asignatura tenga asignada alguna competencia transversal, recomendamos que esta evaluación tenga un peso en la nota final de la asignatura y que además se presente como una nota aparte, de forma que se visualice ante estudiantes y profesores que estas competencias son importantes.

Finalmente, la guía docente debe ofrecer al estudiante una bibliografía básica (no muy extensa) y una bibliografía complementaria para aquellos estudiantes con un interés especial por la asignatura.

Si el coordinador de la asignatura la diseña usando un aplicativo informático que siga las directrices de esta sección, de forma que pueda escoger el máximo posible de información para el diseño a partir de desplegados (las competencias iniciales, la asignación de competencias a objetivos, la selección de objetivos en las actividades, etc.), invertirá su tiempo en pensar en la asignatura y este pensamiento estará guiado por la forma en que introduce la información

6. Evaluación y conclusiones

En este trabajo se han presentado algunos mecanismos de coordinación que facilitan la correcta adquisición de competencias en unos estudios de grado. La coordinación vertical permite una correcta distribución y seguimiento de las competencias entre diferentes asignaturas, la coordinación horizontal facilita una distribución homogénea del trabajo del estudiante durante el curso, la coordinación transversal es importante para asegurar que las competencias transversales se trabajan de forma apropiada.

Referencias

- [1] *Acuerdo del Consejo de Universidades por el que se establecen recomendaciones para la propuesta por las universidades de memorias de solicitud de títulos oficiales en los ámbitos de la ingeniería informática, ingeniería técnica informática e ingeniería química.* <http://www.fic.udc.es/CODDI/documentacion/fichas13marzo.pdf>. Última consulta, febrero de 2011.
- [2] Agència per a la qualitat del sistema universitari a Catalunya. *Guia per a l'avaluació de competències en els Treballs Fi d'Estudis a les Enginyeries.* Disponible en: http://www.aqu.cat/publicacions/guies_competencies/index.html. Última consulta, febrero de 2011.
- [3] García J., Sánchez F. y Gavalda R.. *Cómo diseñar un Grado en Ingeniería Informática.* JENUI 2006
- [4] García J., Sánchez F. y Gavalda R. *Recomendaciones para el diseño de una titulación de Grado en Informática.* IEEE RITA, Revista Iberoamericana de Tecnologías del Aprendizaje. Vol 2, Num 2, Noviembre 2007.
- [5] Ministerio de Ciencia e Innovación. *Guía para la evaluación de competencias en los trabajos de fin de estudios de las Ingenierías.* 2009.
- [6] Sánchez F. *Criterios de diseño y condiciones de entorno de un plan de EEEstudios de Grado.* ReVision. Vol 2, No 1. 2009.
- [7] Sánchez F; García J., Díaz M., Riesco M., Pérez J.R., Juan A. *Estrategia de diseño y aspectos a considerar en los planes de EEEstudios de Grado en Ingeniería Informática.* ReVision. Vol 1, No 1. 2008.
- [8] Sánchez F., García J., Gavalda R., Díaz M., Riesco M., Pérez J.R. y Juan A.A. *Estrategias de diseño para las titulaciones de informática del EEES.* Novática, ISSN 0211-2124, N° 187, pp. 45-48, Mayo-Junio 2007.
- [9] Valderrama E. et al. *La evaluación de competencias en los trabajos de Fin de estudios.* JENUI 2009.
- [10] Valderrama E. et al. *La evaluación de competencias en los trabajos de Fin de estudios.* IEEE RITA, Vol. 5, número 3, Agosto 2010.

Una definición precisa del concepto “Nivel de Dominio de una Competencia” en el marco del Aprendizaje Basado en Competencias

Aurelio Bermúdez, Ismael García-Varea, María T. López, Francisco Montero,
Luis de la Ossa, José M. Puerta, Tomás Rojo, José L. Sánchez

Departamento de Sistemas Informáticos
Universidad de Castilla-La Mancha
Campus Universitario s/n, 02071 Albacete

[fabermudez](mailto:fabermudez@uclm.es) | [ivarea](mailto:ivarea@uclm.es) | [mlopez](mailto:mlopez@uclm.es) | [fmontero](mailto:fmontero@uclm.es) | [ldelaossa](mailto:ldelaossa@uclm.es) | [jpuerta](mailto:jpuerta@uclm.es) | [trojo](mailto:trojo@uclm.es) | [jsanchez](mailto:jsanchez@uclm.es) | [dsi.uclm.es](mailto:dsi@uclm.es)

Resumen

Enmarcados dentro del Espacio Europeo de Enseñanza Superior (EEES), los planes de estudio de los nuevos títulos universitarios de Grado han sido desarrollados bajo el paradigma del Aprendizaje Basado en Competencias (ABC).

El proceso de evaluación bajo el modelo de ABC se construye a partir del concepto “nivel de dominio” de una competencia alcanzado por el alumno. Sin embargo, se han propuesto diferentes definiciones para este concepto que, aunque correctas, son incompletas y/o ambiguas. En este trabajo proponemos una definición clara, completa y precisa de “nivel de dominio”, que generaliza las distintas interpretaciones encontradas en la literatura, mediante una serie de parámetros. Estos parámetros dotan a la definición de la suficiente flexibilidad para poder instanciarla a distintas interpretaciones particulares.

Esta definición ha facilitado la aceptación de este concepto y su aplicación en la elaboración de los estudios de Grado por parte del profesorado del Departamento de Sistemas Informáticos en la Escuela Superior de Ingeniería Informática de Albacete.

Summary

Within the European Higher Education Area (EHEA), the curriculum of the new university degrees has been developed under the Competency based Learning (CBL) paradigm.

The evaluation process under the CBL model is constructed from the concept of “domain level” of competence achieved by the students. However, different definitions have been proposed so far for this concept that, although correct, is incomplete and/or ambiguous. In this paper we propose a clear, complete and accurate

definition of the concept of “domain level” of competence definition, which generalizes the different interpretations found in the literature, through a series of parameters. These parameters provide enough flexibility to implement this definition to particular interpretations. This definition has facilitated its acceptance by the faculty of our department, in order to be implemented in the definition of the curriculum of Computer Science graduate studies at the “Escuela Superior de Ingeniería Informática de Albacete”.

Palabras clave

ABC, nivel de dominio, evaluación en ABC, rúbrica, competencia

1. Introducción

Con la puesta en marcha de los nuevos títulos de Grado, enmarcados dentro del Espacio Europeo de Enseñanza Superior (EEES), la gran mayoría de los planes de estudio de las universidades españolas han sido desarrollados bajo el modelo de Aprendizaje Basado en Competencias (ABC). Es decir, estableciendo como directriz principal en su elaboración la adquisición de una serie de competencias, tanto transversales como específicas, por parte del alumnado. En este contexto, un concepto fundamental en el modelo ABC es el de “nivel de dominio”, ligado a la elaboración de rúbricas [1, 6] y a la evaluación.

Sin embargo, los diferentes autores presentan e interpretan este concepto de forma notablemente distinta. A modo de ejemplo, algunos lo asocian con el nivel de desempeño o logro por parte del alumnado [1]. En otras referencias el nivel de dominio se presenta de forma más abstracta

proponiendo varios niveles de manera heurística [6]. Fruto de este escenario, consideramos que esta ambigüedad del término supone un obstáculo importante de cara a fijar unos criterios de evaluación.

En este trabajo proponemos una definición alternativa, aunque basada en las identificadas, para el concepto de nivel de dominio. Dicha definición se presenta en base a una serie de parámetros con los que, por un lado, se ha intentado cubrir todas las interpretaciones encontradas en la literatura y, por otro, se ha querido garantizar la flexibilidad en el uso del término por parte del equipo docente. Dicha definición se ha puesto en práctica con éxito en los estudios de Grado en Informática, en la Escuela Superior de Ingeniería Informática de Albacete, durante el curso 2010-11.

Este trabajo está organizado como se describe a continuación. En el segundo apartado, contextualizaremos nuestra contribución, tratando y describiendo el alcance del modelo de aprendizaje basado en competencias. En él aparecerá el concepto de nivel de dominio. En el apartado tercero se presentará nuestra principal contribución: la caracterización del concepto de nivel de dominio de una competencia. En el apartado cuarto se presenta un caso de estudio, que consiste en la aplicación de la definición propuesta en los estudios de Grado de la Escuela Superior de Ingeniería Informática de Albacete. El último apartado recogerá las conclusiones alcanzadas con la realización de este trabajo, así como algunas líneas de actuación para el futuro.

2. Diseño basado en competencias

En la actualidad, el diseño del plan de estudios de cualquier titulación universitaria está enfocado a la adquisición de un conjunto de competencias necesarias para el desarrollo de la profesión correspondiente. En el caso de la Ingeniería Informática, existe una recomendación sobre las competencias deseables [4], que ha sido interpretada por cada centro a la hora de implementar sus titulaciones.

Pero, ¿qué se entiende por competencia? Daremos tan solo un par de definiciones. Según la propuesta de directrices para la elaboración de títulos de grado y máster del Ministerio de Educación y Ciencia [2], las competencias son una combinación de conocimientos, habilidades

(intelectuales, manuales, sociales, etc.), actitudes y valores que capacitarán a un titulado para afrontar con garantías la resolución de problemas o la intervención en un asunto en un contexto académico, profesional o social determinado.

Por otro lado, y en la misma línea que la anterior, Villa y Poblete [7], definen competencia como el buen desempeño en contextos diversos y auténticos basado en la integración y activación de conocimientos, normas, técnicas, procedimientos, habilidades y destrezas, actitudes y valores.

Las competencias se agrupan normalmente en dos grandes categorías: genéricas (o transversales) y específicas. Las primeras están relacionadas con la formación integral de las personas y se clasifican, a su vez, en instrumentales (aquellas que se emplean como medio o herramienta para obtener un determinado fin), interpersonales (las referidas a las diferentes capacidades que hacen que las personas logren una buena interacción con los demás) y sistémicas (aquellas relacionadas con la comprensión de la totalidad de un sistema o conjunto). Parece ser que existe un consenso con respecto a cuáles deberían ser las competencias genéricas: habilidades de comunicación en general; gestión de la información (búsqueda, selección, análisis y evaluación de la información procedente de diversas fuentes); habilidades para la utilización de las nuevas tecnologías; trabajo en grupo, ética, reconocimiento de la diversidad; competencias personales como gestión del tiempo, responsabilidad o planificación. Por su parte, las competencias específicas posibilitan una orientación profesional que permite a los titulados una integración en el mercado de trabajo.

El aprendizaje basado en competencias (ABC) consiste en desarrollar las competencias genéricas y específicas con el propósito de capacitar a la persona sobre los conocimientos científicos y técnicos, así como de aplicarlos en contextos diversos y complejos, integrándolos con sus propias actitudes y valores en un modo propio de actuar personal y profesionalmente [7]. Previamente a su aplicación, y partiendo del perfil académico-profesional correspondiente, se deben establecer las competencias a desarrollar por los estudiantes que estén realizando un determinado tipo de estudios.

El mapa de competencias de un proyecto formativo para una titulación dada queda definido por las competencias elegidas para dicha titulación y su distribución en los cursos en los

que se ha organizado. El éxito de este modelo pasa por una adecuada coordinación y colaboración entre todos los miembros del cuerpo formativo. La dificultad principal radica en que todos ellos deben contribuir a la consecución de un objetivo común, marcado por el perfil académico-profesional de que se trate en cada caso, desde múltiples módulos o asignaturas.

Al contrario de lo que sucedía en los modelos de aprendizaje tradicionales, basados en contenidos y centrados en la figura del profesor, en el ABC el estudiante juega un papel muy importante, ya que el proceso de aprendizaje está centrado en la propia responsabilidad del estudiante y en el desarrollo de su autonomía. No está resultando fácil hacer ver al estudiante este papel estelar que se le ha asignado en el nuevo modelo y será también labor del profesorado sacarle del rol de mero receptor pasivo al que estaba relegado. Hay que tener en cuenta, eso sí, que el convencimiento personal del estudiante es clave para lograr el objetivo final. Y es que, como se indica en [7], adquirir y desarrollar las competencias supone la implicación personal de cada estudiante, involucrándose en cada actividad propuesta, reflexionando intelectualmente en cada tarea y realizando su propia valoración del aprendizaje que va realizando.

Dadas sus características, el ABC requiere la aplicación de nuevas metodologías, que deben apoyarse en nuevas e innovadoras actividades docentes. Metodologías que permitan combinar la adquisición de conocimientos con el aprendizaje de competencias. En la literatura se puede encontrar una amplia variedad de estas metodologías de enseñanza y aprendizaje orientadas a facilitar la obtención de las competencias recogidas en un determinado plan de estudios [3]. Entre estas estrategias se encuentran, por citar algunas, el método expositivo, el aprendizaje basado en problemas, el estudio de casos y el aprendizaje cooperativo.

Un aspecto esencial en cualquier modelo de aprendizaje, y en particular en ABC, es el de la evaluación. En este caso, la evaluación del estudiante debe centrarse en la valoración y validación del nivel de consecución de las competencias requeridas.

Como ya se ha comentado, a la hora de medir el grado de consecución de una competencia, se definen una serie de niveles de dominio (también

desempeño, o alcance) que se centran en aspectos concretos de la competencia y se corresponden con distintos momentos en el transcurso de la formación. Dichos niveles son evaluados de forma independiente, por ejemplo, mediante el empleo de rúbricas.

Al ser más concretas, las competencias son más fáciles de evaluar y se puede hacer uso de criterios específicos como indicadores del grado de logro de cada competencia específica. Es habitual manejar varios criterios de evaluación para cada competencia específica, disponiendo así de un buen número de indicadores que permiten evaluar con garantías la competencia. En [5] y [7] pueden encontrarse detalles sobre el desarrollo y la evaluación de numerosas competencias genéricas muy comunes en educación superior.

3. Nivel de Dominio de una Competencia

Como ya se ha comentado en el apartado 1, cuando se habla del nivel de dominio no queda claro si se trata del nivel de profundización con el que se aborda una determinada competencia, o por el contrario, del nivel al que dicha competencia se evalúa.

En nuestra opinión, estos dos conceptos no pueden ir aislados, de hecho en la definición que proponemos a continuación están íntimamente relacionados.

Por otra parte, es evidente que toda competencia propuesta en el plan de estudios deberá ser tratada al menos por una asignatura. Normalmente, las competencias específicas se tratarán en una sola asignatura, o en casos excepcionales en dos o a lo sumo en tres. En lo que respecta a las competencias transversales, parece razonable que sean tratadas en varias asignaturas y en todos (o casi todos) los cursos de la titulación y a diferentes niveles de profundización. En este sentido, también habrá que tener en cuenta el grado o porcentaje de dedicación que cada asignatura trabajará, y por tanto evaluará, una competencia dentro de cada de nivel de profundización considerado.

Por tanto, el nivel de dominio de una determinada competencia (en un momento en concreto de la carrera) se define mediante dos parámetros ($N_P; N_E$), donde:

- N_p es el nivel de profundización al que se ha trabajado dicha competencia, y
- N_E es el resultado de la evaluación al nivel de profundización N_p .

Para completar esta definición veamos, con más detalle, en qué consisten estos dos parámetros.

3.1. Nivel de profundización

En primer lugar debemos establecer los distintos niveles de profundización a los que se tratará una competencia. Estos niveles se definirán de manera categórica y, obviamente, en orden creciente de profundización.

A modo de ejemplo, se podrían definir tres niveles de profundización: Básico, Intermedio y Avanzado (numéricamente 1, 2, y 3).

Evidentemente, estos niveles deberán tratarse y desarrollarse de manera progresiva conforme el alumno vaya avanzando y madurando en sus estudios. Por tanto, los niveles de profundización deberán tener total coherencia con los cursos/cuatrimestres de la titulación. Es decir, para cada curso/cuatrimestre se debe establecer el mismo nivel de profundización para todas las competencias transversales. Esto no es de aplicación estricta a las competencias específicas, dado que la gran mayoría solamente se trabaja en una sola asignatura, y por tanto en un solo curso/cuatrimestre, en cuyo caso deberá tratarse a todos los niveles definidos, aunque también de manera progresiva dentro del curso/ cuatrimestre.

Siguiendo con el ejemplo anterior, en el que hemos considerado tres niveles de profundización, sería deseable que en primer curso (cuatrimestres 1 y 2) se establezca un nivel de profundización $N_p = 1$, en segundo curso (cuatrimestres 3 y 4) un nivel $N_p = 2$, y tercer y cuarto curso (cuatrimestres 5 al 8) un nivel $N_p = 3$.

Por otra parte, parece razonable que una competencia transversal se trabaje (a un nivel de profundidad en concreto) de manera simultánea en varias asignaturas del mismo curso/cuatrimestre. Esto conllevará un mayor esfuerzo en lo que respecta a coordinación, pero proporcionará mayor objetividad en su evaluación (realizada por distintos profesores) y pluralidad en la forma de ser tratada. En nuestra opinión, lo ideal sería que una competencia se trate, a un determinado nivel, en dos o a lo sumo en tres asignaturas de manera simultánea.

De acuerdo a esto último, también es necesario tener en cuenta el porcentaje o grado de dedicación que una asignatura trabaja una determinada competencia, a un determinado nivel N_p . Por ejemplo, supongamos que la competencia COMP se trabaja a nivel $N_p = 1$ en tres asignaturas (A, B, y C) del primer cuatrimestre. Lo razonable es que esas tres asignaturas la trabajen por igual, es decir, se les asigne un grado de dedicación del 33% a cada una, aunque cabe la posibilidad de realizar cualquier otra asignación (digamos 40% a A, 20% a B, y 40% a C), siempre y cuando entre ellas sumen el 100 %. En cualquier caso, esta asignación deberá tenerse en cuenta a la hora de confeccionar la guía del alumno de cada una de esas asignaturas, y, en consecuencia, para establecer la evaluación que de ella se realice en cada asignatura.

3.2. Evaluación al nivel N_p

En primer lugar deberemos establecer los grados (también categóricos) a los que se evaluará una competencia para cada nivel N_p . Esta escala para N_E se establecerá en la rúbrica definida para la competencia, la cual evidentemente deberá establecerse a su vez, a los distintos niveles de profundización establecidos. A modo de ejemplo, podrían definirse cinco grados de evaluación: Muy Bajo, Bajo, Normal, Alto y Muy Alto (numéricamente 1, 2, 3, 4 y 5).

La evaluación N_E en cada nivel de profundización N_p , deberá realizarse atendiendo a los mismos criterios (establecidos en la rúbrica) y, siempre, deberá realizarse al 100% entre todas las asignaturas que trabajen esa competencia a dicho nivel N_p . Es decir, la evaluación de la competencia debería realizarse de forma global y consensuada entre las asignaturas que la trabajen, indistintamente del porcentaje que cada asignatura tenga asignado para tratarla.

A priori, podría pensarse que dos asignaturas podrían otorgar un nivel de dominio (N_p ; N_E) distinto a una competencia dada, por el simple hecho de que un alumno supere una asignatura y no otra (que trabajen la misma competencia). Esto no sería razonable teniendo en cuenta que dicha competencia se debe evaluar al mismo nivel y con los mismos criterios en todas y cada una de las asignaturas en que se trabaje, y por tanto, el nivel de dominio adquirido por un alumno debería ser el mismo, indistintamente además del grado o

porcentaje que cada asignatura dedique a dicha competencia.

3.3. Un ejemplo de nivel de dominio

Consideremos la competencia transversal Correcta Comunicación Oral y Escrita (CCOE), para la que se establecen tres niveles de profundización y cinco grados de evaluación. Es decir, se ha definido una rúbrica para CCOE que tendría un aspecto similar¹ a la Tabla 1, en donde las filas establecen los niveles N_p y las columnas los grados N_E .

CCOE	MB (1)	B (2)	Me (3)	A (4)	MA (5)
Básico (1)					
Interm. (2)					
Avanz. (3)					

Tabla 1. Esquema para elaboración de rúbricas

Supongamos que CCOE se trabaja en dos asignaturas A y B, a un nivel de profundidad $N_p = 2$, tratándose al 50% entre ellas. El grado de evaluación N_E debería obtenerse a partir de los criterios de la fila "Intermedio" de la tabla anterior para ambas asignaturas.

Supongamos además que, en la asignatura A se trabaja esa competencia mediante 7 trabajos (4 escritos y 3 presentaciones orales), y en la asignatura B se realizan 3 trabajos (1 escrito y 2 presentaciones orales) del mismo tipo y dificultad, esto último al nivel $N_p = 2$. Parece que lo razonable sería que ambas asignaturas dedicasen el mismo esfuerzo (de cara al alumno) para trabajar y evaluar la competencia CCOE, es decir, que ambas asignaturas propusieran el mismo número de trabajos. Claramente, esto no es obligatorio, pues ello dependerá del número de competencias (específicas y transversales) que se trabajen en A y B, del criterio del profesorado, y de la planificación realizada en la guía del alumno para esas asignaturas, entre otros factores. Evidentemente, éste y otros muchos aspectos relacionados con la evaluación de una asignatura dependen de muchos factores, los cuales se salen

del ámbito de este trabajo, y por tanto no trataremos aquí.

En definitiva, todo alumno que supere con éxito ambas asignaturas habrá alcanzado un nivel de dominio mínimo de (2,3) en CCOE, habiéndose trabajado al 100% entre ellas.

Por otra parte, es necesario destacar que ese nivel de dominio alcanzado no tiene por qué estar directamente relacionado con la nota numérica que se obtenga en dicha asignatura (la cual también dependerá de otras competencias transversales y, en mayor medida, de las competencias específicas que se trabajen en esa asignatura). Por ejemplo, un alumno con una calificación final de 8 (sobre 10) en las asignaturas A y B podría haber alcanzado un nivel de dominio en CCOE de (2,3), y por otra parte, otro alumno con una nota numérica de 5 en ambas podría alcanzar un nivel de dominio en CCOE de (2,5). Evidentemente, un alumno con una nota numérica de 10 en A y B, debería haber alcanzado también un nivel de dominio en CCOE de (2,5).

4. Un caso de estudio: Grado en Ingeniería Informática de la Escuela Superior de Ingeniería Informática de Albacete

Ante la puesta en marcha de los nuevos planes de estudio de Grado en Ingeniería Informática se plantea el reto de cómo coordinar las competencias específicas descritas en el plan de estudios y sobre todo cómo distribuir las competencias transversales, establecer los niveles de profundización (N_p) y realizar su evaluación de acuerdo a los grados de evaluación (N_E) establecidos. Hay que tener en cuenta que las competencias transversales sólo se relacionan en el plan de estudios pero no se indica su nivel de dominio, es decir los niveles de profundización y evaluación. En este caso nos centramos en las competencias transversales ya que las competencias específicas están más claras en el entorno del plan de estudios y desarrollo de los planes docentes de las asignaturas.

En una primera reunión de coordinadores se plantea definir las condiciones de partida que permitan desarrollar un sistema de coordinación que asegure implantar la docencia y evaluación de las competencias transversales. Estas condiciones quedan resumidas en los siguientes puntos:

¹ Para el caso que estamos tratando no consideramos relevante rellenar el contenido de la rúbrica, no obstante puede encontrarse una similar en [7]

- Las competencias deben ser tratadas a lo largo de la carrera hasta su adquisición total en los últimos cursos.
- No todas las asignaturas deben tratar todas las competencias transversales.
- Al igual que los conocimientos específicos, las competencias transversales deben ir adquiriéndose paulatinamente a lo largo de los estudios.
- Se plantean tres niveles en la profundización de las competencias: Básico, Intermedio y Avanzado.
- En cada nivel estarán implicadas, en el desarrollo de la competencia, a lo sumo tres asignaturas, distribuidas en cada curso.
- Cada asignatura desarrollará, como norma general, a lo sumo dos competencias transversales.

El primer paso consiste en conocer cuáles pueden ser las asignaturas más adecuadas a cada competencia en los distintos cursos. Para ello se establece un cuadro de competencias por asignaturas en cada curso y se plantea a los profesores del curso que identifiquen qué competencias consideran adecuado desarrollar en sus asignaturas y a qué nivel de profundidad (N_p). Con estos datos, se dispone de la información necesaria para iniciar la asignación de competencias por asignaturas, dado que de esta forma ya conocemos qué competencias están dispuestos a desarrollar los profesores de las asignaturas.

El siguiente paso consiste en asignar las competencias y nivel de profundidad a desarrollar en cada asignatura respetando en lo posible las opiniones que el profesorado ha establecido en su elección inicial. De esta forma se genera un cuadro por cada competencia indicando qué asignaturas se encargan de la competencia en cada curso y a qué nivel de profundidad se trabajarán, así como el porcentaje de dedicación, el cual se intentará distribuir de manera equitativa.

El paso más delicado consiste en definir exactamente cuál es la evaluación asociada a los niveles de profundización de cada competencia. Esta tarea recae fundamentalmente en el grupo de asignaturas asociadas a un mismo nivel, la cual consistirá en definir una rúbrica de forma consensuada entre los profesores de las asignaturas implicadas.

Este esquema obliga a una coordinación horizontal y vertical, pero solamente de las asignaturas implicadas en el trabajo de la competencia y también desemboca en una definición de nivel de competencia adaptado a cada competencia en sí y al número de asignaturas involucradas en la misma.

Como consecuencia de este proceso, se obtiene un cuadro por competencia con las asignaturas implicadas en el trabajo y evaluación de la misma, pero además, acompañado de una rúbrica que especifica con claridad cuáles son los niveles de dominio ($N_E; N_P$) de la competencia. De esta forma, cada asignatura tiene a su disposición criterios claros y coordinados para integrar en la planificación de su docencia y evaluación, conociendo el alcance e impacto de la competencia en la planificación docente de la misma.

4.1. Ejemplo: Competencia Transversal: Análisis y Síntesis

Según diferentes autores [1, 7], los alumnos que adquieran la competencia transversal de Análisis y Síntesis deben ser capaces de:

1. Estructurar la información relevante de un problema, de tal forma que facilite la interpretación del fenómeno.
2. Detectar las cuestiones esenciales de una situación problemática, así como la generación de soluciones viables y la selección de las más convenientes, de acuerdo al contexto en que se vive.
3. Hacer abstracciones que identifiquen los elementos esenciales que conforman un fenómeno particular, e integrarlos de una manera coherente.
4. Generar hipótesis y diseñar procesos para verificarlas.
5. Formular juicios críticos sobre los distintos modelos que explican un cierto fenómeno
6. Formular juicios críticos sobre las soluciones que se proponen para un cierto problema.

La competencia de analizar y sintetizar requerirá:

- El establecimiento de un objetivo para el que aplicar la capacidad.
- Poseer un conocimiento básico que permita destacar las características más relevantes.

- Detectar las propiedades de las partes y las relaciones entre ellas.
 - Componer las partes de un modo diferente al original, así como decidir sobre cómo hacer la descomposición o la composición.
 - Tener en cuenta la propia secuencia que se analiza o sintetiza.
- Dentro de las posibilidades de tratar la competencia tendríamos: proporcionar información (textos, vídeos, páginas web, monografías, artículo, etc.) al alumno, para que la analice y la resuma. Escribir instrucciones, recomendaciones y normas sobre cada tipo de trabajo. Recomendar la realización de mapas conceptuales por el alumno. Realizar presentaciones de guiones y/o trabajos requeridos a partir del material proporcionado.

Competencia Análisis y Síntesis

N_p	Primer cuatrimestre	Segundo cuatrimestre		
1	Fund. Program. I (50%)	Est. de Computadores (50%)		
2	Lógica (50%)	Met. De la Pr4og (50%)		
3	Sist. Inteligentes (50%)	Progr. Declar (25%)	Proc. Leng. (25%)	INT. 1
		Dis. Inf. Red. (25%)	Segur Redes (25%)	INT. 2
		Int. Sist. Infor. (25%)	Dis. Gest. Redes (25%)	INT. 3
		Ing. Req. (25%)	Cal. Sist. Soft. (25%)	INT. 4

Tabla 2. Distribución por asignaturas y por nivel profundización de la Competencia Análisis y Síntesis.

Rúbrica Análisis y Síntesis	N_E				
	Muy bajo (1)	Bajo (2)	Medio (3)	Alto (4)	Muy alto (5)
$N_p = 1$: Seleccionar los elementos significativos y relaciones en planteamientos sencillos.	No identifica los elementos más sobresalientes.	Sólo identifica alguno de los elementos más sobresalientes.	Identifica la mayoría de los elementos sobresalientes pero no los agrupa correctamente.	Enumera todos los elementos agrupando la mayoría de forma correcta.	Clasifica todos los elementos de forma ordenada y sistemática
$N_p = 2$: Seleccionar los elementos significativos y relaciones en situaciones complejas.	Las relaciones que establecen son incorrectas.	Establece pocas relaciones y algunas son incorrectas.	Establece la mayoría de relaciones pero algunas son incorrectas.	Establece correctamente la mayoría de las relaciones.	Establece correctamente todas las relaciones argumentando y extrayendo conclusiones.
$N_p = 3$: Identificar las carencias de información y relaciones con elementos externos a la situación planteada.	No identifica las lagunas o las incoherencias de la información.	Identifica sólo parte de las lagunas o incoherencias.	Identifica la mayoría de las lagunas.	Identifica todas las lagunas.	Señala las incoherencias.

Tabla 3. Rúbrica por nivel de dominio para la competencia Análisis y Síntesis

Para establecer el nivel de dominio de la competencia Análisis y Síntesis se identificaron diferentes asignaturas de los primeros tres cursos del Grado adecuadas de acuerdo a sus

planificaciones docentes. Al final del proceso, esta competencia queda con la distribución que aparece en el Tabla 2. Hemos de aclarar que en el sexto cuatrimestre empiezan las tecnologías específicas del plan de estudios de Grado. Es por ello, que en el sexto cuatrimestre sean dos asignaturas por tecnología las encargadas de trabajar y evaluar cada una de las competencias transversales. Por otra parte, también se incluyen los porcentajes de dedicación de cada asignatura.

Una vez definidas las asignaturas que se encargan del trabajo y evaluación de la competencia analizada, por niveles de profundidad y cuatrimestres, se ha consensuado la rúbrica para su evaluación y adquisición, y por tanto para establecer el nivel de dominio correspondiente. Dicha rúbrica se muestra en el Tabla 3.

5. Conclusiones

A la hora de implantar muchas de las metodologías de enseñanza y aprendizaje que contempla el EEES, en el contexto del modelo de Aprendizaje Basado en Competencias, cualquier equipo docente se encuentra con la necesidad de coordinar y distribuir de una manera organizada las distintas competencias a lo largo de las asignaturas y cursos que conforman una titulación.

Si hasta el momento, muchos de los esfuerzos de coordinación se centraban en velar por los contenidos, ahora también deben tenerse en cuenta las habilidades, actitudes, capacidades o valores que el alumnado debe alcanzar en cada momento y el nivel de exigencia al que dichos elementos serán evaluados.

En este contexto, aparece el concepto nivel de dominio de una competencia. Dicho nivel exige un tratamiento coordinado tanto en vertical como en horizontal por parte del equipo docente. En este sentido, en este artículo proponemos una definición del concepto de nivel de dominio de una competencia que, a nuestro juicio, engloba todo el alcance y la dimensión de dicho concepto.

La caracterización del concepto de nivel de dominio propuesta en este artículo ha sido implantada y aceptada con éxito en la Escuela Superior de Ingeniería Informática de Albacete durante el curso 2010-11.

Si es importante la aceptación de esta propuesta por parte de los estamentos implicados (fundamentalmente profesorado y estudiantes), no

lo es menos la valoración de la implantación del sistema una vez finalizada. Por ello, en estos momentos se están estudiando los mecanismos necesarios para medir los niveles de implantación, así como el éxito del sistema propuesto.

Agradecimientos

Deseamos agradecer la colaboración del equipo directivo y de profesores de la Escuela Superior de Ingeniería Informática de Albacete.

Referencias

- [1] J. Biggs and C. Tang. *Teaching for Quality Learning at University (Society for Research Into Higher Education)*. Open University Press, 3 edition, November 2007.
- [2] Ministerio de Educación y Ciencia. Directrices para la elaboración de títulos universitarios de grado y máster, diciembre 2006.
- [3] M. de Miguel et. al. *Metodologías de enseñanza y aprendizaje para el desarrollo de competencias. Orientaciones para el profesorado universitario ante el Espacio Europeo de Educación Superior*. Alianza Editorial, Madrid, 2006.
- [4] Secretaría General de Universidades. Resolución de 8 de junio de 2009, sobre recomendaciones a las memorias de solicitud de títulos en el ámbito de la informática, BOE núm. 187, agosto 2009.
- [5] A. Blanco et. al. *Desarrollo y Evaluación de Competencias en Educación Superior*. Narcea, Madrid, 2009.
- [6] D.D. Stevens and A.J. Levi. *Introduction To Rubrics: An Assessment Tool To Save Grading Time, Convey Effective Feedback and Promote Student Learning*. Stylus Publishing, 2004.
- [7] A. Villa and M. Poblete. *Aprendizaje basado en competencias. Una propuesta para la evaluación de competencias genéricas*. Mensajero, Bilbao, 2007.

La promoción de los estudios de Informática mediante la olimpiada de Informática

Francisco J. Alfaro, Juan J. Pardo, José P. Molina, Pedro J. García
Departamento de Sistemas Informáticos. Universidad de Castilla-La Mancha
Escuela Superior de Ingeniería Informática, 02071, Albacete
{Fco.Alfaro, JuanJose.Pardo, JosePascual.Molina, PedroJavier.Garcia}@uclm.es

Resumen

Durante los últimos años las carreras de Informática han sufrido un descenso generalizado en el número de alumnos. Esto ha provocado que muchas universidades hayan aumentado sus esfuerzos para acercar a los alumnos de secundaria a sus estudios y publicitarlos entre los posibles interesados.

En esta ponencia se describe la Olimpiada de Informática que desde hace cuatro años venimos organizando en la Universidad de Castilla-La Mancha para la promoción de nuestras titulaciones entre los alumnos de secundaria de nuestra región. El interés en esta competición ha ido creciendo año a año y hoy en día se encuentra plenamente consolidada.

1. Introducción

Desde hace varios años, los estudios universitarios de Informática en España se enfrentan a una situación que podríamos calificar de paradójica. Por una parte, las titulaciones relacionadas con la Informática se encuentran entre las más demandadas dentro del mercado laboral, incluso en una situación económica sumamente adversa como la actual. Por otra parte, y a pesar de ello, el número de estudiantes que optan por matricularse en dichas titulaciones ha decrecido constante y notablemente sin que, además, se vislumbre un cambio significativo de tendencia a corto plazo. En general, este descenso en el número de matriculados es común a la gran mayoría de carreras técnicas, especialmente las ingenierías, pero sin duda el caso de la Informática resulta particularmente sorprendente, teniendo en cuenta tanto las mencionadas perspectivas profesionales como la enorme popularidad e implantación que todo lo relacionado con la Informática tiene en la sociedad actual.

Las razones que se han sugerido en diversos estudios para esta escasa popularidad actual de las carreras de Informática son varias. Aunque no está den-

tro de los objetivos de este trabajo analizar dichas causas, citaremos las dos que más frecuentemente se indican. Por una parte, se ha señalado una falta de preparación adecuada de los estudiantes para los estudios técnicos, cuyo origen se situaría en las etapas de educación primaria y secundaria y cuya consecuencia sería un temor (más o menos justificado y más o menos consciente) a fracasar en este tipo de carreras, con el consiguiente rechazo hacia las mismas. Además, en el caso concreto de la Informática, se ha detectado que existe una imagen bastante negativa de la naturaleza de la labor del profesional en Informática, e incluso del carácter del propio profesional. En efecto, a menudo la percepción que muchas personas (potenciales estudiantes incluidos) tienen del ingeniero informático se ajusta más a visiones tópicas y distorsionadas que a la realidad: personas eternamente sentadas frente al ordenador, obsesionadas con todo lo informático, con un comportamiento asocial, etc.. Para colmo, esta idea estereotipada del ingeniero informático como un “bicho raro” (en este sentido, no es extraño el empleo de vocablos ingleses más o menos despectivos como “nerd”, “geek”, “freak”, etc.) se ha visto amplificada habitualmente por medios como el cine y la televisión, de gran influencia en la sociedad en general y en los jóvenes en particular.

Aunque es cierto que el temor de los alumnos a no estar preparados para cursar titulaciones de Informática debería comenzar a combatirse desde etapas preuniversitarias (por ejemplo, mediante una mayor presencia de materias relacionadas con la Informática en los currícula de primaria y secundaria), no es menos cierto que desde la propia Universidad también se pueden tomar acciones que solucionen (o al menos atenúen) este fenómeno. En este sentido, por ejemplo, sería fundamental mostrar claramente a los potenciales estudiantes en qué consisten realmente los estudios universitarios de Informática, cuáles son las especialidades de las distintas titulaciones, etc. Análogamente, desde la Universi-

dad pueden emprenderse acciones orientadas a mejorar la imagen del trabajo de ingeniero informático, que consistirían principalmente en divulgar y enfatizar aquellos aspectos positivos de las profesiones relacionadas con la Informática que habitualmente se omiten, se desconocen o se malinterpretan (por ejemplo, la capacidad de comunicación interpersonal que debe existir entre los miembros de un equipo que desarrolla un proyecto informático, o entre un profesional informático y sus posibles clientes).

Teniendo en cuenta lo anterior, parece claro que es necesario que desde la Universidad se realice una activa labor de promoción de las titulaciones de Informática a varios niveles, que se concretaría en distintas acciones orientadas en general a “publicitar” (sobre todo entre potenciales estudiantes) la realidad tanto de los estudios de Informática como de las profesiones asociadas a los mismos. Muchas escuelas y facultades han emprendido iniciativas en este sentido, tales como jornadas de puertas abiertas, visitas organizadas de alumnos de secundaria a los centros universitarios, visitas promocionales de profesores universitarios a los centros de secundaria, talleres y cursos abiertos a estudiantes de secundaria, ciclos de cine “informático”, etc. El presente trabajo se centra en una de estas iniciativas, concretamente en la Olimpiada de Informática [7] para estudiantes de secundaria organizadas por la Universidad de Castilla-La Mancha (UCLM), que este año 2011 celebra ya su quinta edición.

Básicamente, se trata de una competición donde equipos formados por alumnos de centros de secundaria de la región, divididos en dos categorías (Bachillerato y Ciclos Formativos) deben realizar sucesivas pruebas consistentes bien en resolver cuestiones o problemas sencillos sobre diversas temáticas informáticas (programación, por ejemplo), bien en realizar algún trabajo que requiera cierto dominio de herramientas informáticas (ofimática, etc.). Las distintas pruebas se realizan on-line en una primera fase en la que participan todos los equipos, que suman más o menos puntos en cada prueba dependiendo de su habilidad para superarla. Una vez concluida esta primera fase, los equipos con mejor puntuación se clasifican para las pruebas finales de cada categoría, que se realizan de forma presencial en dependencias de la UCLM y cuyo resultado decide las clasificaciones definitivas. Cabe destacar la importancia del seguimiento de la competición por parte de los me-

dios de comunicación de la región, incluyendo la difusión de las clasificaciones parciales y final, lo cual supone un aliciente para los equipos y contribuye a popularizar esta Olimpiada. Por otra parte, también debe mencionarse que para un buen desarrollo de esta competición resulta imprescindible la colaboración de profesores de los centros de secundaria que tutoricen y supervisen directamente la labor de los equipos participantes, especialmente la realización “honestá” de las pruebas on-line.

En suma, con esta iniciativa se pretende que los estudiantes de secundaria participantes entren en contacto, activa y amablemente, y a un nivel necesariamente elemental, con la auténtica naturaleza de los estudios superiores de Informática. Al mismo tiempo, la Olimpiada sirve para hacer más “visibles” las titulaciones de Informática que se imparten en la UCLM a estos potenciales alumnos y a la sociedad de la región en general, de acuerdo con la necesidad de promocionarlas mencionada anteriormente.

En este punto, cabe recordar que competiciones similares (centradas en la Informática o en otras materias) se han venido organizando en otras universidades y regiones desde hace tiempo, por lo que la nuestra no es una iniciativa totalmente pionera. Sin embargo, creemos que las experiencias acumuladas en la realización de las ediciones celebradas hasta la fecha pueden resultar interesantes para la comunidad de docentes universitarios de Informática, y por esta razón hemos tratado de compartir dichas experiencias mediante su reflejo en el presente trabajo.

Además, nos gustaría que este trabajo contribuyera a despertar el interés en definir un marco global para desarrollar una Olimpiada Informática a nivel nacional que incluyera pruebas centradas en varias facetas de la Informática. Esta estructura común integraría todas las iniciativas locales o regionales que se están llevando a cabo y permitiría, al igual que en otras disciplinas (como Matemáticas o Física), establecer una competición jerárquica con unas fases locales que permitan a los mejor clasificados acceder a una fase nacional, y de ahí a la internacional.

El resto del artículo se organiza del siguiente modo: en la Sección 2 se describen otras iniciativas relacionadas con ésta en el resto de España. En la Sección 3 se describe ampliamente la Olimpiada de Informática de Castilla-la Mancha, centrándonos principalmente en la estructura organizativa de la misma y en la evolución que ha experimentado. Por últi-

mo en la Sección 4 presentamos algunos resultados obtenidos con la Olimpiada, junto con unas líneas sobre actividades planeadas para el futuro.

2. Otras competiciones orientadas a la promoción de los estudios de Informática

La Facultad de Informática de la Universidad de Murcia [8] ha organizado hasta la fecha tres ediciones de la Olimpiada Informática de la Región de Murcia para alumnos de secundaria (OIRM) [1]. En su caso la olimpiada consta de dos concursos: un concurso de proyectos informáticos, y un concurso presencial de resolución de problemas. El segundo de ellos incluye, además, una fase previa de preparación on-line. La participación en ambos concursos ha ido creciendo de manera considerable y progresiva desde el inicio de la Olimpiada. Si se considera la preparación on-line, en 2008 se inscribieron 44 equipos, en 2009 fueron 169, y en 2010 fueron 384. La participación presencial también se ha duplicado cada año, pasando de los 44 alumnos en total en la edición de 2008, a los 77 alumnos en 2009, y los 148 alumnos en 2010 de toda la Región de Murcia.

El Departamento de Ingeniería en Sistemas Informáticos y Telemáticos de la Universidad de Extremadura (UEX) ha organizado la I Olimpiada Informática de Extremadura [6], celebrada el 12 de marzo de 2011 en la Escuela Politécnica de Cáceres. Esta Olimpiada extremeña se ha planteado como objetivo difundir y promocionar la Informática, y en especial la actividad de programación de ordenadores, entre los estudiantes extremeños de secundaria. Para ello, las competiciones se han orientado a potenciar el uso de los ordenadores como herramientas para resolver problemas, fomentar el ingenio y la creatividad de los estudiantes, aumentar su interés por estos estudios y acercarlos la Universidad de Extremadura. Como es lógico, el certamen ha estado dirigido a todos los alumnos de secundaria matriculados durante el curso académico 2010/2011 en alguno de los centros extremeños. Concretamente, esta Olimpiada consta de dos concursos independientes, *Programación y Lógica*, pudiendo los alumnos participar en tan sólo uno de ellos o en los dos. Ambos concursos consisten en una prueba individual y presencial a celebrar en los laboratorios del mencionado centro.

La Universidad de Valladolid [9] tiene mucha ex-

periencia en organizar concursos de programación basados en un juez on-line pero, hasta donde nosotros sabemos, centrados en estudiantes universitarios. En concreto, sus alumnos han participado en las 16 últimas ediciones del Concurso Internacional de Programación de la ACM (Association for Computing Machinery), sólo para alumnos universitarios, denominado ACM-ICPC Contest (o simplemente ACM Contest) [2], obteniendo en general buenos resultados, que culminaron con la presencia en la final del año 2001. De hecho, fue la primera universidad española en conseguir tan difícil objetivo. Además, organizó la fase regional del Sudoeste de Europa (SWERC) de este concurso en los años 1999 y 2000, corriendo con el peso de la organización el Departamento de Informática de la ETS de Ingeniería Informática.

Por último, la Olimpiada Informática Española (OIE) [4] es un concurso de programación para estudiantes de secundaria de ámbito estatal que se lleva celebrando desde el año 1998. La IOI (International Olympiad in Informatics) [3], una de las cinco olimpiadas científicas internacionales reconocidas y subvencionadas como tales por la UNESCO, reconoce a la OIE como la prueba que determina la selección de los 4 representantes españoles para participar en dicha olimpiada internacional. El organismo que impulsó la celebración de la OIE fue la Fundación Aula [5], fundación asociada a la escuela privada Aula Escuela Europea, con sede en Barcelona. Desde el 2007 la OIE es organizada conjuntamente por la Universidad Politécnica de Cataluña y la Fundación Aula. Desde entonces, la OIE consta de dos fases: una fase clasificatoria on-line, abierta a todos los centros educativos de secundaria españoles, donde se selecciona a los mejores 25 estudiantes de toda España para que participen en una fase final presencial. En la actualidad, desde la Olimpiada Informática Española se organizan 10 concursos on-line de programación anuales, se ofrecen correcciones automáticas de 200 problemas de programación, y se da servicio a 1250 usuarios registrados.

Así pues, hasta donde nosotros sabemos, no existe ningún certamen organizado a nivel nacional que se centre en la difusión de los estudios de Informática y que, al mismo tiempo, cubra varios aspectos de la Informática y no sólo la parte de programación. Aun siendo importante dicha parte, en nuestra modesta opinión la Informática es algo más que

programar, y de ahí que, cuando surgió la idea de organizar una Olimpiada de Informática en Castilla-La Mancha, nuestro planteamiento siempre fue que el certamen cubriera varios aspectos relativos a esta disciplina. En la siguiente sección se explica detenidamente lo que hemos venido desarrollando en este sentido en los últimos años desde la Universidad de Castilla-La Mancha.

3. La Olimpiada de Informática de Castilla-La Mancha

La Olimpiada de Informática de Castilla-la Mancha nació hace 5 años con el objetivo general de introducir a los estudiantes de secundaria en el mundo de la Informática, y especialmente para presentarles los estudios de Informática como una alternativa interesante a tener en cuenta a la hora de elegir sus estudios universitarios.

En este sentido, desde el primer momento se intentó que el certamen sirviera para desterrar la idea que muchas veces se tiene de que los estudios de Informática son bastante duros de superar. Para ello se plantearon unas pruebas de dificultad gradual, de tal modo que la mayoría de los inscritos fuesen capaces de resolver fácilmente al menos el 75 % de los problemas propuestos, mientras que el otro 25 % serviría para poder establecer realmente una clasificación entre los participantes.

Por otra parte, como ya hemos indicado, queríamos evitar ofrecer una visión sesgada de la Informática, por lo que el certamen se planteó para cubrir diferentes áreas de la Informática (sistemas operativos, ofimática, programación, etc.), lo que llevó a estructurar la competición en tres módulos diferentes.

Otro de los objetivos de esta competición era y es transmitir el valor del trabajo en grupo, por lo que, a diferencia de otros certámenes de este tipo, se decidió que la competición se desarrollara por equipos de tres personas.

Aunque la Olimpiada se organiza desde la Universidad, va dirigida a alumnos de secundaria, por lo que era muy importante la existencia de una comunicación fluida y estrecha entre los dos ámbitos educativos. Esta comunicación se ha conseguido, por un lado, con la incorporación al Comité Organizador de profesores de secundaria que aportan su conoci-

miento de estos estudios y de las materias de Informática allí impartidas, y, por otro, involucrando a los profesores de los institutos que deben ejercer una tarea de coordinación, supervisión y acompañamiento de los participantes.

3.1. Organización de la Olimpiada

La Olimpiada de Informática de Castilla-la Mancha está estructurada en dos competiciones independientes. La primera de ellas va dirigida a estudiantes de Bachillerato, mientras que la segunda va dirigida a estudiantes de Ciclos Formativos de Grado Medio y Superior. Esta división viene dada por el hecho de que el nivel y las materias de Informática en ambos estudios son completamente diferentes. Cada una de estas dos competiciones ha sido dividida en tres módulos independientes, cada uno de los cuales se centra en un aspecto de la Informática. Los participantes deben trabajar cada uno de los módulos durante un periodo de aproximadamente mes y medio y, al finalizar este tiempo, disponen de una semana para la realización de una prueba sobre ese módulo que permita establecer la clasificación de los equipos. Estas pruebas son realizadas on-line sobre la plataforma Moodle. A continuación describiremos cada una de las dos competiciones.

Olimpiada de Bachillerato Esta competición va dirigida principalmente a alumnos de Bachillerato, pero no se impide participar a aquellos alumnos de los últimos cursos de la ESO que lo deseen. Atendiendo a los contenidos de Informática que se imparten en estos niveles¹, los tres módulos que se plantean para esta competición son los siguientes:

1. **Informática básica.** En este módulo se pretende que los participantes trabajen conceptos básicos de Informática. Concretamente, el módulo se centra en la composición de un ordenador, representación de la información y la utilidad y uso de los sistemas operativos. Tras un trabajo de preparación de aproximadamente 4 meses los participantes deben resolver un cuestionario on-line de 25 preguntas de respuesta múltiple.
2. **Ofimática.** El hecho de que la ofimática sea una de las áreas más conocidas y utilizadas de la

¹Consensuado con profesores de algunos institutos que están integrados dentro del Comité Organizador.

Informática hace casi obligatorio que esta área forme parte de las tratadas dentro de esta competición. El carácter meramente práctico que tiene esta faceta de la Informática ha marcado el tipo de prueba que se hace para la valoración de este módulo. En este caso la prueba que deben realizar los participantes es el desarrollo de tres casos de estudio muy simples. Cada uno de ellos es resuelto con la utilización de una de las herramientas ofimáticas más usadas (procesador de texto, hoja de cálculo y base de datos). Para la realización de estos casos pueden usarse tanto el paquete ofimático de Microsoft como el paquete ofimático OpenOffice.

3. **Programación.** Es el módulo que más alejado queda de los conocimientos de Informática que un alumno de Bachillerato normalmente adquiere en sus estudios, pero, dado que es una de las áreas fundamentales de la Informática, se consideró que debía formar parte de la Olimpiada de Informática. Dentro de este módulo se pretende que los participantes se desenvuelvan a nivel bajo en la programación. Por ello, no se les exige el uso de ningún lenguaje de programación concreto y se les pide que los programas los hagan en un lenguaje de pseudocódigo. Al finalizar el módulo la prueba que deben resolver los alumnos está formada por dos partes, una primera en la que deben contestar un cuestionario formado por 20 preguntas de opción múltiple y luego una parte práctica en la que deben escribir un programa para la resolución de un problema relativamente simple.

Olimpiada de Ciclos Formativos Esta competición va dirigida a alumnos matriculados en Ciclos Formativos tanto de Grado Medio como Superior. Estos participantes tienen unos conocimientos más profundos de Informática, ya que sus estudios son específicos del área. Esto ha hecho que los módulos planteados para esta competición sean diferentes a los de la competición de Bachillerato. No sería justo mezclar en una misma competición a estudiantes de Bachillerato con estudiantes de Ciclos, pues podría producirse un efecto de desánimo entre los estudiantes de Bachillerato que los alejase de nuestros estudios al verse superados por los estudiantes de Ciclos, que es justamente lo contrario de lo que se desea conseguir con la Olimpiada.

Los módulos planteados para la competición de Ciclos Formativos son los siguientes:

1. **Sistemas operativos.** Este módulo se divide en tres subapartados. En el primero los participantes deben familiarizarse con conceptos generales relacionados con los sistemas operativos. Cada uno de los otros dos subapartados se centra en uno de los dos sistemas operativos más utilizados hoy en día: Microsoft Windows y Linux. La prueba final del módulo consiste en un cuestionario de 30 preguntas de elección múltiple que los alumnos deben contestar en un tiempo determinado.
2. **Bases de datos.** En este módulo los participantes deben mostrar su habilidad para la creación y gestión de bases de datos. Para ello se utiliza el lenguaje de gestión de base de datos estándar SQL. Tras el trabajo en el módulo, los participantes deben enfrentarse a una prueba que está formada por un cuestionario de 20 preguntas de elección múltiple y por un caso práctico en el que deben crear y gestionar una base de datos.
3. **Programación.** Al igual que en la competición de Bachillerato, los participantes deben enfrentarse a un módulo de programación. Durante este módulo los alumnos deben ser capaces de realizar programas para la resolución de todo tipo de problemas. Los participantes tienen la posibilidad de realizar sus programas en un lenguaje imperativo tradicional como es C o en un lenguaje imperativo orientado a objetos como es Java. La prueba de este módulo, al igual que para los participantes de Bachillerato, está compuesta por un cuestionario de 20 preguntas de elección múltiple y por una parte práctica en la que los participantes deben realizar un programa, en uno de los lenguajes indicados, que resuelva el caso práctico planteado.

3.2. Desarrollo temporal de la competición

Cada año, con el inicio del curso académico empieza una nueva edición de la Olimpiada de Informática de Castilla-la Mancha con la inscripción de los equipos, que como se ha comentado anteriormente están formados por 3 componentes bajo la dirección de un profesor del centro al que pertenece el equipo. Tras la inscripción, a cada uno de los equipos se les proporciona un usuario y clave con la que poder

acceder a la plataforma *Moodle* donde, progresivamente, se publican todos los materiales de preparación de los diferentes módulos. Los equipos pueden inscribirse hasta el día de inicio de la prueba correspondiente al primer módulo, que suele realizarse la primera semana después de las vacaciones de Navidad. Finalizada la prueba de este primer módulo, los responsables a cargo de su organización publican la clasificación de cada uno de los grupos con la puntuación obtenida, y se inicia el siguiente módulo con la publicación de los materiales para su preparación.

La preparación de este segundo módulo se desarrolla hasta la realización de su prueba final, que tiene lugar en las primeras semanas del mes de marzo, siempre antes de las vacaciones de Semana Santa. Tras la realización de la prueba de este segundo módulo su puntuación es añadida a la obtenida por los equipos en la prueba anterior, y así se obtiene una nueva clasificación.

De forma similar se hace con el tercer módulo, cuya prueba se realiza en la última quincena del mes de abril, y cuya puntuación también es sumada a la acumulada por los equipos hasta ese momento, para así obtener la clasificación final de la fase on-line.

Así pues, una vez realizadas las pruebas on-line de los tres módulos se establece la clasificación final de los equipos, y con ello concluye la fase on-line de la competición y se inicia la fase presencial.

Dicha fase presencial se desarrolla un día concreto de la primera semana de mayo en los laboratorios de las escuelas de Informática involucradas en la organización. A esta fase presencial asisten los mejores equipos (alrededor de diez) de cada una de las dos competiciones. El número de equipos no se fija desde el inicio, sino que se fija en función de la puntuación final de la fase on-line, estableciendo un corte allí donde exista una diferencia de puntuación que se considere significativa. De esta forma se evita que posibles anomalías en la fase previa dejen fuera de la final a algún equipo sólo por cuestión de unas décimas.

La prueba presencial tiene una estructura similar a las pruebas on-line. Concretamente, en una primera parte los participantes tienen que resolver un cuestionario de preguntas de elección múltiple donde el 50% de las preguntas son del primer módulo, el 25% del segundo módulo y otro 25% del tercer módulo. La segunda parte es práctica, y en ella los participantes tienen que resolver un caso práctico co-

rrespondiente al segundo módulo, y otro correspondiente al tercer módulo.

Tras calificar la prueba presencial, se calcula la puntuación final de los equipos, en la que el 25% de la misma corresponde a la puntuación de la fase on-line y el 75% a la puntuación obtenida en la prueba presencial.

3.3. El papel de los tutores

Si importante es el papel de los participantes en el desarrollo de la Olimpiada, no menos importante es el papel que juegan los tutores en el desarrollo de la misma, ya que ellos son a la vez participantes y organizadores.

Como participantes, son los encargados de animar a sus alumnos a inscribirse en la Olimpiada y de ayudarles a preparar cada uno de los módulos durante sus clases. Además, son considerados los responsables de los grupos y como tales deben encargarse de la inscripción de los diferentes grupos.

Como organizadores, su tarea consiste en velar por el correcto desarrollo de las pruebas on-line, ya que debido a la gran extensión de la comunidad es imposible que esta tarea la realicen los miembros del Comité Organizador. También recae sobre los tutores de los equipos clasificados para la fase presencial la organización del desplazamiento de sus equipos hasta la sede de la fase presencial, solicitando los permisos correspondientes, buscando la financiación necesaria y acompañando a sus alumnos en el desplazamiento.

Esta ardua tarea que desarrollan los tutores se ve mínimamente recompensada, por un lado, por la satisfacción personal que obtienen de un trabajo bien hecho y, por otro, por el reconocimiento por parte de las autoridades educativas de la Comunidad Autónoma de un punto en el concurso de traslados.

Conscientes de la importancia que tiene el papel que desarrollan estos tutores en el correcto funcionamiento de la Olimpiada, al finalizar la prueba presencial se organiza una reunión entre el Comité Organizador del certamen y los tutores de los equipos que han participado en el mismo. En esta reunión se hace un análisis en profundidad de la edición finalizada, para intentar corregir los problemas que se hayan podido plantear, y se planifica el calendario de las pruebas de la siguiente edición.

3.4. Estadísticas de la Olimpiada

A lo largo de las sucesivas ediciones de esta Olimpiada se ha podido observar un significativo aumento en el número de participantes, tanto en alumnos de Bachillerato como de Ciclos Formativos, así como de sus tutores.

La gráfica de la figura 1 muestra la progresión de participantes en la modalidad de Bachillerato, mostrando el número de equipos inscritos a lo largo de las cuatro primeras ediciones, así como el número de equipos que llegaron a la prueba final que se realizaba de forma presencial en las instalaciones de la propia Universidad. Como puede observarse en la gráfica, el número de equipos finalistas no es fijo sino variable, y también crece en las últimas ediciones. Esta variabilidad es debida a la forma en la que los equipos son seleccionados para esa prueba final, basado por supuesto en las puntuaciones obtenidas en las fases anteriores, pero dejando el corte a criterio del Comité Organizador. Como ya se ha indicado, dicho criterio no es arbitrario, sino razonado, ya que se establece el corte allí donde se observa una mayor diferencia de puntuación entre los primeros equipos y los siguientes, con el objeto de invitar a la prueba a todos aquellos equipos que merezcan un puesto en la misma.

El gráfico de la figura 2 presenta los datos correspondientes a la modalidad de Ciclos Formativos, en la cual participan menos equipos si la comparamos con la modalidad anterior, pero que igualmente ha experimentado un notable aumento de participación a lo largo de las distintas ediciones. En relación al número de equipos finalistas, sirve aquí la misma explicación dada para la modalidad de Bachillerato.

Por último, en la figura 3 se muestran los datos de la Olimpiada completa (esto es, el resultado de sumar los datos presentados en las dos anteriores), que, como puede observarse, corroboran la tendencia al alza de la que se ha hablado. Así, de 64 equipos que participaron en la primera edición se ha pasado en cuatro años a 184. Sumando todos los equipos se obtiene, además, un total de 529 equipos participantes, si bien cabe apuntar que algunos equipos han repetido participación de un año para otro, en algunos casos cambiando alguno o algunos de los miembros que lo componían. En cualquier caso, y considerando que cada equipo estaba formado por 3 alumnos, puede decirse que alrededor de 1500 estudiantes han

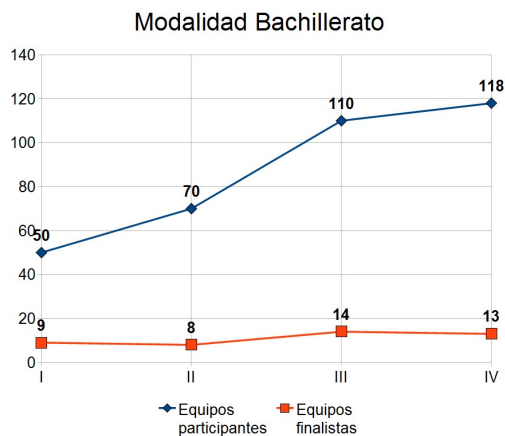


Figura 1: Evolución de participantes en la modalidad de Bachillerato.

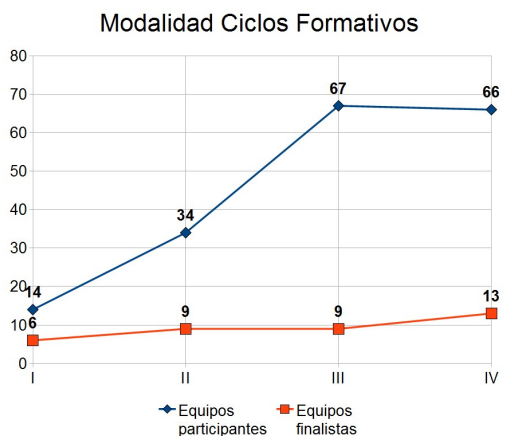


Figura 2: Evolución de participantes en la modalidad de Ciclos Formativos

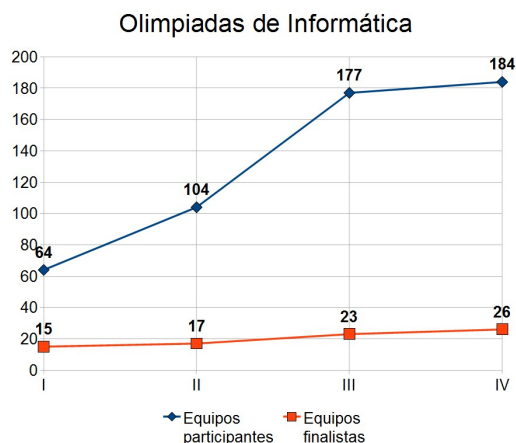


Figura 3: Evolución total de participantes en las Olimpiadas de Informática de Castilla-La Mancha.

tomado parte en alguna de las competiciones organizadas.

Por último, y aunque no es un objetivo de la Olimpiada, también ha crecido el número de profesores que forman parte del Comité Organizador, lo que da una idea del interés creciente de este proyecto no sólo fuera, sino también dentro de la propia Universidad. Sirva de ejemplo que la primera edición arrancó con sólo 8 profesores, pero que en la quinta, y actual, ya suman 25.

Aunque no disponemos de datos rigurosos de cuántos alumnos que han participado en alguna edición de las Olimpiadas de Informática han terminado estudiando Informática o alguna otra carrera técnica, por conversaciones particulares que los miembros del Comité Organizador hemos mantenido con muchos de ellos, puede afirmarse que la mayoría considera su participación en esta competición como una experiencia positiva, y son bastantes los que efectivamente han terminado estudiando Informática o alguna otra carrera técnica. En particular, ha habido algunos estudiantes que han reconocido que su

participación en la Olimpiada de Informática les hizo decidirse por estos estudios cuando previamente no lo habían considerado.

4. Conclusión y trabajo futuro

La Olimpiada de Informática de Castilla-la Mancha, que surgió como una experiencia piloto de promoción de los estudios de Informática, ha conseguido consolidarse entre los alumnos y profesores de secundaria en tan sólo 5 ediciones, y se ha mostrado como un método eficaz de acercamiento de los alumnos de secundaria al mundo de la Informática.

De cara a un futuro próximo, desde nuestro punto de vista, sería interesante la unión de las diversas universidades para crear una competición a nivel nacional, complementaria a la actual Olimpiada Informática Española (OIE), en la que, al igual que ocurre en ésta de Castilla-la Mancha, se abarcasen las diferentes áreas que actualmente no tienen cabida en la OIE y que también son parte integral de la Informática.

Referencias

- [1] F. de Informática de la Universidad de Murcia. Olimpiada informática de la región de murcia para alumnos de bachillerato y secundaria, <http://olimpiada.inf.um.es/>.
- [2] <http://cm.baylor.edu/welcome.icpc>. The acm international collegiate programming contest (icpc).
- [3] <http://ioinformatics.org/index.shtml>. International olympiad in informatics.
- [4] <http://olimpiada-informatica.org/>. Olimpiada informática española.
- [5] <http://www.fundacioaula.org/>. Fundación privada aula.
- [6] <http://www.oieex.org/#/page1.do>. Olimpiada informática de extremadura.
- [7] <http://www.olimpiadasinformatica.uclm.es/>. Olimpiadas de Informática de la Universidad de Castilla-La Mancha.
- [8] <http://www.um.es/>. Universidad de Murcia.
- [9] <http://www.uva.es/>. Universidad de Valladolid.

Experiencia Docente en Informática Aplicada a la Traducción I del Grado de Traducción e Interpretación de la Universidad de Murcia

Antonio Ruiz Martínez y Mercedes Valdés Vela
Departamento de Ingeniería de la Información y las Comunicaciones
Facultad de Informática - Universidad de Murcia
30100 Campus de Espinardo - Murcia
arm,mdvaldes @um.es

Resumen

En este artículo describimos los principales aspectos docentes relacionados con la asignatura de Informática Aplicada a la Traducción I (IAT I) del Grado de Traducción e Interpretación de la Universidad de Murcia. En particular, en esta asignatura se tratan aspectos de Informática general que constituyen la base para que, en un curso posterior los alumnos cursen adecuadamente IAT II, asignatura más específica, centrada en sistemas de traducción asistida y localización. En el artículo se describen los contenidos abordados, la metodología seguida, la forma de evaluación, la encuesta realizada al principio de curso para evaluar los conocimientos iniciales así como la encuesta final y los resultados obtenidos.

Summary

In this paper we describe the main teaching issues related to the subject called Computer Science applied to Translation I (CSAT I) that belongs to the Translation and Interpretation Degree of the University of Murcia. Particularly, in this subject we deal with the main issues related to Computer Science that are needed so that, subsequently, pupils can carry out a more specialized subject called CSAT II, where they will work with computer-assisted translation and localization tools. In this paper, we describe the contents dealt with, the methodology followed, evaluation mechanisms, the poll taken at the beginning of the subject in order to evaluate the initial knowledge of the pupils along with the final poll and, finally, the results obtained.

Palabras clave

enseñanza, informática, traducción

1. Introducción

La implantación del Espacio de Enseñanza Superior (EES) junto con la aparición de los nuevos grados y másteres, ha supuesto el diseño y/o rediseño de los planes de estudios de las titulaciones universitarias. El diseño de los nuevos grados y másteres ha implicado una revisión de los contenidos hasta ahora impartidos en las distintas titulaciones. En nuestra opinión, este hecho ha sido tremendamente positivo ya que ha obligado a analizar, desde un punto de vista crítico, qué estamos enseñando y cómo lo estamos enseñando. Así, este análisis ha permitido detectar, tanto los puntos fuertes, como las posibles carencias en los planes de estudios existentes hasta la fecha. De este modo, durante el diseño de los nuevos planes se han tenido en cuenta las lecciones aprendidas de los planes pasados, manteniendo los puntos positivos, superando los distintos problemas encontrados, e incorporando contenidos para cubrir las necesidades del mercado laboral.

En los estudios de Traducción e Interpretación (TeI) (Francés e Inglés) de la Universidad de Murcia, este proceso de adaptación al EES tuvo lugar el curso 2009/2010. Hasta el curso 2008/2009, se trataba de dos licenciaturas. El nuevo grado se diseñó a la luz del Libro Blanco de Traducción e Interpretación [1], las necesidades del ámbito profesional y de mercado global [3] y regional de la Comunidad Autónoma de Murcia en particular, y la opinión de expertos del mundo de la traducción, algunos procedentes de organismos de la Unión Europea.

La materia de Informática es fundamental en la formación de los futuros traductores e intérpretes tal como refleja el Libro Blanco de la titulación. En el diseño de la materia de Informática del grado se realizó entre distintos profesores de Informática que impartían docencia en la licenciatura (entre ellos uno de los autores). La experiencia adquirida

Asignatura	Informática Aplicada a la Traducción I	
ECTS	6	
Curso	1º	
Titulación	GTI Inglés	GTI Francés
Alumnos	66	35
Grupos prácticas	3	1

Cuadro 1: Datos descriptivos

a lo largo de los años en las licenciaturas de Tel nos ha permitido conocer las necesidades de los alumnos de esta carrera. Así, hemos tratado de no cometer los errores que se suelen dar en las asignaturas relacionadas con la Informática cuando están inmersas en carreras pertenecientes a otras disciplinas [2]. Por ejemplo, se suele caer en el error de dar más importancia a la enseñanza teórica que a la práctica. Sin embargo, lo más útil para los futuros profesionales es adquirir experiencia en el manejo de ciertas herramientas informáticas. Otro error es la lenta renovación de contenidos. En asignaturas eminentemente teóricas, es permisible renovar los contenidos con una menor frecuencia ya que los conceptos fundamentales de Informática cambian más lentamente. Pero cuando la práctica adquiere mayor protagonismo, los contenidos deben renovarse en paralelo con el avance de la tecnología y, en particular, del software asociado al ejercicio de la correspondiente profesión.

Esta materia ha sido dividida en dos asignaturas: Informática Aplicada a la Traducción (IAT) I e IAT II. IAT I se imparte en el primer cuatrimestre (siendo el segundo año que se imparte) mientras que IAT II se imparte en el sexto cuatrimestre (se impartirá por primera vez el próximo curso). En la Tabla 1 se muestran los datos descriptivos de la asignatura.

En este artículo nos centramos en la asignatura IAT I. De modo que describiremos sus contenidos, actividades y metodología, presentaremos los resultados obtenidos y discutiremos los distintos problemas que han surgido así como las estrategias que, en el futuro, nos pueden ayudar a superarlos.

2. Perfil del Alumnado de la Titulación

Los alumnos que solicitan matricularse en este grado son de dos tipos principalmente. En su mayoría provienen de bachillerato. A este respecto debemos destacar que la nota de corte para acceso al grado en

el cupo general en inglés se quedó en 10,313 puntos al ser esta especialidad más demandada que la de francés, que se quedó en 6,840 puntos.

Por otro lado, hay un buen porcentaje de alumnos egresados de otras carreras relacionadas con los idiomas. La mayoría de éstos simultanean los estudios con el trabajo.

Con el objetivo de determinar el nivel de informática con el que entran los alumnos a la titulación, al inicio de curso los alumnos rellenaron una encuesta. En ella les preguntábamos acerca de sus conocimientos sobre el sistema operativo, el uso de herramientas ofimáticas, correo electrónico, herramientas groupware, manejo de gráficos, así como si había cursado en los estudios previos alguna asignatura de informática.

De los resultados de la encuesta podemos mencionar que, como ya se preveía, la mayoría de los alumnos están acostumbrados al manejo del correo electrónico o herramientas groupware orientadas al ocio como las redes sociales. En cuanto al manejo de herramientas ofimáticas, la mayoría de ellos tienen unos conocimientos de nivel básico-medio de procesadores de textos y herramientas de diseño de presentaciones (el 24% declaraba sólo haber utilizado el procesador de textos). Sin embargo, la mayoría había utilizado muy poco o nada herramientas de manejo de hojas de cálculo. Además sólo el 20% había cursado estudios de informática en el instituto aunque un amplio número de ellos (66%) había realizado cursos de informática por cuenta propia.

3. Objetivos de la Asignatura

IAT I tiene un objetivo triple:

1. Servir de base para que, posteriormente, en la asignatura IAT II, los alumnos se dediquen exclusivamente al manejo de las herramientas específicas de traducción asistida por ordenador y al software de localización.
2. Dotar a los alumnos de conocimientos informáticos que les serán útiles durante el propio grado para, por ejemplo, realizar trabajos de otras asignaturas.
3. Proporcionar a los alumnos una visión general de distintas herramientas informáticas de propósito general que pueden usar

habitualmente en un entorno profesional.

En las siguientes secciones describimos los distintos aspectos relacionados con la asignatura así como una discusión sobre la experiencia durante su impartición.

4. Competencias de la Asignatura

Las competencias específicas que cubre la asignatura son:

- Desarrollar tareas de documentación y búsqueda de información.
- Utilizar herramientas informáticas de trabajo en grupo para trabajar en equipo.
- Manejar con propiedad la terminología básica de la Informática, en especial, aquellos términos relacionados con el almacenamiento de información.
- Distinguir entre la parte física de un ordenador (hardware) y la parte lógica (software).
- Describir las funcionalidades de los componentes de un ordenador y explicar el funcionamiento básico del mismo.
- Clasificar los ordenadores teniendo en cuenta su potencia y tamaño y su propósito.
- Describir algunos de los tipos de información que se pueden almacenar en un fichero y establecer la correspondencia con los diferentes formatos de fichero y con los programas capaces de manejarlos.
- Describir las bases de algunos de los esquemas de codificación de caracteres más extendidos para lograr destreza en el manejo de ficheros de texto en diferentes idiomas.
- Diferenciar los elementos que intervienen en una red de ordenadores y explicar el funcionamiento básico y los componentes de algunos servicios de red.
- Diseñar hojas de cálculo para manipular datos calculados mediante fórmulas de usuario, funciones predefinidas, referencias, gráficos y demás elementos que ayuden a la generación automática de información asociada al ejercicio de la profesión de.
- Elaborar documentos de texto, con índices de contenidos, referencias cruzadas, números de página, cabeceras, y demás elementos útiles para la generación de memorias bien

organizadas y bien presentadas con ayuda de herramientas de procesamiento de textos.

- Diseñar sitios Web sencillos y bien estructurados que satisfacen las características de internacionalización y localización.
- Estructurar información en diferentes idiomas con el apoyo de lenguajes de etiquetas como XML útil para entender formatos típicos de intercambio de información entre herramientas de traducción asistida.
- Definir sistema de traducción asistida por ordenador e identificar sus elementos.

5. Metodología y Actividades Realizadas

Las actividades que se realizan a lo largo del curso se dividen en presenciales (40%) y no presenciales (60%). Los tres tipos de actividades presenciales son las clases expositivas, las sesiones de prácticas y los seminarios. Una de las cuatro horas semanales se dedica a clase expositiva al grupo completo (de forma separada inglés y francés). En ellas los profesores explican los contenidos teóricos.

Dependiendo del contenido teórico abordado durante esa semana, las tres horas restantes pueden ser sesiones de prácticas o seminarios. En ambos tipos de actividades los alumnos están repartidos en grupos de tamaño 30. Con independencia de la titulación a la que pertenezcan, los alumnos pueden escoger el grupo cuyo horario se adapte mejor a sus necesidades. No obstante, hay un grupo cuyo horario se adapta mejor a aquellos alumnos que justifican tener un contrato de trabajo.

En las sesiones de prácticas exponemos brevemente los conceptos básicos que se necesitan para el posterior desarrollo de una serie de ejercicios en el ordenador. Durante el tiempo restante los alumnos utilizan su PC para realizar, de forma individual, una serie de ejercicios mientras que el profesor los puede supervisar.

Los seminarios son muy parecidos a las clases prácticas en el sentido de que requieren parte de exposición de contenidos y parte de manejo de una herramienta informática. La diferencia fundamental es que los contenidos de los seminarios requieren un porcentaje mayor del tiempo de la sesión se desarrolle en la modalidad expositiva con el objetivo de introducirles en una determinada temática y, a continuación, la mayor parte del trabajo del alumno

se realiza en la modalidad de trabajo autónomo donde, a partir de los contenidos iniciales, tendrán que investigar cómo llegar a la consecución de unos determinados objetivos que se plantean en clase y que se pueden abordar en una sesión posterior. La asistencia a sesiones de prácticas y seminarios, aunque recomendable, no es obligatoria.

Las actividades no presenciales son de dos tipos: boletines entregables de ejercicios prácticos o supuestos entregables. El boletín de ejercicios se suele proponer en la segunda parte de una sesión de prácticas. El alumno podrá resolver esos ejercicios de forma presencial y, si es necesario, trabajará de manera autónoma en casa.

Los supuestos entregables se proponen para que el alumno lleve a la práctica los contenidos que le han sido explicados en los seminarios.

Finalmente debemos comentar que, aparte de las tutorías presenciales, también ofrecemos la posibilidad de tutorías electrónicas ya que facilita y dinamiza el aprendizaje de alumno [9].

Además de estas actividades relacionadas directamente con la docencia, al principio de curso, llevamos a cabo la encuesta inicial voluntaria mencionada en la sección 1 con el fin de determinar el nivel de informática con el que entran los alumnos a la titulación. Al final de la asignatura realizamos otra encuesta voluntaria con el objetivo de mejorar la asignatura en los próximos cursos.

6. Contenidos de la Asignatura

Los contenidos teóricos de la asignatura se dividen en tres grandes bloques. El primer bloque se llama *Fundamentos de la Informática* y está dedicado al equipo informático en sí. En él, además del tema de introducción, encontramos un tema por cada una de las partes fundamentales del equipo informático como son el hardware, la información y el software.

En el tema del hardware introducimos los componentes físicos del ordenador como son el procesador, la memoria principal, el disco duro y los distintos periféricos. Procuramos dar una idea general sobre algunos componentes del ordenador que suelen oírse frecuentemente (placa base, familias de procesadores, etc.). Dado que el sistema operativo actúa de interfaz entre el hardware y el resto del equipo informático, es durante este tema donde introducimos sus aspectos fundamentales.

En el tema dedicado a la información abordamos los conceptos de bit (elemento fundamental de representación de información) junto con el byte y sus múltiplos como medidas de capacidad de almacenamiento de información. También tratamos los conceptos de fichero y formato. No dedicamos mucho tiempo a la codificación de caracteres (ASCII, UNICODE, ...) ya que se ve más adelante con más detalle.

En el tema dedicado al software hacemos una clasificación de los tipos de programas, desde los más cercanos a la máquina como son los sistemas operativos y las utilidades del sistema, hasta los más cercanos al usuario como son las aplicaciones de usuario final. Les prestamos especial interés a las suites ofimáticas y, como colofón del tema damos una visión general de cómo, mediante un caso de uso de pedido de traducción de un determinado documento a una empresa, las herramientas descritas sirven en el proceso de traducción, desde que se recibe el pedido hasta que éste se entrega.

Para afianzar los contenidos teóricos, de manera acompañada se van realizando una serie de sesiones de prácticas que tienen que ver con la teoría abordada. Así, la idea es, una vez impartido el tema de hardware realizar una serie de ejercicios sobre el sistema operativo (fundamentos básicos y tareas simples de administración). Acompañando a los contenidos sobre información hacemos prácticas sobre el manejo de ficheros, aplicaciones asociadas a cada extensión concreta, cambio de idioma del equipo y método de entrada, para trabajar con documentos escritos en distintos idiomas, etc. Cuando estamos impartiendo los contenidos del tema de software, se realizan prácticas sobre generación de documentos PDF, lo cual resulta muy útil para la presentación de documentos de la propia y otras asignaturas.

Especial atención le prestamos al manejo de la suite ofimática Open Office. Concretamente, dedicamos varias sesiones al procesador de textos. Los alumnos vienen, en su mayoría, habiendo manejado las funcionalidades básicas de algún otro procesador. En nuestra asignatura, pretendemos que se acostumbren al uso de índices automáticos, estilos y plantillas para dar a sus documentos un aspecto uniforme durante todas las entregas. Dedicamos también un par de sesiones al manejo de hojas de

cálculo con *Calc*, lo cual les resultará útil a la hora de presupuestar los encargos de traducción que se les presenten en el ejercicio de su profesión, en especial si actúan de traductores *freelance*.

El segundo bloque está dedicado a *Servicios y Herramientas TIC*. En un primer tema explicamos, de manera muy general, los conceptos fundamentales de las redes de ordenadores, mencionando aquellos que se oyen diariamente como, IP, ancho de banda, etc.

En un segundo tema explicamos los servicios telemáticos más útiles y sus herramientas asociadas, desde las comunes como navegadores y correo electrónico hasta las más complejas como aquellas que facilitan el trabajo en grupo o groupware (compartición de documentos, calendarios grupales, creación de encuestas, etc.).

Para afianzar estos contenidos se realizan unas sesiones de prácticas de manejo de herramientas telemáticas y groupware. En ellas trabajan con distintos clientes de correo viendo su configuración, creación de filtros, etc. También se usan herramientas para facilitar el trabajo en grupo como Doodle, Google Calendar o el manejo colaborativo de documentos con Google Docs. Es importante destacar que aparte de enseñarles estas herramientas, el contexto y las actividades que se plantean intentan reflejar sus posibilidades de uso en un entorno profesional.

El último bloque, llamado *Traducción, Interpretación y TICs* sirve para empezar a centrarse en las herramientas más especializadas en el ejercicio de la profesión como traductores e intérpretes. También, dedicamos un tema a introducir los conceptos de internacionalización y localización de sitios Web para lo cual, debemos hablar de los distintos esquemas y normas de codificación de caracteres y de los lenguajes de etiquetas como HTML y XML.

Acompañan a estos contenidos el acceso a sitios Web útiles en la traducción como traductores automáticos, diccionarios on-line, el diseño de un sitio Web y un pequeño seminario donde se estructuran datos con XML. El formato XML es utilizando en algunos de los principales formatos relacionados con herramientas de traducción como XLIFF (localización) o TMX (memorias de traducción) de ahí el incluir XML en los contenidos.

7. Recursos

La asignatura se imparte en dos aulas distintas. Por un lado, el aula de teoría tiene todo lo necesario para el desarrollo de las clases expositivas (pizarra, ordenador con conexión a Internet integrado con el campus virtual y un cañón de vídeo). Por otro lado, el aula donde se desarrollan las prácticas y los seminarios. Esta sala, denominada *aula multimedia*, está dotada de 30 PCs (un alumno por puesto) con componentes específicos para el desarrollo de prácticas propias de la traducción e interpretación. Por último, hay que señalar que en la facultad y el aulario donde se imparten las clases existen ALAs (aulas de libre acceso) a las que los alumnos pueden acceder en todo momento para realizar el trabajo autónomo.

En cuanto a los recursos software, destacamos que el aula multimedia está dotada de un software de control remoto, comunicación y gestión de clase basado en Netop [5] permitiendo que el profesor, desde su puesto, tutorice, controle y supervise las tareas de los alumnos. Así, durante la parte expositiva de la sesión práctica los alumnos ven en sus monitores lo que el profesor está haciendo en su ordenador (mostrando transparencias, usando la aplicación que es el objetivo de la sesión, etc).

Para prácticas y seminarios usamos software libre: la suite ofimática Open Office, Nvu como editor para desarrollar sitios Web, Essential XML Editor para creación de documentos XML, Thunderbird como cliente de correo y Firefox como navegador (aunque también utilizan Outlook y Internet Explorer). La mayoría de estos recursos están también instalados en los PCs de las ALAs. Para el desarrollo de ejercicios sobre herramientas groupware utilizamos herramientas basadas en Web como Google Docs, Doodle y Google Calendar.

Por último, cabe destacar que hacemos gran uso del campus virtual de la universidad, denominado SUMA [6, 7], ya que este tipo de herramientas potencia el aprendizaje de los alumnos [8]. Esta plataforma Web proporciona varias vías de comunicación entre los alumnos y los profesores (tutorías, foros, chat, etc), espacios virtuales de trabajo (repositorios de documentos, tablón, etc), herramientas para el diseño, realización y corrección de exámenes y funcionalidades que facilitan el seguimiento del alumno (calificaciones, actas, etc)

entre otras. Por otra parte, los alumnos disponen de su propia carpeta para la asignatura donde pueden depositar sus trabajos. Finalmente, hay que comentar que uno de las principales funcionalidades que utilizan los alumnos es la tutoría electrónica (implantada tal como se menciona en [10]).

8. Evaluación

Utilizamos distintos instrumentos para evaluar cada una de las partes de la asignatura. Los contenidos teóricos se evalúan mediante un examen de preguntas tipo test y varias cuestiones breves de desarrollo. La calificación de la parte práctica se obtiene mediante la evaluación de los boletines de ejercicios prácticos entregados. La parte dedicada a seminarios se evalúa mediante la entrega de los correspondientes supuestos resueltos. Si el profesor lo considera necesario, convoca al alumno a entrevista para defender tanto su trabajo de prácticas como la resolución del supuesto.

La política de mantenimiento de aprobados parciales es igual en febrero, junio y septiembre. Si el alumno tiene alguna parte aprobada (teoría, práctica o seminarios), se le mantiene durante las convocatorias que resten del curso.

En cuanto a la obtención de la calificación final de la asignatura, ésta sólo se puede calcular si el alumno tiene una calificación en cada parte igual o superior a cuatro en febrero y a cinco en junio y septiembre. En ese caso la calificación final se obtiene mediante una media ponderada acorde con la carga de trabajo que conlleva cada parte, donde la teoría y los seminarios tienen una ponderación del 25% y el mayor peso lo tienen las prácticas con el 50%.

La política de mantenimiento de aprobados parciales expuesta anteriormente hace necesario establecer una política de consumo de convocatorias. En febrero, un alumno se considera presentado a la asignatura sólo si se ha presentado al examen teórico. Por el contrario, si ha presentado los trabajos asociados a las prácticas y/o a los seminarios y no ha asistido al examen se le considera no presentado. Sin embargo, tanto en junio como en septiembre, cualquier parte vincula el consumo de la convocatoria, tanto asistir al examen, como entregar el trabajo correspondiente durante el periodo establecido a los efectos de la convocatoria en curso. A los alumnos que consumen

Grupo	Inglés	Francés
No Presentado	16%	40%
Suspense	14%	28%
Aprobado	18%	12%
Notable	46%	20%
Sobresaliente	6%	0%

Cuadro 2: Calificaciones Convocatoria Febrero (curso 2010/2011).

una convocatoria y no superan la asignatura le otorgamos una calificación que consideramos que es justa al ser alta dentro del suspenso: cuatro puntos.

8.1. Resultados de la Evaluación

En la tabla 2 presentamos los resultados obtenidos en cada uno de los grupos en la convocatoria de Febrero de 2011. Como podemos observar las calificaciones que obtiene el grupo de inglés son bastante superiores a las del grupo de francés. En inglés, la mejor calificación es de 9,4 frente al 7,9 de francés. Esta situación, desde nuestro punto de vista, es coherente con el nivel con el que entran los alumnos en cada una de las dos titulaciones ya que los contenidos, actividades, etc son las mismas para ambos. Para apoyar esta hipótesis, recordar que, como se dijo en la sección 2 la nota de corte para acceso al grado de Traducción e Interpretación en inglés en el cupo general era bastante más alta que la de francés.

Además de los resultados obtenidos mediante las distintas calificaciones decidimos realizar una encuesta para que los alumnos valoraran la asignatura. De los distintos resultados podemos destacar que el 76,8% indica que la asignatura les ha ayudado a mejorar sus conocimientos. A un 84% los contenidos prácticos les han parecido adecuados o muy adecuados. El 76% valora los seminarios como adecuados o muy adecuados. Los contenidos teóricos han sido considerados en un 67% como adecuados o muy adecuados. En cuanto al grado de dificultad de la asignatura los alumnos la han valorado en un 50% como normal y sólo un 9% la ha valorado como muy difícil. En cuanto al método de trabajo en la asignatura, a los alumnos les ha parecido bien o muy bien en un 65%.

9. Discusión

Uno de los problemas que hemos tenido que abordar es el hecho de acompañar las sesiones de prácticas con las de teoría. Encontramos dos problemas de esta índole. Por un lado, la hora semanal de teoría se imparte los viernes por la tarde. Así que, para todos los grupos, las sesiones de prácticas semanales se imparten antes de la sesión de teoría cuando lo ideal es, en una semana impartir primero la teoría y después realizar las prácticas. Obviamente, la solución ha sido realizar la práctica de una sesión teórica la siguiente semana. Para no desaprovechar las sesiones prácticas de la semana primera, se dedicó ese tiempo a aprender la forma de entrega de los trabajos, así como el formato adecuado para sus memorias, a gestionar su espacio en el campus virtual y, en general, a manejar las herramientas que éste proporciona. De esta manera, el resto de sesiones se dedicarán a la tarea en sí ya que los aspectos de forma de entrega y demás deben estar claros.

Por otro lado, al ser la asignatura muy práctica, la diferencia entre el tiempo dedicado a teoría y el tiempo dedicado a los contenidos prácticos correspondientes es muy dispar. Así acompañando al tema dedicado al software del bloque primero de teoría realizamos las prácticas con Open Office (Writer y Calc). De modo que éstas nos llevan varias semanas mientras que en teoría ya vamos por el bloque segundo permaneciendo desacoplados durante varias semanas.

Consideramos que la evaluación continua de las entregas del alumno a lo largo del cuatrimestre es interesante. Sin embargo, esta modalidad es infactible para las subsiguientes convocatorias. Los alumnos no pueden contar con la tutorización del profesor tanto como en el periodo de docencia de la asignatura. Además los profesores necesitan poder centrarse en las asignaturas del siguiente cuatrimestre o en la investigación. Para incentivar que los alumnos vayan trabajando los entregables propuestos a lo largo del cuatrimestre en el que se imparte la asignatura y así superar, cuanto menos las prácticas y los seminarios en dicho periodo, hemos considerado dos mecanismos. En primer lugar, como se mencionó en la sección 8, hemos establecido que la nota mínima de cada parte para poder calcular la calificación final es menor en febrero (cuatro puntos)

que en el resto de convocatorias (cinco puntos). En segundo lugar, aunque no lo hemos llevado a la práctica este curso, se podría cambiar la política de mantenimiento de aprobados parciales de modo que las partes superadas en febrero se mantengan hasta la convocatoria de septiembre pero en la de junio y septiembre no se guarde nada.

10. Conclusiones y Vías Futuras

En este artículo hemos descrito los aspectos fundamentales de la docencia de la asignatura de IAT I en el grado de Traducción e Interpretación de la Universidad de Murcia. Por los resultados obtenidos tanto en la evaluación del aprendizaje como en las encuestas finales realizadas a los alumnos está bien encaminada hacia la consecución de sus objetivos. Sin embargo, no sabremos a ciencia cierta si hemos logrado preparar a los alumnos para cursar la segunda parte de la materia de Informática mientras que no alcancen el tercer curso. En ese momento recibiremos el *feedback* de los profesores de dicha asignatura y adaptaremos la nuestra con las sugerencias que nos proporcionen.

Por otro lado, estamos seguros de que ha sido cubierto el objetivo de servir como instrumento para ayudar a los alumnos en sus trabajos de otras asignaturas del grado, al menos por los alumnos que han superado la asignatura. Sin embargo, somos conscientes de que obtendremos información feaciente de estos resultados si aplicamos una serie de cambios en la asignatura. Así, consideramos que una manera de asegurarnos de que este objetivo se cubre, es comprobarlo desde dentro de nuestra propia materia. Ya de por sí la materia de informática es instrumental, pero dentro de la propia materia hay contenidos que son más instrumentales que otros. Por ejemplo, las herramientas *groupware* facilitan la colaboración entre los alumnos a la hora de desarrollar cualquier trabajo en equipo propuesto dentro y fuera de nuestra asignatura. Otro contenido especialmente instrumental, obviamente, es el procesador de textos. De forma que, en el próximo curso podemos plantear la docencia de estos dos contenidos mediante actividades de trabajo en grupo en las que se pida la generación de una memoria sobre un tema de investigación. Cada equipo deberá usar las herramientas *groupware* para editar documentos colaborativamente, plantear

reuniones grupales, votar acerca de quién se encarga de cada cosa, etc.. Por otro lado, el documento deberá tener unas características concretas (índices, referencias bibliográficas, etc) que servirán para evaluar si se han adquirido las competencias relacionadas con el procesador de textos.

El mismo tipo de actividad puede extrapolarse a la modalidad de proyectos multidisciplinares en colaboración con otra/s asignatura/s del mismo curso. De hecho, en la asignatura de Lingüística (compartida por ambas titulaciones) se les suelen pedir trabajos en forma de documentos formateados y también de presentación. Sería, por tanto, interesante plantear trabajos conjuntos en los que los responsables de Lingüística establezcan el contenido del trabajo mientras nosotros establezcamos los aspectos estéticos relacionados con lo aprendido al estudiar la herramienta de procesamiento de textos.

El problema de este planteamiento es que el orden de los contenidos tanto teóricos como prácticos tal y como está actualmente no sería adecuado. De todos modos no es descabellado pensar en abordar al inicio de curso las herramientas groupware y el procesador de textos a la luz de los resultados de la encuesta inicial en los que se pone de manifiesto que los alumnos, en su mayoría tienen cierta experiencia, tanto en el uso de herramientas de groupware orientadas al ocio como en la utilización de algún procesador de textos.

Por último y más a largo plazo, nos gustaría que gran parte de los contenidos correspondientes a prácticas y seminarios se trataran completamente en la modalidad de aprendizaje basado en proyectos. Nuestra idea es plantearles al inicio del curso el diseño por equipos de un sitio Web de una supuesta empresa de traducción en diversos idiomas con las características necesarias para ser internacional. El sitio Web podría tener enlaces de distintos tipos como documentos de textos e incluso un repositorio de estilos y plantillas corporativas. También podría tener un enlace a una hoja de cálculo para que cualquier interesado pudiera calcular un presupuesto. Todo esto cubriría el trabajo con la suite ofimática y el diseño web. Por otro lado, este proyecto implica trabajo en grupo, para lo cual les será útil el uso de herramientas groupware.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el 'Programa de Ayuda a los Grupos de Excelencia de la Fundación Séneca 04552/GERM/06'.

Referencias

- [1] Agencia Nacional de Evaluación de la Calidad y Acreditación, *Libro Blanco. Título en Traducción e Interpretación*.
- [2] Virgos Bel, F., Segura Casanovas, J., Marín Sánchez, J., *El proceso de enseñanza/aprendizaje de fundamentos de informática para los grados en ingenierías no informáticas del marco EEES: un planteamiento integrado a partir del modelo de libro OCUPAI*, XV Jornadas de Enseñanza Universitaria de la Informática (JENUi) 2009.
- [3] Orf, D. *Estudio de situación del mercado español de servicios profesionales de traducción*, Agrupación de Centros Especializados, Madrid, 2005.
- [4] Universidad de Murcia. *Título de Grado de Traducción e Interpretación*. <http://www.um.es/letras/estudios.php>
- [5] Netop. *Netop : Secure remote control, classroom management & communication*. <http://www.netop.com/solutions/>
- [6] Prendes, M. P. *SUMA 2.0. Campus Virtual. Guía para el uso de la nueva versión de SUMA Campus Virtual*. Universidad de Murcia. 2009.
- [7] Universidad de Murcia. *SUMA*. <https://suma.um.es/suma/sumav2/>
- [8] Escribano Otero, J. J., García García, M. J., Bemposta Rosende, S. *Campus virtuales: ¿moda pasajera o potenciadores del aprendizaje?*, XVI Jornadas de Enseñanza Universitaria de la Informática (JENUi) 2009.
- [9] Pérez, M., Pérez, R. *Las tutorías electrónicas en el horizonte de la formación virtual: un ejemplo desde la filología inglesa*. RED. Revista de Educación a Distancia, 3(19):11, 2008.
- [10] Prendes Espinosa, M. P. *Guía para la implantación del proyecto de tutoría electrónica en la Universidad de Murcia*. Vicerrectorado de Innovación y Convergencia Europea. Universidad de Murcia.

Sesión 4A:
Calidad y evaluación de la docencia
Formación para la profesión

¿Cómo aprenden los estudiantes de Informática?*

José Manuel Badía, Sergio Barrachina, M. Asunción Castaño, Juan Carlos Fernández

Dpto. de Ingeniería y Ciencia de los Computadores

Universidad Jaume I de Castellón

badia@icc.uji.es, barrachi@icc.uji.es, castano@icc.uji.es, jfernand@icc.uji.es

Resumen

Cada persona tiene un estilo de aprendizaje diferente. En este trabajo se analiza si hay similitudes en la forma de aprender de los estudiantes de nuestra universidad e intentamos deducir en qué medida depende de factores como la titulación o el sexo. Además, comparamos los resultados obtenidos por los estudiantes de Informática de nuestra universidad y los de otras dos universidades, intentando determinar si existe un perfil general que caracterice su estilo de aprendizaje. Finalmente, completamos el análisis realizando una comparación entre los estudiantes de Informática y sus profesores.

Summary

Each person has a different learning style. In this work we analyze whether there are similarities in how the students of our university learn. We are trying to deduce how this style depends on factors such as the qualifications or sex. In addition, we compare the results obtained by the students of Computer Science of our University and two other universities. We try to determine if there is a general learning style profile for these students. Finally, we complete our analysis by comparing the learning style of computer science students and their teachers.

Palabras clave

Estilo de aprendizaje, análisis estadístico.

1. Introducción

No existe una única definición de estilo de aprendizaje, sino que muchos autores dan su

propia definición. Básicamente un estilo de aprendizaje es la forma en que un individuo aprende. Los estudios existentes al respecto ofrecen un marco conceptual que nos permite entender los comportamientos diarios en el aula, cómo se relacionan con la forma en que están aprendiendo los alumnos y los tipos de acciones que pueden resultar más efectivas en un momento dado.

Los primeros estudios realizados sobre estilos de aprendizaje datan de los años 70 y son numerosos. En [3] se presenta un breve estado del arte sobre el tema, mientras que en [11, 12] podemos encontrar una extensísima recopilación de referencias a trabajos realizados desde 1971, con más de 2.500 publicaciones. También existe, al menos que conozcan los autores de este artículo, una revista [13] y un congreso internacional [4] dedicados de forma monográfica a los estilos de aprendizaje.

Hay diferentes taxonomías de estilos de aprendizaje. El punto de partida de varias de estas taxonomías es la propuesta por David Kolb [10]. De ella derivaron, entre otras, las planteadas por Peter Honey y Alan Mumford [9], por Catalina Alonso y Peter Honey [1] o por Richard M. Felder y Linda K. Silverman [5]. Esta última taxonomía es probablemente la más conocida y utilizada en el ámbito de las titulaciones de Ingeniería.

Los estudios realizados sobre estilos de aprendizaje revelan que cada persona tiene un estilo diferente de aprendizaje en función de sus capacidades individuales. Una consecuencia interesante de esta afirmación es que, si el estudiante descubre cómo aprende mejor, puede conseguir un aprendizaje más eficiente. Otra consecuencia derivada del planteamiento anterior es que cada estilo de aprendizaje requiere distintas estrategias didácticas. La tarea del profesor será conocer qué estrategias didácticas son más adecuadas para cada estilo. En este sentido, por ejemplo, Felder y Silverman proponen una serie

* Trabajo desarrollado en el marco del proyecto de innovación educativa 10G136-116, financiado por la UJI.

de actividades y técnicas enfocadas a su clasificación de aprendices [5]. Ahora bien, la existencia de diferentes estilos puede hacer que dentro de la misma aula, una misma actividad sea apreciada por un tipo de aprendices y no por otro, o que una actividad genere satisfacción en un grupo y rechazo en el otro. Por tanto, el diseño de cada asignatura debería combinar actividades enfocadas a todos los estilos de aprendizaje. Si los docentes conociésemos el perfil del estilo de aprendizaje de nuestros alumnos sería posible primar aquellas actividades que estuviesen orientadas a dicho perfil. Todo ello haría más satisfactoria y eficaz la actividad del docente en Informática y permitiría conseguir un aprendizaje más significativo y permanente de *todos* nuestros alumnos.

En los últimos años se han realizado dos estudios, que conozcan los autores del artículo, sobre el perfil general del estilo de aprendizaje que tienen los estudiantes universitarios de Informática en España. El primero fue llevado a cabo en el curso 2006/07 en la Universidad Politécnica de Valencia (UPV) [2]. Y el segundo tuvo lugar en el curso 2008/09 en la Escuela Universitaria Politécnica de Teruel (EUPT) [8]. En ambos casos, para obtener los datos sobre el estilo de aprendizaje de los alumnos se desarrolló una aplicación vía web que proporcionaba una encuesta anónima basada en el cuestionario propuesto por Richard M. Felder y por Barbara A. Soloman [7].

En este artículo presentamos los resultados que hemos obtenido sobre la forma en la que aprenden los estudiantes de la Universitat Jaume I (UJI). Para recoger la información que necesitamos sobre estos estudiantes hemos desarrollado una plataforma web que implementa el cuestionario de estilos de aprendizaje planteado por Felder y Soloman. Con ella cada alumno puede descubrir cómo aprender mejor y los profesores pueden conocer el perfil general del estilo de aprendizaje de sus estudiantes para diseñar estrategias didácticas teniendo en cuenta dicho perfil.

El procesamiento de la información recopilada con la plataforma nos ha permitido estudiar la influencia de diversos factores en el estilo de aprendizaje de nuestros estudiantes. Por un lado comparamos los resultados obtenidos por los estudiantes de Informática de la UJI con los de la UPV y la EUPT, intentando determinar si existe un perfil general de los estudiantes de este ámbito. Por otro lado, comparamos a los estudiantes de

Informática de la UJI con los de Psicología de la misma universidad, con el fin de determinar la influencia del ámbito de la titulación en el modo de aprender. Este último estudio se complementa con un análisis de la influencia del sexo de los estudiantes en su estilo de aprendizaje, aspecto que no estaba contemplado en los estudios referenciados previamente.

Más arriba hemos comentado que un profesor debería combinar en una asignatura estrategias didácticas enfocadas a los diferentes estilos de aprendizaje de sus estudiantes. Sin embargo, un profesor que desconozca esto corre el riesgo de sesgar de forma inconsciente sus actividades hacia estrategias didácticas dirigidas a su propio estilo de aprendizaje, con las que obviamente se encontrará más cómodo. Sería interesante contrastar el perfil general del estilo de aprendizaje de nuestros estudiantes de Informática con el perfil de aprendizaje de sus profesores. En este artículo, a diferencia de los citados previamente, realizamos también este estudio con los profesores de los departamentos de Informática de la UJI.

El resto del artículo se estructura de la siguiente forma. En el Apartado 2 se explica la clasificación de los diferentes estilos de aprendizaje que proponen Felder y Silverman y se proponen algunas actividades orientadas a cada estilo. El Apartado 3 describe las características de la aplicación web que hemos desarrollado para implementar el cuestionario de Felder y Soloman. En el Apartado 4 se analizan los resultados obtenidos comparando los estilos de aprendizaje de distintos grupos de alumnos en función de los factores antes comentados. Finalmente, el Apartado 5 recoge las conclusiones extraídas del estudio.

2. Estilos de aprendizaje según Felder

Richard M. Felder y Linda K. Silverman clasifican los estilos de aprendizaje a partir de cuatro dimensiones: la forma en la que se percibe la información (*sensitivo* o *intuitivo*), el tipo de información que se prefiere (*visual* o *verbal*), la forma de procesar la información que se percibe (*activo* o *reflexivo*) y la manera de adquirir el conocimiento (*secuencial* o *global*). El estilo de aprendizaje de una persona vendrá dado por la combinación de las respuestas obtenidas en las cuatro dimensiones anteriores. A continuación se describen brevemente algunas de las

características de los tipos de aprendices derivados de estas cuatro dimensiones.

A los aprendices *sensitivos* les suelen gustar los hechos, los datos y la experimentación. Son concretos, prácticos y resuelven problemas siguiendo métodos bien establecidos. En el otro extremo de esta misma dimensión están los aprendices *intuitivos*, que prefieren teorías, abstracciones, formulaciones matemáticas y descubrir posibilidades y relaciones. Los aprendices *visuales* prefieren obtener información mediante representaciones visuales y recuerdan mejor lo que ven. Mientras que los aprendices *verbales* sacan más provecho de información escrita o hablada. Los aprendices *activos* tienden a retener y entender la información mejor haciendo algo con ella de manera activa y en grupo. Por el contrario, los aprendices *reflexivos* prefieren pensar solos y con calma sobre la información. Los aprendices *secuenciales* suelen adquirir la comprensión en pasos sucesivos, uno como continuación lógica del otro. Los aprendices *globales* suelen aprender a grandes saltos, absorben material casi al azar, sin ver las conexiones, y de repente “todo les encaja”.

Con el fin de cubrir la forma en la que los diferentes estudiantes de un aula captan y procesan la información, Felder también propone actividades que el profesor puede utilizar [5] enfocadas a cada estilo de aprendizaje concreto. E incluso plantea que un número relativamente pequeño de técnicas en el repertorio del instructor sería suficiente para alcanzar las necesidades de la mayoría de los estudiantes de un aula. Así, por ejemplo, podría relacionar el material que se va a ver con lo que ya se ha visto y con lo que se verá más adelante en el curso y en otras asignaturas (*aprendices globales*), utilizar gráficas, esquemas y diagramas (*visuales*) antes, durante y después de la presentación de material verbal (*verbales*), proporcionar ejemplos concretos del tema que se está tratando (*sensitivos*), dejar tiempo para que piensen sobre lo que se ha dicho o escrito (*reflexivos*), plantear actividades de análisis (*secuenciales*) que podrían ser resueltas de forma colectiva en grupos de 3 o 4 estudiantes en el aula (*activos*).

Para una información más detallada sobre estos estilos de aprendizaje y posibles actividades orientadas a ellos, recomendamos visitar la página web de Richard M. Felder [6]. En ella se recopilan los numerosos artículos (más de cien) que este ha escrito sobre estilos de aprendizaje y su influencia en la enseñanza.

3. Aplicación web de recogida de datos

Richard M. Felder y Barbara A. Soloman diseñaron un cuestionario [7] para conocer las preferencias del aprendizaje de los estudiantes en cada una de las cuatro dimensiones del modelo planteado por Felder y Silverman. Dicho cuestionario consta de 44 preguntas, 11 por cada dimensión. Y cada pregunta tiene 2 posibles respuestas, una orientada hacia un tipo de aprendiz de la dimensión a la que está vinculada y la otra, hacia el otro tipo de aprendiz de dicha dimensión. Cada respuesta tiene asociado un valor de +1 o de -1 dependiendo del tipo de aprendiz que le corresponda. Una vez cumplimentado el cuestionario, se suman los valores acumulados para cada tipo de aprendiz. Por lo tanto, el cuestionario finalmente indica, en una escala del 1 al 11, cuán sensitivo o intuitivo, visual o verbal, activo o reflexivo y secuencial o global es el aprendiz.

En cuanto a la interpretación de los resultados proporcionados por el cuestionario original de Felder, una puntuación entre 1 y 3 en una escala indica que el aprendiz está bastante equilibrado en las dimensiones de dicha escala. Una puntuación entre 5 y 7 revela una preferencia moderada por una de las dimensiones de dicha escala, lo que supone que el aprendiz aprende más fácilmente en un entorno de aprendizaje que favorezca dicha dimensión. Una puntuación entre 9 y 11 en una de las escalas manifiesta que el aprendiz tiene una fuerte preferencia por una de las dimensiones de dicha escala, pudiendo tener verdaderas dificultades en un entorno de aprendizaje que no apoye esa preferencia.

Para recoger las preferencias del aprendizaje de nuestros estudiantes y compañeros profesores según la clasificación de Felder, implementamos una plataforma que permite cumplimentar el anterior cuestionario vía web. La plataforma calcula los resultados alcanzados por el usuario en ese test para cada una de las dimensiones del estilo de aprendizaje y los muestra. De esta forma, el usuario puede ver de forma inmediata cuál es su estilo de aprendizaje. También se le muestra una breve descripción de las características de los cuatro tipos de aprendices y se comentan algunas estrategias dirigidas a mejorar el aprendizaje de cada uno de ellos.

La aplicación permite, por otro lado, mostrar los resultados obtenidos para los estudiantes según la titulación, el grupo de alumnos o el sexo. De esta manera un profesor puede conocer las

características del estilo de aprendizaje de su grupo de alumnos, el perfil del aprendizaje de los estudiantes de una titulación o el perfil de todos los estudiantes del ámbito de la Informática que han cumplimentado el cuestionario.

Por último, cabe indicar que la aplicación es capaz de leer los listados de estudiantes de un curso a partir de las hojas de cálculo que los profesores disponen para sus cursos. De esta forma es muy fácil obtener los estilos de aprendizaje de los estudiantes de un curso determinado.

4. Análisis de la información recogida

El objetivo inicial del estudio era analizar el perfil del estilo de aprendizaje de los estudiantes de Informática de la UJI. Para ello se solicitó la participación de los alumnos que han iniciado este curso, 2010-2011, el nuevo grado en Ingeniería Informática y de los estudiantes de los restantes cursos de las actuales ingenierías en Informática.

El perfil de todos los alumnos de Informática de la UJI se comparó con el realizado previamente para los estudiantes de la UPV [2] y de la EUPT [8].

A continuación nos planteamos si existirían similitudes entre la forma de aprender de estos estudiantes de Informática y la de estudiantes de titulaciones de ámbitos más lejanos a la Ingeniería. Para ello solicitamos a los alumnos del grado de Psicología de la UJI que cumplimentasen también el cuestionario.

La Tabla 1 muestra el número de estudiantes matriculados en las distintas titulaciones y universidades que hemos analizado. Además, se especifica el número y porcentaje de participación de dichos estudiantes en los estudios realizados. A la vista de estos datos podríamos indicar que la muestra sobre la que hemos realizado el estudio parece suficientemente significativa.

Universidad	UJI		UPV	Zaragoza
	Inf.	Psi.		
Matriculados	801	217	3476	118
Participantes	183	55	364	37
%Participación	22,8%	25,5%	10,5%	31,4%

Tabla 1. Tamaño y participación de las poblaciones de estudiantes.

La población estudiantil de Psicología de la UJI está formada en su gran mayoría por mujeres (165 frente a 52), mientras que las mujeres en las titulaciones de Informática son una minoría (117

frente a 684). Interesa pues estudiar si el factor sexo podría estar influyendo en la comparativa entre el estilo de aprendizaje de los estudiantes de Informática y de Psicología. Para ello, hemos realizado un análisis estadístico factorial para intentar aislar el efecto en las respuestas de los dos factores: titulación y sexo.

Finalmente, contrastamos el estilo de aprendizaje de todos los estudiantes de Informática de la UJI con el estilo de aprendizaje de los profesores de Informática de dicha universidad. Cumplimentaron el cuestionario 32 de los 129 profesores que pertenecen a los departamentos de Informática de la UJI.

4.1. Análisis estadístico realizado

A continuación se describen brevemente los tipos de análisis estadísticos realizados con los resultados. Para todos ellos se ha utilizado la herramienta informática *STATGRAPHICS* [14], que permite obtener tanto los resultados numéricos de múltiples tipos de análisis como una breve interpretación de los mismos.

Para analizar estadísticamente los resultados obtenidos en poblaciones diferentes para cada dimensión del aprendizaje hemos realizado tres tipos de contrastes de hipótesis. Por un lado hemos comparado las medias y medianas de las respuestas utilizando respectivamente el *test t de Student* y el *test W* (Wilcoxon). Por otro lado, hemos comparado las distribuciones de frecuencias en las respuestas mediante un *test Chi-cuadrado*. Utilizamos el *p-valor* obtenido en los tres tipos de pruebas como medida para contrastar el cumplimiento de la hipótesis nula en cada caso.

Con el fin de facilitar la interpretación de los resultados presentados en el resto del artículo, recordaremos brevemente el significado del *p-valor* en dichas pruebas. En el caso de la comparación de las medias y medianas, la hipótesis nula a contrastar es la igualdad de dichos parámetros entre las respuestas de las dos poblaciones estudiadas en cada caso. Un *p-valor* muy reducido, p.e. 0,05, permite rechazar dicha hipótesis con un 95% de confianza. Por el contrario, valores más elevados del *p-valor* nos llevan a no rechazar la hipótesis nula y, por tanto, nos permiten afirmar que las medias o las medianas de las respuestas son estadísticamente semejantes.

En el caso del *test Chi-cuadrado*, la hipótesis nula a contrastar es que las dos distribuciones de frecuencias comparadas son independientes. Por ejemplo, si comparamos las respuestas obtenidas

por alumnos de dos titulaciones diferentes, un p-valor pequeño, p.e. 0,1, indica que podemos rechazar dicha hipótesis con un 90% de confianza y, por tanto, que las respuestas dependen de la titulación. Obviamente, valores grandes de este p-valor indican que no es posible rechazar la hipótesis de independencia y, por tanto, que las respuestas se distribuyen de forma estadísticamente semejante.

En aquellos casos en los que queremos estudiar simultáneamente la influencia en una dimensión del aprendizaje de varios factores, como por ejemplo, la titulación y el sexo, se ha utilizado un análisis multifactor de la varianza (conocido como ANOVA, del inglés "ANalysis Of VAriance"). En este caso, se considera como hipótesis nula la igualdad de medias entre las subpoblaciones que definen los distintos valores que pueden tomar los factores a analizar. Por ejemplo, al analizar las respuestas de los estudiantes de Informática y Psicología, este tipo de análisis nos permite descomponer la variabilidad en las respuestas obtenidas en componentes independientes que pueden ser atribuidas a la titulación y al sexo de los estudiantes, e incluso a la interacción entre ambos factores. Los p-valores proporcionados por este análisis se interpretan de la misma forma que se ha comentado para los p-valores del test-t.

4.2. Perfil general de los estudiantes de Informática

En este apartado se compara la forma de aprender de los estudiantes de Informática de la UJI, de la UPV y de la EUPT. En la Figura 1 pueden verse los diagramas de caja y bigotes con los resultados obtenidos en las dos primeras universidades para las cuatro dimensiones del estilo de aprendizaje. Cada caja delimita los valores del segundo y tercer cuartil de las respuestas de las encuestas para una dimensión del aprendizaje. La línea de la caja separa estos dos cuartiles y representa la mediana. El aspa que aparece dentro de la caja muestra el valor medio. Finalmente, los extremos del diagrama, los bigotes, se corresponden con los valores máximo y mínimo obtenidos en las respuestas de la dimensión asociada.

La Figura 1 muestra que tanto los estudiantes de Informática de la UJI como los de la UPV son moderadamente sensitivos y visuales. También se observa que el aprendizaje de los estudiantes en las dos universidades está bastante equilibrado en las dimensiones activo/reflexivo y secuencial/global. Estos comportamientos

coinciden también en las preferencias del estilo de aprendizaje de los alumnos de la EUPT. Aunque estos últimos datos no se han incluido en la Figura por claridad.

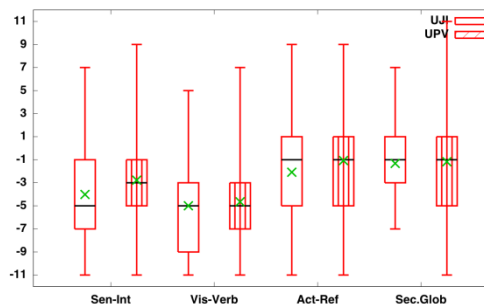


Figura 1. Alumnos de Informática de la UJI y de la UPV.

	Media	Mediana	Distribución
Sen/Int	0,0024 ≠	0,0013 ≠	0,0001 ≠
Vis/Verb	0,6108 =	0,3275 =	0,0003 =
Act/Ref	0,0283 ≠	0,0401 ≠	0,25 =
Sec/Glob	0,8162 =	0,7183 =	0,03 ≠

Tabla 2. Comparación de alumnos de Informática de la UJI, UPV y EUPT.

En la Tabla 2 se muestran los p-valores del test-t, test-W y test Chi-cuadrado del análisis estadístico que compara, respectivamente, la media, mediana y distribución de frecuencias de respuestas de los estudiantes de Informática de la UJI, de la UPV y de la EUPT. En ella se especifica también la interpretación de estos p-valores atendiendo al resultado del contraste de hipótesis; es decir, si el parámetro que se compara (media, mediana o distribución) es o no similar en las tres universidades estudiadas. Según estos datos, en la dimensión sensitivo/intuitivo el estilo depende de la universidad, tanto en la media y mediana de las respuestas como en su distribución de frecuencias. En las dimensiones visual/verbal y secuencial/global, las medias y medianas son similares, pero la distribución de frecuencias de las respuestas depende de la universidad. Finalmente, en la dimensión activo/reflexivo, ocurre lo contrario, las medias y medianas son distintas en las tres universidades pero las respuestas se distribuyen del mismo modo en los tres casos.

Resumiendo, la información recogida y analizada en las tres universidades parece

hacernos intuir que no podemos hablar de un perfil general del estudiante universitario español de Informática, al menos si incluimos las cuatro dimensiones del estilo de aprendizaje en la afirmación.

4.3. Estudiantes de Informática vs. Psicología

A continuación se contrasta el estilo de aprendizaje de los alumnos de Informática de la UJI y de los alumnos del grado de Psicología de la misma universidad. La Figura 2 muestra los diagramas de caja y bigotes para las cuatro dimensiones del aprendizaje de los estudiantes de ambos ámbitos. En ella se observa que los alumnos de los dos ámbitos son bastante sensitivos y visuales, aunque no podemos concretar nada sobre las otras dos dimensiones del aprendizaje.

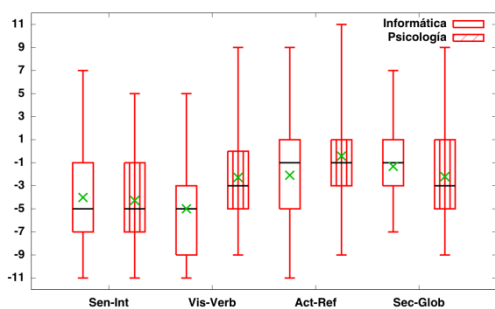


Figura 2. Alumnos de Informática y de Psicología de la UJI.

En la Tabla 3 se muestran los p-valores del test-t, test-W y test Chi-cuadrado. De ella se deduce que los estudiantes de ambos ámbitos son similares en media, mediana y distribución de frecuencias en la dimensión sensitivo/intuitivo. También revela que en la dimensión visual/verbal no coinciden ni las medias, ni las medianas ni las distribuciones. En cuanto a la dimensión activo/reflexivo la distribución de frecuencias de respuestas es similar, pero las medias y medianas son diferentes. A partir de este hecho y observando los diagramas de caja y bigotes de esta dimensión en la Figura 2, se puede concluir que los estudiantes de Psicología tienen un aprendizaje equilibrado en esta dimensión, mientras los estudiantes de Informática presentan un perfil ligeramente más activo. Finalmente, en la dimensión secuencial/global, las medias y medianas son similares en ambos ámbitos, pero la

distribución de frecuencias de sus respuestas es distinta.

	Media	Mediana	Distribución
Sen/Int	0,6945 =	0,9113 =	0,6485 =
Vis/Verb	0,00004 ≠	0,00011 ≠	0,0038 ≠
Act/Ref	0,011 ≠	0,0186 ≠	0,3398 =
Sec/Glob	0,1178 =	0,13 =	0,024 ≠

Tabla 3. Comparación de los alumnos de Informática y Psicología de la UJI.

4.4. Una perspectiva de la influencia del sexo

La población de la titulación de Informática de la UJI es mayoritariamente masculina, mientras que los hombres están en franca minoría en la población de estudiantes de Psicología. En este apartado nos preguntamos hasta qué punto los resultados obtenidos en el apartado anterior están influenciados por este hecho. Lo que nos lleva a plantear si existen o no diferencias estadísticas entre la forma en la que aprenden hombres y mujeres y su relación con la titulación en la que cada uno de estos dos grupos se matriculan. Este tipo de análisis suele resultar polémico. Las conclusiones que extraemos aquí corresponden a las poblaciones y titulaciones estudiadas y distan de ser concluyentes en el ámbito general.

La Tabla 4 muestra los resultados del análisis de varianza realizado con los resultados obtenidos para los estudiantes de las titulaciones de los dos ámbitos. En concreto, muestra los p-valores obtenidos al extraer la influencia de los factores titulación y sexo y de la interacción entre ambos. Por otro lado, las figuras 3 y 4 intentan reflejar gráficamente la influencia separada de cada uno de los dos factores.

	Titulación	Sexo	Interacción
Sen/Int	0,5426 =	0,0788 ≠	0,6495 =
Vis/Verb	0,0049 ≠	0,2355 =	0,8954 =
Act/Ref	0,0065 ≠	0,1870 =	0,3052 =
Sec/Glob	0,5660 =	0,0840 ≠	0,1128 =

Tabla 4. Análisis de varianza de los estudiantes de la UJI.

Los resultados obtenidos indican que los alumnos y alumnas tienen estilos de aprendizaje diferentes en las dimensiones sensitivo/intuitiva y secuencial/global, independientemente de la titulación escogida: Informática o Psicología. Por

el contrario, en las otras dos dimensiones: visual/verbal y activo/reflexivo, el estilo de aprendizaje no depende del sexo, pero sí que es diferente en función de su titulación. Estas conclusiones quedan contrastadas estadísticamente por el análisis de varianza realizado con un 90% de confianza.

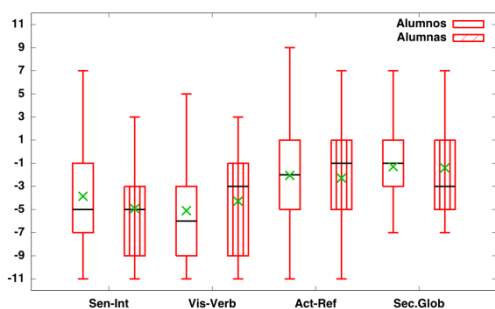


Figura 3. Alumnos y alumnas de Informática de la UJI.

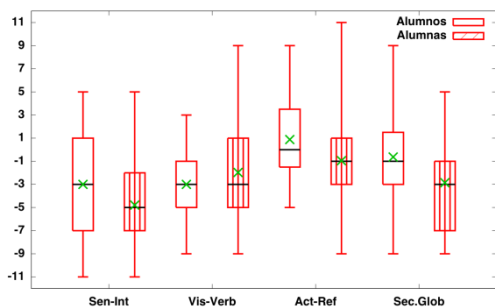


Figura 4. Alumnos y alumnas de Psicología de la UJI.

4.5. Estudiantes vs. Profesores de Informática

La Figura 5 muestra los diagramas de caja y bigotes para el perfil del estilo de aprendizaje de los estudiantes y profesores de Informática de la UJI. La figura refleja que tanto alumnos como profesores son sensitivos y visuales, pero no nos permite alcanzar ninguna conclusión de este tipo sobre las otras dos dimensiones.

En este caso los p-valores relativos a las comparaciones de medias y medianas, mostrados en la Tabla 5, se encuentran en la mayoría de los casos próximos a 0,1. Consecuentemente, las diferencias de las respuestas de alumnos y profesores solo pueden detectarse claramente al bajar el nivel de confianza del 95% al 90% en el análisis estadístico de estos parámetros. La tabla recoge la interpretación del resultado del contraste

de hipótesis en base a este último nivel de confianza.

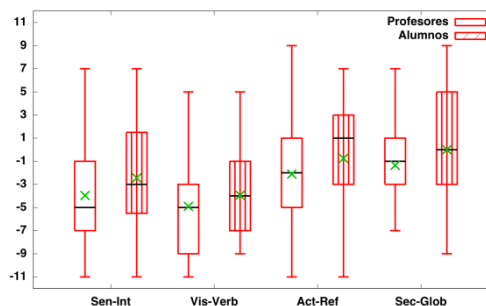


Figura 5. Alumnos y profesores de Informática de la UJI.

	Media	Mediana	Distribución
Sen/Int	0,0877 ≠	0,094 ≠	0,2247 =
Vis/Verb	0,2214 =	0,1724 =	0,8052 =
Act/Ref	0,107 ≠	0,0925 ≠	0,212 =
Sec/Glob	0,0586 ≠	0,0908 ≠	0,0013 ≠

Tabla 5. Comparación de alumnos y profesores de Informática de la UJI.

De dichos resultados se deduce que en la dimensión visual/verbal profesores y alumnos tienen similares medias, medianas y distribuciones. Por el contrario, los tres parámetros indican que su estilo de aprendizaje es diferente en la dimensión secuencial/global. En las dimensiones sensitivo/intuitivo y activo/reflexivo las medias y medianas son distintas, mientras la distribución de frecuencias de las respuestas es parecida. Teniendo esto en cuenta y observando los diagramas de caja y bigotes de estas dimensiones en la Figura 5, se puede concluir que los profesores tienen un aprendizaje ligeramente más sensitivo y activo que los alumnos.

5. Conclusiones

En este artículo presentamos una plataforma web que permite, por una parte, que cada alumno descubra su estilo de aprendizaje atendiendo a la clasificación que Felder y Silverman proponen para estos [5]. Y, por otra, que los profesores sean conscientes del perfil general del estilo de aprendizaje de sus alumnos. En base a dicho perfil el profesor puede planificar actividades, como las formuladas en [5], que permitan conseguir un

aprendizaje más significativo y permanente de sus alumnos.

A partir de la plataforma desarrollada, presentamos un estudio que se centra en una muestra limitada, aunque creemos que significativa, de alumnos de tres universidades.

De dicho estudio podemos concluir, en primer lugar, que no parece existir un perfil general de los estudiantes universitarios de Informática, aunque sí que presentan un perfil moderadamente sensitivo y visual.

Por otro lado, el análisis estadístico nos muestra que los alumnos de Informática y Psicología solo son similares en media, mediana y distribución de sus respuestas en la dimensión sensitivo/intuitivo. No obstante, los estudiantes de Psicología también muestran un perfil sensitivo y visual.

Al separar estadísticamente la influencia de la titulación y el sexo en el perfil de los estudiantes anteriores, hemos podido contrastar que alumnos y alumnas tienen un perfil diferente en las dimensiones sensitivo/intuitivo y secuencial/global. Mientras los otros dos perfiles parecen depender de la titulación y no del sexo.

Por último, hemos podido comprobar que los profesores de Informática solo coinciden con sus alumnos en las tres pruebas estadísticas en la dimensión visual/verbal, ya que ambos son moderadamente visuales y sus respuestas se distribuyen de forma similar.

Agradecimientos

Queremos expresar nuestro agradecimiento a los estudiantes y profesores de la UJI que han participado en este estudio. A Juan Carlos Cano y Francisco J. Martínez, por compartir los datos de los estudios que realizaron en sus respectivas universidades. Y a Amelia Simó, por su ayuda en el desarrollo e interpretación de resultados del análisis estadístico.

Referencias

- [1] Alonso, C., Gallego, D. y Honey, P. *Los estilos de aprendizaje: procedimientos de diagnóstico y mejora*. Mensajero, 1994.
- [2] Calafate, C.T., Cano, J.C., Manzoni, P. “El estilo de aprendizaje como característica distintiva entre alumnos: un enfoque estadístico”. Actas de las *XIII Jornadas de Enseñanza Universitaria de la Informática (JENUI-2007)*, p. 529—536. 2007.
- [3] Castaño, M.A., Marqués, M., Satorre, R., Jaume i Capó, A., López, D. “Tengo una respuesta para usted sobre estilos de aprendizaje, creencias y cambios en los estudiantes”. Actas de las *XVI Jornadas de Enseñanza Universitaria de la Informática (JENUI-2010)*, p. 275—282. 2010.
- [4] *III Congreso Internacional de Estilos de Aprendizaje*. Cáceres 2008. <http://www.caceres2008.es/informacion.htm>.
- [5] Felder, R.M., Silverman, L.K. “Learning and Teaching Styles in Engineering Education,”. *Engr. Education*, 78(7), p. 674—681. 1988.
- [6] Felder, R.M. *Página web de Felder*. <http://www4.ncsu.edu/unity/lockers/users/f/felder/public/>. Última visita: febrero 2011.
- [7] Felder, R.M., Silverman, L.K., Soloman, B.A. Index of Learning Styles (ISL). <http://www4.ncsu.edu/unity/lockers/users/f/felder/public/ILSpage.html>. Última visita: febrero 2011.
- [8] Garrido, P., Martínez, F.J., Calafate, C.T., Cano, J.C., Manzoni, P. “Adaptación de los métodos de enseñanza a los métodos de aprendizaje de los alumnos”. Actas de las *XVI Jornadas de Enseñanza Universitaria de la Informática (JENUI-2010)*, p. 267—274. 2010.
- [9] Honey, P. y Munford, A. *The manual of learning styles*. Honey, P., 1992.
- [10] Kolb, D.A. *Experiential Learning: Experience as the Source of Learning and Development*. Prentice Hall, 1984.
- [11] Kolb, A. y Kolb D. *Experiential Learning Theory Bibliography. Volume 1: 1971-2005*. http://www.learningfromexperience.com/images/uploads/Bibliography_1971-2005.pdf. Última visita: febrero 2011.
- [12] Kolb, A. y Kolb D. *Experiential Learning Theory Bibliography. Volume 2: 2006-2009*. http://www.learningfromexperience.com/images/uploads/Bibliography_2006-2009.pdf. Última visita: febrero 2011.
- [13] *Revista de Estilos de Aprendizaje*. <http://www.uned.es/revistaestilosdeaprendizaje/>. Última visita: febrero 2011.
- [14] *STATGRAPHICS*. <http://www.statgraphics.net>. Última visita: febrero 2011.

Extensión de Moodle para la gestión colaborativa de proyectos

Lapuente, R. ESSI ¹ UPC ² r.lapuente@gmail.com	Piguillem, J. ESSI ¹ UPC ² j.piguillem@gmail.com	Mayol, E. ESSI ¹ UPC ² mayol@essi.upc.edu	Alier, M. ESSI ¹ UPC ² malier@gmail.com	López, J. ESSI ¹ UPC ² cokezone@gmail.com
---	---	--	--	--

Jordi Girona 1-3, 08034 Barcelona

¹Department of Engineering of Services and Information Systems

²Universitat Politècnica de Catalunya (BarcelonaTech)

Resumen

Con el objetivo de dar soporte a las asignaturas que conllevan la realización de prácticas o pequeños proyectos y a los Proyectos de Fin de Carrera, se ha llevado a cabo un proyecto de innovación docente que integra la plataforma de soporte a la docencia Moodle con Trac, un gestor de proyectos potente, sencillo y flexible.

Los estudiantes esta habituados al uso de Moodle para realizar actividades colaborativas en algunas asignaturas, pero no tienen un soporte específico para la planificación, asignación y seguimiento de tareas.

Con esta integración, se enmarcan los proyectos gestionados por Trac dentro del contexto de un curso de Moodle.

En este artículo se describe dicha integración y la aplicación del sistema combinado Moodle-Trac en la docencia en este tipo de asignaturas.

1. Introducción

Una de las premisas en las que se basa el nuevo Espacio Europeo de Educación [1] es la consideración del aprendizaje no sólo centrado en conocer unos conceptos teóricos, sino en un aprendizaje más práctico en que el estudiante asimila y ejercita unas técnicas y prácticas que le han de servir en su futuro profesional.

Para ello, la unidad de medida (créditos ECTS) utilizada en este marco refleja los resultados del aprendizaje y volumen de trabajo realizado por el estudiante para alcanzar los objetivos establecidos en cada asignatura.

Tal y como se refleja en el Marco Europeo de Calificaciones para el Aprendizaje Permanente (EQF-MEC) [2], los resultados del proceso de

aprendizaje abarcan los conocimientos teóricos, las destrezas prácticas y técnicas, y las competencias sociales, entre las cuales destaca el trabajo en equipo.

Este nuevo marco ha conducido a definir nuevas asignaturas en las que el trabajo en equipo es una de las bases fundamentales para el aprendizaje del estudiante. Prueba de ello son la implantación y adopción de nuevas metodologías de aprendizaje como son el Aprendizaje Colaborativo y el Aprendizaje Basado en Proyectos (PBL) [3], así como la utilización de herramientas informáticas de soporte a la formación y aprendizaje como pueden ser los Sistemas de Gestión del Aprendizaje o Learning Management System (LMS) [4].

Un ejemplo de estos sistemas más utilizado y difundido en el ámbito universitario es sin duda Moodle [5]. Moodle es un LMS orientado a la construcción de espacios on-line en los que el profesor y los estudiantes colaboran en el proceso de aprendizaje y en el que comparten conocimientos, información y documentación.

En cambio, Moodle no dispone de herramientas específicas que faciliten la coordinación, planificación y seguimiento de tareas de un grupo de estudiantes que realizan un trabajo o proyecto de forma colaborativa.

En este sentido, se ha propuesto una extensión del sistema Moodle con una herramienta de gestión de tareas, como Trac [6]. Trac es un sistema que facilita el seguimiento de proyectos de desarrollo de software que intenta ser muy flexible y no determinar ni imponer el proceso o metodología a seguir por el equipo que realiza el proyecto.

Así pues, este artículo es la descripción del sistema desarrollado y la experiencia de su

aplicación en asignaturas en las que los alumnos realizan actividades colaborativas en pequeños grupos que requieren de una cierta organización y coordinación de tareas, como la realización de una práctica o pequeño proyecto usando la plataforma Moodle.

2. Asignaturas EEES

Las asignaturas en el nuevo EEES se basan, entre otras, en el aprendizaje colaborativo y el aprendizaje basado en proyectos. Así pues, la herramienta propuesta debe dar soporte a estas tipología de asignaturas y a los estudiantes cuando requieran de colaboración y comunicación constante entre ellos para realizar el proyecto de forma presencial o semi-presencial.

Por otra parte, la herramienta también debe aportar una mejora considerable de comunicación entre el profesor y el estudiante en asignaturas en que este último realiza un trabajo dirigido por el profesor.

Finalmente, la herramienta es también adecuada para revisar y tener un mejor conocimiento del trabajo realizado por los estudiantes, cuando éstos deben ser evaluados por la intensidad de trabajo realizado.

Así pues, las asignaturas que se escogieron para aplicar y probar el sistema se pueden agrupar en tres grupos, según las características de trabajo de las mismas:

- Asignaturas en que los estudiantes realizan un trabajo de forma individual bajo la dirección o tutela de un profesor. Como ejemplo, las asignaturas de Proyectos Fin de Carrera (PFC), Proyectos Final de Grado (PFG), Trabajo Final de Máster (TFM).
- Asignaturas donde los estudiantes realizan, en pequeños grupos, un trabajo o proyecto de forma conjunta siguiendo una metodología definida, pero de forma autónoma y bajo una supervisión indirecta del profesor. En este grupo, hemos escogido la asignatura de Proyecto de Ingeniería del Software y Bases de Datos (PESBD).
- Asignaturas más centradas en la propia gestión de proyectos desde un punto de vista metodológico. Por ejemplo, las asignaturas de Gestión de Sistemas de Información (GSI) y Planificación y Gestión de Proyectos y Sistemas Informáticos (PGPSI).

Todas y cada una de estas asignaturas pueden tener definido un proceso de ejecución o flujo de trabajo. Pero en la mayoría de casos, no coinciden en cual. Por lo tanto, es de especial relevancia el usar una herramienta de gestión flexible y adaptable como Trac.

En muchos casos, al tratarse de la realización de proyectos software, estas metodologías pueden basarse en estándares como Unified Process [7], SWEBOK [8], Scrum [9], etc. Pero en realidad, cuando el proyecto es más pequeño, los propios estudiantes relajan i flexibilizan las etapas y fases definidas en estas metodologías adaptándolas a sus necesidades y a las particularidades del proyecto y a la forma como ellos se organizan el trabajo.

De hecho, en una encuesta realizada a un conjunto de profesores que dirigían proyectos finales de carrera en la FIB, reconocieron que usaban metodologías diversas para el seguimiento de sus proyectistas y entregables asociados. En cambio, para la coordinación de tareas más pequeñas les era especialmente necesaria un alto nivel de flexibilidad [10].

3. Atenea/Moodle

La plataforma Atenea es una herramienta basada en Moodle integrada en el campus virtual de la UPC y que está a disposición de alumnos y profesores para ayudarles en la gestión diaria de las asignaturas.

Moodle, software libre de Sistemas de Gestión del Aprendizaje o Learning Management System (LMS). Actualmente Moodle es el LMS más utilizado en España en universidades (donde la mayoría de instituciones lo implementan), escuelas, institutos, formación ocupacional y formación continuada.

Moodle implementa un aula virtual en la que profesor y alumnos pueden compartir recursos y participar en distintas actividades colaborativas online, donde se pueden realizar cursos online o complementar la actividad de los cursos presenciales.

El profesor dispone de una serie de herramientas para poder evaluar el trabajo de sus estudiantes y gestionar los cursos.

En cambio, no dispone de funcionalidades para la coordinación de un equipo de trabajo o para la organización y seguimiento de las tareas de un proyecto.

El diseño interno de Moodle permite que se puedan desarrollar módulos o plugins para ampliar su funcionalidad, de forma más o menos sencilla, evitando modificar el código original.

Pero a la vez, la comunidad Moodle dispone de un estilo o normas de programación que deben seguirse para mantener la calidad del producto.

4. Trac

Trac, software libre, que implementa un sistema de gestión de proyectos online. Originalmente Trac era un sistema orientado a ser usado en consola o terminal UNIX, pero hace años se ha incorporado una interfaz web, que es la forma más común en que se usa. Trac es un proyecto software libre con muchos desarrolladores, muy utilizado en proyectos en todo tipo de sectores y para el que hay una gran librería de extensiones disponible.

Trac fue diseñado en su origen para ser una herramienta de gestión y tracker para proyectos de software y por ello posee una excelente integración con la mayoría de gestores de versiones y de código (svn, git, mercurial, bazaar u otros).

Trac ofrece una serie de opciones y funcionalidades a sus usuarios:

- Gestión de proyectos: Roadmap, Timeline, Milestones...
- Sistemas de Ticket: tareas, seguimiento de bugs ...
- Granularidad de permisos.
- Wiki.
- Soporte para múltiples proyectos.
- Comunicación vía email.
- Un gran número de plugins de los usuarios para: ofrecer Web Services, creación de archivos Pdf o muchos otros.

Además, Trac es altamente configurable y adaptable con lo que le convierte en el candidato

perfecto para integrarse con Atenea y, de este modo, complementar sus características. Trac es una excelente herramienta de trabajo en grupo (groupware) para gestionar proyectos.

5. Prototipo Moodle-Trac

El prototipo desarrollado ha sido un sistema innovador desde un punto de vista docente y tecnológico, fruto de la integración de dos plataformas basadas en software libre.

Hemos integrado una plataforma de trabajo extensamente usada en ámbitos docentes de la UPC, como es Moodle, con un gestor de tareas y proyectos. Concretamente, se ha integrado la implementación de Moodle disponible en la UPC, denominada Atenea, con el gestor de proyectos Trac.

Con esta integración, se han extendido las funcionalidades ofrecidas a los usuarios de Atenea para la organización, coordinación y colaboración de equipos de trabajo en proyectos entre los estudiantes que lo realizan conjuntamente y de forma autónoma, como para equipos dirigidos por un profesor.

Un requisito especialmente relevante en el sistema ha sido el aprovechar y adaptar el mecanismo de autenticación de usuarios de la plataforma, a los mecanismos y protocolos establecidos en la propia universidad. De esta forma, se han evitado las múltiples veces en que un estudiante o profesor deben identificarse para acceder a una u otra aplicación. De forma que con una única autenticación inicial (al acceder a Atenea) se podrá acceder a todas y cada una de las funcionalidades aportadas por Atenea i Trac.

Desde un punto de vista más técnico, el sistema se ha diseñado estableciendo un canal de comunicación entre las dos aplicaciones de forma que puedan interoperar.

Mediante el uso de Web Services (protocolo

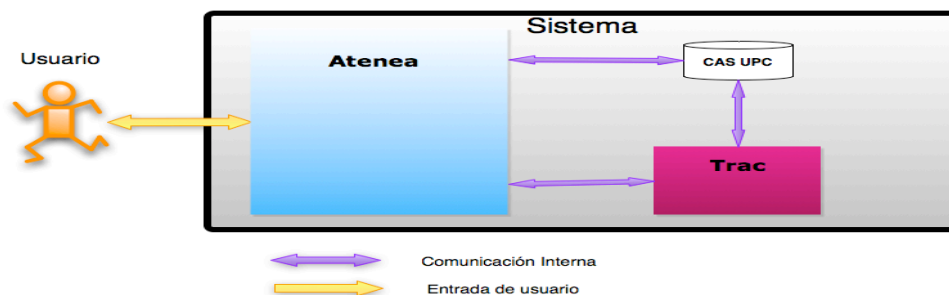


Figura 1. Mecanismo de comunicación entre Atenea y Trac

XML-RPC) y a~nadiendo m3dulos de comunicaci3n en las dos aplicaciones, se ha conseguido una interacci3n remota, transparente y autom3tica. Por tanto, las dos aplicaciones se comunican de forma totalmente transparente al usuario y obteniendo informaci3n contextual necesaria sin que 3ste sea consciente de ello.

La extensi3n realizada en Moodle (Atenea) ha consistido en a~nadir un m3dulo de actividad (en terminolog3a Moodle) o plugin, el cual, se instala como un complemento independientemente al sistema Moodle elegido. Una vez instalado y configurado con los datos del servidor donde reside el gestor de proyectos de Trac, ya est3 listo para entrar en funcionamiento.

Se ha dise~nado de la forma m3s simple posible para facilitar su instalaci3n y siguiendo las directrices y estilo de desarrollo definida por la comunidad Moodle.

De forma similar, en el sistema Trac se ha incorporado otro plugin que define el conjunto de Web Services que se ofrecen a Moodle (o cualquier otra plataforma) para interactuar con Trac remotamente. Estos Web Services permiten que desde Moodle se pueda consultar y actualizar informaci3n propia de Trac.

Adem3s, se ha conseguido que la administraci3n de Trac se haga de forma autom3tica a trav3s de Web Services, y en definitiva, remotamente desde la plataforma Moodle. Aspecto in3dito hasta el d3a de hoy. Hasta ahora, era necesaria la figura de un administrador para realizar ciertas tareas directamente desde la consola de administraci3n de Trac, mientras que con nuestra aportaci3n, esta tarea se puede hacer autom3ticamente, con la

informaci3n que se encuentra en Moodle, gracias a los Web Services desarrollados.

Las tareas de validaci3n y autenticaci3n de usuarios, as3 como toda su gesti3n, la dejamos en manos del sistema de autenticaci3n propio de la UPC denominado CAS. Se han instalado y adaptado los correspondientes m3dulos de seguridad de Moodle y de Trac para aprovechar este servicio.

Por otro lado, la comunicaci3n Moodle-Trac se realiza a trav3s de los Web Services, lo que conlleva la necesidad de realizar otro control de la validaci3n y autenticaci3n de estos Web Services. Este aspecto, se ha tenido que desarrollar expl3citamente para nuestro sistema.

6. Uso del sistema

Del conjunto de asignaturas que se mencionan en un apartado anterior como potenciales candidatas a beneficiarse del uso del sistema, finalmente se escogieron dos representantes como asignaturas piloto para realizar la evaluaci3n del sistema Moodle-Trac. Estas asignaturas piloto presentan dos escenarios distintos de utilizaci3n del sistema, presentando particularidades y problem3ticas distintas.

Por un lado, se escogi3 la asignatura de Gesti3n de Sistemas de Informaci3n (GSI) y por otro la realizaci3n del Proyecto Final de Carrera (PFC).

Gesti3n de Sistemas de Informaci3n (GSI)

La motivaci3n principal para escoger esta asignatura ha sido el dar soporte a las actividades de laboratorio de la misma. En las sesiones de laboratorio, se realiza la gesti3n del desarrollo y la auditoria de un proyecto inform3tico.

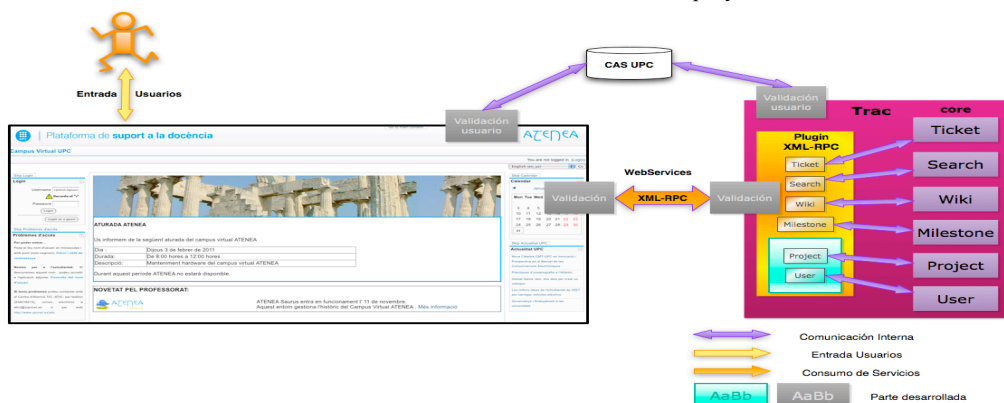


Figura 2. Arquitectura detallada del sistema

Esta gestión se realiza entre equipos de estudiantes asumiendo distintos roles. Los alumnos se organiza en equipos de 3-4 personas. En cada equipo se designa un jefe o responsable del proyecto. Cada equipo realiza dos actividades: por un lado, la propia gestión del proyecto, y por otro, se evalúa o audita la gestión de un proyecto gestionado por otro grupo de estudiantes. Así, cada equipo ejecuta a la vez el rol de gestor i el de auditor de proyectos.

Finalmente, se elabora una memoria del proyecto incluyendo el informe de definición, un informe de calificaciones y una planificación del mismo. A la vez que se presenta la memoria resultante de la auditoría del otro proyecto.

En esta experiencia, el sistema Moodle-Trac se ha usado para facilitar la coordinación entre los miembros del equipo en su tarea de gestionar el proyecto y para facilitar la comunicación y coordinación entre el equipo auditor y el equipo gestor. El uso de Moodle como repositorio e intercambio de documentación, así como mecanismo para la evaluación por parte del profesor de las tareas realizadas por los estudiantes no es novedad en el sistema, pues son funcionalidades propias de Moodle.

La prueba del sistema Moodle-Trac en esta asignatura ha consistido en validar la utilidad del mismo en el desarrollo de los siguientes aspectos:

- *La coordinación interna de los diferentes miembros de cada equipo.* Por un lado, la coordinación entre los miembros del equipo y su jefe de proyecto, para realizar la gestión de su proyecto, y por otro, para coordinar la actividad de los auditores entre sí.
- *La comunicación entre equipos.* Se valida que el sistema es realmente útil y facilita la comunicación entre el equipo gestor del proyecto y el equipo auditor.
- *La compartición de documentación e informes entre equipos.*

En esta experiencia han participado el profesor de la asignatura y un grupo de 40 estudiantes. En realidad, no todos los alumnos han usado suficientemente la herramienta para valorar su utilidad. De hecho, han sido un total de 6 equipos que la han usado tanto desde el punto de vista de equipo gestor como de equipo auditor.

Proyectos Finales de Carrera (PFC)

En realidad, no se trata de una asignatura como tal. La realización del PFC consiste en la realización de un trabajo/proyecto de cierta

envergadura por parte de cada estudiante en su último curso de carrera, integrando todos los conocimientos y técnicas aprendidas durante la misma.

En este caso, se ha elegido la realización del PFC para validar y evaluar la utilidad del sistema Moodle-Trac en la mejora de la comunicación y coordinación entre el profesor y el alumno.

La realización del PFC consiste en que un estudiante (Proyectista), de forma individual, lleva a cabo un proyecto en su totalidad bajo la dirección o tutela de un profesor (Director). La comunicación y coordinación entre director y proyectista consiste en un encargo de tareas y la supervisión, por parte del director, de los resultados obtenidos. Al mismo tiempo, que el Proyectista debe realizarlas pidiendo el apoyo necesario y organizándose el trabajo de forma adecuada.

La propia normativa de PFC en la Facultad de Informática de Barcelona (FIB) establece un flujo o proceso de trabajo a seguir por ambos participantes definiendo unas etapas a seguir, con un conjunto de documentos a entregar y unas temporalidades a respetar. En cambio, no define la metodología a seguir para el desarrollo del proyecto, pues definirla y seguirla es responsabilidad del director y del proyectista.

Con esta experiencia, aplicada en algunos proyectos finales de carrera, se ha pretendido evaluar si:

- *Organización del trabajo.* Si el sistema ha sido de ayuda para el proyectista para organizarse el trabajo y las tareas a realizar.
- *Coordinación.* Si el sistema ha facilitado la tarea de coordinar o dirigir las tareas que el profesor encarga al proyectista, así como las tareas o peticiones que el proyectista solicita al director.
- *Supervisión.* Si el sistema ha facilitado la tarea de supervisión del proyectista por parte del director en referencia a la calidad del trabajo realizado y el tiempo o etapas en que lo ha realizado.

El sistema ha sido usado y evaluado en dos proyectos finales de carrera por dos profesores y alumnos distintos.

7. Evaluación de la experiencia e impacto

La evaluación de la experiencia realizada que se presenta en este apartado, se ha obtenido a partir

de las opiniones expresadas por todas las personas implicados en el desarrollo y uso del sistema.

Por un lado, a los estudiantes se les hizo una encuesta para que valorasen el uso de la herramienta para organizarse y coordinarse con el resto de compañeros. Por otro, a los profesores se les solicitó un informe evaluando la utilidad del sistema y cómo, desde su punto de vista, los alumnos habían utilizado el sistema. Finalmente, se aporta la valoración del propio equipo desarrollador del sistema y el interés institucional del mismo.

Gestión de Sistemas de Información (GSI)

Encuesta a los estudiantes

Los estudiantes encuestados estaban habituados a usar Moodle en otras asignaturas, en cambio no tenían experiencia en el uso de Trac, ni en el uso de otras herramientas de gestión de proyectos.

Ellos mismos reconocieron que para realizar trabajos en grupo se organizaban con el uso de otra herramientas como el correo electrónico, calendarios y/o agendas para planificarse y marcarse la temporalidad de las diferentes tareas a realizar.

De todos modos su valoración global fue positiva pues les facilitaba la organización, coordinación y control del proceso de trabajo a realizar.

Como herramienta de gestión y compartición de información y documentación, el comentario más generalizado ha sido que ya utilizaban mecanismos externos para compartir documentos en la red, pero que para compartir información con el profesor, el uso del sistema integrado les ha sido especialmente útil. Por otro lado, han manifestado una cierta dificultad en el uso del subsistema Trac, pues no lo encontraban del todo intuitivo.

La mayoría de ellos consideraban que la extensión de Moodle/Atenea con el sistema de gestión Trac completamente integrado era una iniciativa que les facilitaba el trabajo y coordinación entre ellos.

Valoración del profesor de GSI

Atenea ha funcionado perfectamente como intranet oficial de la asignatura a pesar de que algunos alumnos preferían otra plataforma docente propia de la facultad.

Como mecanismo de comunicación interna dentro del equipo, el nuevo sistema ha sido utilizado mínimamente, pues los estudiantes

estaban habituados al uso de otros mecanismos de comunicación (correo electrónico, por ejemplo) que han seguido utilizando. En cambio, donde los responsables del proyecto han manifestado mayor utilidad del sistema ha sido en la comunicación oficial entre el equipo desarrollador y el equipo auditor, y con el profesor.

El subsistema Trac ha tenido una buena acogida, a pesar de que el sistema no se empezó a utilizar desde el principio de curso y que las sesiones de formación que recibieron los estudiantes fueron muy reducidas.

Un aspecto a revisar para futuras versiones, sería el resolver la confusión terminológica entre la del sistema Trac y la del estándar PMBOK [11] (milestones, tickets, etc.). Además, la principal dificultad que se han encontrado los estudiantes en el uso efectivo del sistema ha sido que el nivel de granularidad de las tareas o actividades de Trac es más fina que las representadas en un diagrama Gantt, generando una cierta confusión a la hora de hacer el seguimiento real del proyecto respecto a la planificación.

En realidad, esta confusión se ha generado debido a la extrema flexibilidad de configuración de tareas de Trac. Lo que ha provocando una dificultad en su uso, prefiriendo utilizar otros mecanismos que los alumnos están más habituados a usar, aunque no sean los más adecuados.

La valoración general del profesor ha sido positiva, pero haciendo notar sería necesario un proceso de formación inicial más profundo en las posibilidades de la herramienta, tanto desde el punto de vista de comunicación y compartición de información (Atenea), como de organización y seguimiento del proyecto, así como de la flexibilidad que permite el propio Trac.

Dirección de Proyectos Finales de Carrera (PFC)

Los proyectos finales de carrera tienen definido un workflow administrativo por la propia facultad, pero el proceso de realización y organización del mismo está en manos del propio director y proyectista.

El entorno combinado Moodle-Trac y el uso de un repositorio de código (CVS, GIT o SVN) proporciona un apoyo a las actividades de planificación y seguimiento, así como la gestión de los documentos de trabajo generado en las reuniones entre proyectista y director, las entregas, el código y su documentación.

En los proyectos en los que se ha aplicado esta herramienta se ha observado como se registran los datos del proceso de ejecución del proyecto y se hace un seguimiento más cuidadoso del mismo.

Los datos almacenados en Moodle y en Trac son muy útiles al proyectista de cara a elaborar su memoria final del proyecto, pues el seguimiento y la valoración de los esfuerzos realizados son menos sesgadas que en la mayoría de los proyectos en que no se dispone del detalle de estos datos.

El hecho de utilizar herramientas que también se utilizan en el mundo real para la gestión de proyectos (trackers, repositorios, wikis, etc) aporta un valor adicional al valor formativo del Proyecto Final de Carrera.

Valoración del equipo desarrollador

Con la ampliación de la plataforma Moodle con un sistema de gestión de tareas y proyectos, creemos que hemos conseguido una herramienta que permite que los equipos de trabajo puedan comunicarse, estructurar la realización de un proyecto y hacer el seguimiento de una forma más sencilla, rápida, sostenible y con un ahorro considerable de tiempo.

En particular, el sistema integrado permite a los profesores estar en contacto y coordinarse con los alumnos de una forma semi-presencial más efectiva y cómoda que hasta ahora y, en definitiva, hacer el seguimiento del estudiante de forma más continuada. Aspecto muy útil y necesario con la nueva incorporación del Plan Bolonia en la UPC.

Los usuarios han sido capaces de crear a voluntad espacios de trabajo donde se han organizado el trabajo a realizar, se han comunicado de forma más eficiente y ordenada, y donde han compartido información de forma más actualizada y segura.

La facilidad de uso de las plataformas Moodle y Trac se ha mantenido y ha permitido que los usuarios diseñen y personalicen su propio conjunto de tareas, avisos y alertas por cada proyecto en que participan. En definitiva han podido definir y seguir la metodología de trabajo que mejor se adaptaba a sus necesidades de forma sencilla y adaptable. Aunque reconocemos, que el proceso de formación en la asignatura de GSI ha sido poco efectivo y posiblemente requiere un replanteamiento por nuestra parte para futuras ediciones en ésta y otras asignaturas.

En cambio, como soporte a la realización de proyectos final de carrera, la experiencia ha sido más satisfactoria.

De forma complementaria, el equipo desarrollador está especialmente satisfecho por haber logrado superar la dificultad tecnológica de integración de los dos sistemas. Ambos están implementados con tecnologías diferentes. Por un lado Moodle está programado en PHP mientras que el Trac este programado en Python.

Estos lenguajes tienen características y diseños completamente diferentes y la integración no debía comportar la reprogramación de ninguno de los dos sistemas a integrar. Además, la integración debía ser completamente transparente y el máximo de adaptable posible, lo cual nos llevó a usar tecnologías orientadas a la integración como son los Web Services.

Interés institucional

Desde un punto de vista institucional, el proyecto ha recibido diferentes muestras de interés dentro del ámbito de la propia UPC. Por una parte, la propuesta de integración Moodle-Trac se presentó a la 6ª convocatoria del Premio Davyd Luque a la Innovación en las TIC organizado por UPCNet y la UPC este año 2010, obteniendo el Primer Premio. Lo cual supone un reconocimiento del carácter innovador del mismo y una valoración explícita a su alto impacto potencial. Tanto dentro de la comunidad UPC, al basarse en la plataforma Atenea ampliamente utilizada en la universidad, como en entornos que utilizan regularmente la plataforma Moodle.

Por otro lado, desde la FIB, el interés fue evidente desde el primer momento. De hecho, la participación de la FIB ha sido determinante en el éxito de la propuesta. Ha aportado el servidor donde implantar el sistema, ha mostrado especial interés en desarrollar una versión del sistema a integrar en su propia plataforma docente (Racó de la FIB) y ha facilitado la elaboración de una encuesta a profesores directores de proyectos finales de carrera en las fases iniciales del proyecto, así como la encuesta realizada a los usuarios finales del sistema.

8. Conclusiones

Tal y como se refleja en Marco Europeo de Calificaciones para el Aprendizaje Permanente (EQF-MEC), los resultados de un proceso de

aprendizaje abarcan un amplio abanico de conocimientos teóricos, destrezas prácticas y técnicas, y competencias sociales, entre las cuales destaca el trabajo en equipo.

Con la finalidad de favorecer este trabajo en equipo, y más concretamente, la coordinación y la colaboración dentro de los equipos, se ha llevado a cabo este proyecto.

En este proyecto se ha integrado, dentro de una plataforma de aprendizaje ampliamente utilizada y popular como es Moodle, un sistema gestor de tareas o proyectos. La implementación de Moodle escogida ha sido la existente en la UPC, denominada Atenea, que no disponía de herramientas que facilitaran esta coordinación. Por lo tanto se extendió Atenea con un nuevo recurso que permite a los estudiantes planificar, gestionar y llevar el control de las tareas inherentes a la realización de un trabajo o proyecto entre varias personas. Para tal efecto escogimos el sistema Trac, que es un sistema gestor de proyectos sencillo y de fácil utilización.

El sistema Moodle-Trac resultante se puso a disposición de los estudiantes de una asignatura denominada Gestión de Sistemas de Información (GSI), y a algunos estudiantes que realizaban su Proyecto Final de Carrera.

La experiencia fue realmente positiva y la evaluación de la herramienta en los dos contextos fue positiva, pues permitía en gran medida organizarse y planificar las tareas entre todos los participantes (alumnos y profesores), a la vez que les servía como mecanismo de control de la evolución del proyecto. El hecho de tratarse de dos herramientas integradas, hacia transparente el trabajar con dos sistemas distintos y permitía combinar las ventajas de cada uno de los sistemas independientes en una sola plataforma como es Moodle.

De la experiencia realizada, se han detectado algunas mejoras a aplicar para potenciar su uso, ya que algunos alumnos manifestaban que esta coordinación ya la gestionaban a través de otras herramientas que aunque no estaban diseñadas para este propósito, estaban habituados a usar asiduamente y les eran más cómodas.

Una de las particularidades del sistema resultante es que no está restringido a una asignatura concreta, sino que es una herramienta de carácter general y aplicable para dar soporte a cualquier actividad realizada en colaboración. Así

pues, en la UPC, cualquier asignatura que usa Atenea, tiene disponible este nuevo recurso.

Para otros centros que disponen de una instalación de Moodle, con la simple instalación del sistema Trac y de los plugins desarrollados (en Moodle y Trac), el sistema puede ser usado por cualquier asignatura de la universidad.

Agradecimientos

El proyecto presentado en este artículo corresponde a un Proyecto de Innovación Docente financiado por la Convocatòria d'Ajuts per a Projectes de Millora de la Docència (CAPMD) del Institut de Ciències de l'Educació (ICE) de la UPC del año 2009 y por la Convocatòria d'Ajuts per al Finançament de Projectes per a la Millora de la Qualitat Docent a les Universitats Catalanes (MQD) del AGAUR de la Generalitat de Catalunya del año 2009.

Por otro lado, el proyecto ha obtenido el Primer Premio en la 6ª Convocatoria del Premio Davyd Luque a la Innovación en las TIC organizado por UPCNet y la UPC en el año 2010.

Finalmente, agradecer a todas las entidades (y personal de éstas) que han participado directa o indirectamente en la realización del proyecto (FIB, ESSI, UPCnet, ICE, AGAUR, UPC) y a todos los profesores y alumnos de la FIB que han participado en la realización de las encuestas y que han aportado valoraciones y propuestas de mejora al proyecto durante el periodo de realización.

Referencias

- [1] Espacio Europeo de Educación superior, <http://www.eees.es/>
- [2] Marco Europeo de Calificaciones para el Aprendizaje Permanente, ec.europa.eu/education/pub/pdf/general/eqf_broch_es.pdf
- [3] Markham, Th.; Larmer, J.; Ravitz, J. Project Based Learning Handbook: A Guide to Standards-Focused Project Based Learning for Middle and High School Teachers, 2nd Edition, 2003, Buck Institute for Education.
- [4] Bersin, J.; Howard, Ch.; O'Leonard, K.; Mallon, D. Learning Management Systems, Bersin & Associates
- [5] Estadísticas de uso de Moodle, <http://moodle.org/stats/>
- [6] Trac, <http://trac.edgewall.org/>

- [7] Jacobson, I.; Book, G.; Rumbaugh, J. The Unified software Development Process, Addison-Wesley, 1999.
- [8] IEEE, C.S., Guide to the Software Engineering Body of Knowledge: 2004 Edition - SWEBOK. 2005.
- [9] Schwaber, K. Agile Project Management with Scrum. Microsoft Press. 2004.
- [10] Lapuente, R. Adaptació de l'eina Trac a la gestió de projectes acadèmics, Master Thesis, 2010. <http://hdl.handle.net/2099.1/9748>
- [11] A Guide to the Project Management Body of Knowledge (PMBOK® Guide) - Fourth Edition, Spanish, 2009, Project Management Institut

Marco para el desarrollo de la competencia transversal “Comunicación Eficaz”

David López, Alex Ramírez

Dpto. de Arquitectura de Computadores. Universitat Politècnica de
Catalunya. Módulo C6. C/Jordi Girona 1-3. 08034 Barcelona
{david, aramirez}@ac.upc.edu

Resumen

Hay bastante consenso entre los profesores y los empleadores sobre la importancia que tiene para los ingenieros saberse comunicar. También hay consenso respecto a las bajas capacidades comunicativas de nuestros titulados. Esto provoca que la competencia transversal “Comunicación eficaz” sea vista como imprescindible en nuestras titulaciones. Pero, al igual que otras competencias transversales, nos falta base teórica para definir en qué consiste la competencia y cómo trabajarla (y evaluarla). En este trabajo se propone un marco de referencia para desarrollar esta competencia en los estudios superiores de Ingeniería Informática, aunque puede aplicarse a otras ingenierías.

Summary

There is a considerable consensus among teachers and employers about the importance of communication for engineers. At the same time, there is a consensus on the low communication skills of our graduates. So this skill is seen as essential in the new degrees. But, like other professional skills, we lack the theoretical basis for defining what communication is and how to learn and to evaluate it. This paper proposes a framework for developing this skill in the new Computer Science degree, although it can be applied in other Engineering degrees.

Palabras clave

Comunicación eficaz, competencias transversales, competencias profesionales, formación para la profesión.

1. Motivación

Uno de los cambios que nos ha traído la adaptación al Espacio Europeo de Educación Superior (EEES) es la inclusión de las competencias profesionales o transversales en los planes de estudios. No es, sin embargo, una moda europea: en los EE.UU., la *Accreditation Board for Engineering and Technology* (ABET) ya las incluía en sus *Engineering Criteria 2000* [5], y hay un consenso muy amplio en la necesidad de que los futuros ingenieros dominen estas competencias. Sin embargo, los esfuerzos realizados para que el estudiantado adquiriera las competencias no están teniendo el éxito esperado.

Según Evans y Gabriel [2], el problema principal está en que, para muchos académicos, en una competencia transversal se aplican las siguientes ideas: 1) como habilidad, es un conocimiento práctico; 2) por ser práctico, se puede separar de los conocimientos fundacionales de una asignatura, área o campo de conocimiento – en otras palabras, es discreta; 3) al ser discreta, se puede aplicar de manera amplia, es decir, es transversal en el sentido de generalizable a través de diferentes asignaturas, áreas y campos de conocimiento; y 4) siendo práctica, discreta y generalizable, su conocimiento puede ser aprendido una vez, y para toda la vida. Este último punto es el que marca el fracaso de muchas iniciativas para introducir las competencias transversales. Los autores observan que tener una asignatura en el plan de estudios dedicada a una competencia, y aislada de los conocimientos recibidos en el resto de las asignaturas, no es suficiente para que el alumnado llegue a adquirir esta competencia a un nivel adecuado.

Aunque todas las competencias son importantes, y nos permitirán tener mejores

ingenieros, en este artículo nos centraremos en la competencia “Comunicación Eficaz” (o simplemente “Comunicación”), que es una de las competencias en las que hay más consenso respecto a la importancia que tiene para los futuros ingenieros.

Proponemos incluir la competencia en los planes de estudios de cualquier ingeniería (en particular la Informática), pero no por medio de una asignatura o unas pocas actividades, sino por medio de una adecuada planificación a lo largo de los estudios, lo que permitirá que los estudiantes puedan *aprender* a comunicarse.

Para lograrlo, se debe hacer una planificación sobre qué es lo que debe enseñarse (dividir la competencia en subcompetencias), y a qué nivel de profundidad. Con esto se desarrollará un *mapa de la competencia* con el que definir cuándo y cómo se realizará este aprendizaje (asignaturas implicadas, actividades a realizar, etcétera). Finalmente se debe desarrollar el material de soporte y la forma en que podemos evaluar la adquisición de la competencia.

Este artículo propone un mapa de la competencia Comunicación, así como una reflexión de las características que deberían tener tanto las actividades a implementar como el sistema de evaluación de la competencia. Aunque este mapa se esté implementando actualmente en la *Facultat d'Informàtica de Barcelona*, el estado es aún embrionario, y no se pueden aportar resultados del mismo.

2. Las carencias de los ingenieros

En un estudio bastante completo sobre las carencias de los ingenieros realizado en 2004 [3] encontramos una serie de conclusiones que nos deberían llevar a la reflexión:

- La habilidad de comunicarse es “esencial para el éxito del ingeniero”.
- Hay una gran diferencia entre las habilidades comunicativas que *requieren* los puestos de trabajo y las que *tienen* los titulados en ingeniería.
- Las habilidades comunicativas se desarrollan poco o de manera inadecuada en las escuelas de ingeniería.
- Las escuelas que quieren distinguirse por proveer a sus graduados la educación de

mayor calidad deben “desarrollar fuertes programas de comunicación”.

Según el estudio de Tenopir y King de 2004 sobre la comunicación en la actividad de los ingenieros [7], estos dedican más del 50% de su tiempo laboral a generar y recibir información. Un estudio más reciente [6] realizado entre directivos de departamentos de *Information Technologies* (IT), mostró que las 4 habilidades que más apreciaban estos directivos en un futuro jefe de equipo IT eran: 1) liderazgo, 2) capacidad para comunicarse a múltiples niveles, 3) comunicación verbal y 4) comunicación escrita.

Datos de esta naturaleza han llevado a las escuelas de ingeniería a tomar cartas en el asunto. Durante muchos años ha habido una proliferación de cursos de comunicación en universidades de ingeniería en EE.UU. y Canadá. Sin embargo, los estudiantes de ingeniería siguen puntuando muy bajo en las habilidades de comunicación. ¿Por qué? A nuestro entender por dos motivos: 1) se suele dar un curso de “comunicación” en lugar de introducir realmente la competencia en el plan de estudios y 2) el profesorado no se implica suficientemente en la competencia.

¿Por qué no se implica el profesorado? Debemos reconocer que los ingenieros no hemos aprendido nunca a comunicarnos. Sólo aquellos con un don natural para la comunicación saben hacerlo. Muchos profesores no son conscientes de su baja capacidad de comunicación, pero sólo hace falta asistir a una clase o una presentación de un artículo por parte de uno de nuestros colegas para observar el terrible nivel medio de comunicación de los profesores de Ingeniería en España. Y lo que es peor: muchos profesores no dan importancia a la comunicación.

Como ejemplo, nos centraremos en un aspecto: la escritura. Hay un problema en la capacidad de escritura de nuestros alumnos que los profesores de Ingeniería no reconocemos como propio. En la Universidad española suponemos que un estudiante sabe escribir (en el sentido literario del término) a pesar de las evidencias en contra que encontramos cada día. Desgraciadamente, muchos profesores opinan que, si los alumnos no saben escribir, no es en la Universidad donde deben aprender, ya que deberían venir con la lección aprendida. Si juntamos ambos datos (ni saben ni les

enseñamos), tenemos como resultado ingenieros que no se comunican correctamente por escrito.

¿Dónde aprenden a escribir nuestros estudiantes antes de llegar a la Universidad? No es en clases de matemáticas, ni de física, ni de tecnología. Es en clases de lengua y literatura donde se aprende escritura literaria, que tiene unas características muy diferentes a la escritura necesaria en Ingeniería. Además, los estudiantes tienden a aislar el conocimiento, de manera que lo que aprenden en una asignatura nunca se relaciona con lo que se aprende en otra. Así, lo aprendido en literatura no tiene nada que ver con física, y nuestros estudiantes, habiendo elegido el bachillerato tecnológico, no han aprendido a escribir manuales ni memorias técnicas, ni creen que tenga nada que ver con la Ingeniería.

Además, nuestros estudiantes no perciben como importante la comunicación escrita debido a que pueden entregarnos memorias con faltas de ortografía, con frases sin coherencia de tiempo o género (o directamente sin sentido) y nadie les dice nada. Algunos profesores opinan que no forma parte de sus atribuciones evaluar si el trabajo está bien escrito. Otros no quieren dedicar una parte de la nota a algo que no sea técnico. La cuestión puede ser discutible, pero debemos reflexionar sobre el mensaje que transmitimos a nuestros estudiantes si ignoramos la importancia de una escritura correcta.

Pero la competencia Comunicación es mucho más que saber escribir una memoria. También es saber preparar presentaciones y llevarlas a buen puerto ante diferentes tipos de público, asumiendo el rol necesario en cada caso, así como resumir, escuchar, dialogar, buscar la información más relevante, extraer conclusiones, o participar de manera constructiva en un *brainstorming*.

Hay por tanto una estrecha relación con otras competencias tales como trabajo en equipo, pensamiento crítico (*critical thinking*) o uso adecuado de fuentes de información.

3. El mapa de la competencia

La competencia Comunicación es, para muchos, saber escribir un informe y presentarlo con ayuda de la herramienta *powerpoint*. Sin embargo es bastante más que eso. Por ello, el primer paso que debemos dar es identificar qué subcompetencias o atributos componen la competencia comunicación.

Estas subcompetencias (a las que a falta de un nombre mejor, llamaremos *elementos* de la competencia) no se pueden trabajar todas al mismo nivel ni al mismo tiempo. Hace falta, pues, definir para cada uno de estos *elementos* objetivos específicos para los tres niveles de adquisición que corresponderían a los tres primeros niveles de la taxonomía de Bloom [1] (conocimiento, comprensión y aplicación), que son los que se IEEE y ACM consideran que se deben adquirir en los estudios de grado. A la matriz resultante de cruzar los elementos de la competencia con los objetivos para cada nivel lo denominaremos el *Mapa de la Competencia*.

3.1. Elementos que componen la competencia Comunicación

Para la competencia Comunicación, tras un largo periodo de consulta bibliográfica y de reflexión, se han identificado los siguientes elementos:

- *Utilización de gráficos*. Los gráficos se usan principalmente para comunicar información de forma intuitiva y estructurada. Los estudiantes deben saber identificar el tipo de gráfico más adecuado para la comunicación (oral o escrita) que se desea realizar, ya que es fundamental para el éxito de la misma.
- *Capacidad de síntesis*. Identificar las partes más importantes de un proyecto o escrito, y seleccionar cuáles se desean comunicar es una parte fundamental de la comunicación. Por ejemplo, muchos estudiantes en el momento de presentar el proyecto de final de carrera fallan a la hora de ser sintéticos (si por ellos fuera, se pasarían horas explicando hasta la más nimia decisión de diseño).
- *Elaboración de argumentos, razonamientos y conclusiones*. Comunicar es convencer. Para ello hay que saber argumentar, realizar razonamientos claros y fáciles de seguir por el destinatario de la comunicación; y sobre todo hay que saber presentar unas conclusiones breves, precisas y claras. No debemos olvidar que una charla será un éxito si al cabo de unos días el público es capaz de recordar algunas conclusiones. Y sólo se conseguirá si estas han sido bien elaboradas y presentadas.
- *Elaboración de ejemplos, metáforas y símiles*. En el caso de informes o presentaciones orales, comunicarse es acompañar a alguien por un camino que el autor ya ha recorrido.

Para ello hay que ser capaz de simplificar el camino todo lo posible. Y no todo se puede explicar con fórmulas. Los estudiantes deben saber elaborar ejemplos adaptados al público receptor de la comunicación, y usar con soltura metáforas y símiles, que son la marca del buen comunicador.

- *Elaboración de memorias escritas e informes.* Cada tipo de documento tiene unas características especiales. Los alumnos deben tener unos criterios claros sobre cómo se elabora cada uno de ellos, y metodologías para organizar, escribir, incluir citas, etcétera.
- *Presentaciones públicas.* Los alumnos deben conocer las reglas básicas de una buena presentación: desde la organización hasta el uso de lenguaje corporal, pasando por técnicas para atraer y mantener la atención, cómo contestar preguntas, etcétera.
- *Participación en debates y actividades tipo brainstorming.* Saber comunicarse con un grupo también forma parte de la competencia. El estudiante debe saber las normas básicas de un debate (y de un *brainstorming*, que son diferentes), y cuáles actitudes son positivas y cuáles negativas. Saber cuándo intervenir y cuándo dejar hablar a los demás, usar ejemplos, convencer y dejarse convencer, estar abierto a ideas, ser respetuoso, no interrumpir... en resumen, *saber aportar* al debate es un tema muchas veces olvidado en el arte de la comunicación.
- *Comunicación interpersonal.* Incluye escucha activa y comunicación asertiva. La escucha activa se define como la capacidad de comunicar, con lenguaje no verbal, que estás atento a lo que te están diciendo (asentir de vez en cuando, hacer pequeñas preguntas para aclarar ideas, pero sin cambiar de tema, etc). La comunicación asertiva se define como un comportamiento comunicacional maduro en el cual la persona no agrede ni se somete a la voluntad de otros, sino que manifiesta sus convicciones y defiende sus derechos.

Algunos de estos elementos suelen ser ignorados en las propuestas de aprendizaje de la competencia. Sin embargo, son fundamentales en la comunicación.

3.2. Objetivos específicos

Los objetivos específicos para cada uno de los elementos del apartado anterior se pueden encontrar en la Tabla 1.

En esta tabla, para cada uno de los elementos (o subcompetencias) se ofrecen los objetivos específicos de nivel 1, 2 y 3 (correspondientes a los niveles de conocimiento, comprensión y aplicación según Bloom). Nótese que esto no es una rúbrica, y por tanto no sigue la estructura de la misma. Se están utilizando adjetivos que serían inadecuados para una rúbrica dado que no son fácilmente mesurables (por ejemplo: *conclusiones adecuadas* o *buena presentación*). La función de los objetivos específicos es describir qué se pretende conseguir con las actividades que se planifiquen para alcanzar dichos objetivos. Sin embargo, los verbos usados (describir, enumerar, identificar, completar, construir, preparar, etcétera) sí son verbos de acción centrados en el estudiante, y están orientados a facilitar el desarrollo de actividades para cubrir estos objetivos. Posteriormente, en la planificación de las actividades específicas y especialmente en las rúbricas que se usen para evaluarlas, es donde debe utilizarse un lenguaje mucho más objetivo y measurable.

4. Estrategias

El mapa de la competencia nos permite ver tres puntos débiles de algunas propuestas relacionadas con la competencia Comunicación:

- Muchas de las propuestas se centran en un par de elementos de la competencia (la parte oral y escrita), ignorando el resto.
- Cuando se proponen actividades en asignaturas del plan de estudios, suelen ser directamente de nivel 3.
- Intentamos avanzar en todos los elementos a la misma velocidad, cuando es más que posible que se necesiten unos prerrequisitos (por ejemplo, sería interesante haber superado el nivel 2 del elemento *capacidad de síntesis* antes de atacar el nivel 2 del elemento *elaboración de memorias escritas e informes*)

Es decir, solemos trabajar pocos elementos, en paralelo y directamente al nivel de aplicación. Esto lleva a mejorar unos pocos aspectos, pero se

fracasa a la hora de conseguir que los estudiantes adquieran una buena capacidad de comunicación.

¿Podemos hacer todo el mapa en una única asignatura? La respuesta es que probablemente es imposible. La mejor manera de aprender la competencia es a lo largo de todo el plan de estudios, lo que además permite integrarla como parte de la profesión.

Tabla 1: Mapa de la competencia *Comunicación*

4.1. La asignatura “Comunicación”

Viendo el mapa de la competencia, se podría pensar en tener una asignatura dedicada a

Elemento	Objetivos de nivel 1	Objetivos de nivel 2	Objetivos de nivel 3
Utilización de gráficos	Definir los diferentes tipos de gráficos de soporte. Explicar en qué casos se suelen usar.	Identificar, dado un gráfico, la información más relevante. Identificar la información redundante o innecesaria. Proponer mejoras.	Dada una información a representar, decidir qué tipo de gráfico es el mejor en función del medio y el público potencial. Implementar este gráfico
Capacidad de síntesis	Dada una fuente (texto, clase, conferencia,...) determinar las informaciones más relevantes / ideas más importantes.	Dada una fuente y un resumen, realizar una crítica del resumen: adecuación de la longitud, ideas que ha obviado o a las que ha dado demasiada importancia, ...	Elaborar un resumen de una fuente, bajo unas restricciones (de tiempo, de espacio, de medio de presentación, ...)
Elaboración de argumentos, razonamientos y conclusiones	Conocer las bases de la argumentación, y cómo elaborar unas conclusiones adecuadas en función del medio (informe, presentación, ...)	Dada una memoria o presentación, identificar estos elementos. Detectar si se ajustan a las bases conocidas.	Dado un trabajo, elaborar una línea argumental, razonando los pasos que se siguen y elaborando y justificando unas conclusiones adecuadas.
Elaboración de ejemplos, metáforas y símiles	Describir en qué consisten estas figuras (tema ligado con lingüística y literatura). Identificar estos elementos en una actividad (texto, clase, presentación, ...)	Dada una actividad, evaluar la efectividad de las figuras, proponiendo alternativas.	Dado un trabajo (propio o ajeno), elaborar las figuras necesarias para una buena explicación.
Elaboración de memorias escritas e informes	Reconocer los diferentes tipos de comunicaciones escritas. Definir las características de una comunicación escrita eficiente y enumerar los pasos a seguir para su elaboración.	Identificar, ante un caso ejemplo, qué criterios cumple y qué deficiencias tiene. Completar ejemplos incompletos. Proponer mejoras.	Dado un trabajo (propio o ajeno), elaborar una memoria o informe escrito adecuado a los criterios aprendidos. Seguir correctamente los pasos de elaboración de una comunicación escrita.
Presentaciones públicas	Describir las reglas básicas de una buena presentación y del material de apoyo (transparencias, vídeo, ...). Describir las técnicas para mantener la atención del público, lenguaje no verbal, ...	Dada una presentación a la que se asiste, identificar qué se ha hecho bien y qué se ha hecho mal. Reconocer las técnicas utilizadas para realizar la presentación. Proponer mejoras.	Dado un trabajo (propio o ajeno), realizar una presentación pública siguiendo las reglas estudiadas. Romper las reglas de manera adecuada.
Participación en debates y actividades tipo <i>brainstorming</i>	Describir en qué consiste la técnica de <i>brainstorming</i> . Describir las reglas básicas de un buen debate o un <i>brainstorming</i> .	Determinar, dado un <i>brainstorming</i> o un debate al que se ha asistido, qué se ha hecho bien y qué se ha hecho mal. Identificar actitudes positivas y negativas.	Organizar y dirigir un debate o <i>brainstorming</i> . Plantear las reglas y resolver conflictos.
Comunicación interpersonal	Explicar en qué consiste la escucha activa y la comunicación asertiva. Enumerar las características de una comunicación efectiva.	Autoevaluar la capacidad del alumno de estas técnicas. Dada una comunicación, identificar cuándo se usan estas técnicas y su efectividad.	Participar en una comunicación interpersonal usando las técnicas aprendidas.

comunicación en los primeros cursos, donde se incidiera en los niveles 1 y 2 (conocimiento y comprensión) de todos los elementos, de manera que en las siguientes asignaturas ya pudiéramos realizar actividades orientadas al nivel 3. Sin embargo, la competencia Comunicación no tiene por qué ser más importante que otras, y no podemos dedicar una asignatura a cada competencia, ni hacer una *única* asignatura dedicada a *todas* las competencias, pues al final no serviría de nada.

Si se desea dedicar una asignatura a la comunicación, creemos que es mejor un seminario en últimos cursos dedicado a trabajar el nivel 3 de algunos elementos, usando realimentación. Lo ideal sería un seminario de pocos alumnos (20 como máximo) donde se desarrollaran debates, escritura, presentaciones, comunicación personal, etcétera, pero orientado a que cada estudiante descubriera sus fortalezas y debilidades en comunicación, con consejos sobre cómo potenciar las primeras y mejorar en las segundas. Pero antes de este seminario, el estudiante debe haber aprendido las bases y haber practicado.

4.2. Distribución de las competencias a lo largo del plan de estudios

Si queremos que realmente nuestros estudiantes adquieran la competencia “Comunicación”, hace falta distribuir las actividades relacionadas a lo largo del plan de estudios. Para ello es necesario:

- Planificar en qué asignaturas se quiere adquirir cada elemento de la competencia a cada nivel. Una asignatura que trabaje la competencia Comunicación no tiene por qué trabajar todos los elementos de la misma, ni tratar los elementos que trabaje al mismo nivel. Hay que decidir qué celdas del mapa de la competencia se trabajarán en cada asignatura. Un ejemplo podría ser: en la asignatura Estadística se trabajará el elemento “Utilización de gráficos” a nivel 1 y 2, y “Capacidad de síntesis” a nivel 1. No resulta complicado adquirir el nivel 1 y 2 al mismo tiempo si se realizan las actividades adecuadas (lo veremos en la siguiente sección). Posteriormente, en la asignatura Proyecto de Redes se trabajará el elemento “Participación en debates y actividades tipo *brainstorming*” a nivel 1 y 2, y “Capacidad de síntesis” a nivel

2. Y así hasta mapear sobre cada celda al menos una asignatura.

- A la hora de planificar, no hay que forzar las cosas. La distribución de elementos debe ser adecuada a la naturaleza de la asignatura (hay asignaturas donde la elaboración de informes o las presentaciones orales ya forman parte de las actividades). Así pues, más que forzar que se realicen presentaciones en una asignatura donde nunca se han hecho, mejor dejarlo para una asignatura donde siempre se ha defendido de manera oral un proyecto.
- No forzar que muchas asignaturas del mismo nivel trabajen la misma competencia. Trabajar comunicación en 3 asignaturas del tercer cuatrimestre y en ninguna del cuarto cuatrimestre no es una buena solución.
- Una vez adquirida una competencia, exigirla. Si en una asignatura se ha trabajado la *capacidad de síntesis* a nivel 2, las asignaturas posteriores deberían exigir un nivel de *capacidad de síntesis* acorde con lo que sabe el alumno. Esto exige una coordinación vertical, de manera que los profesores de una asignatura sepan el nivel de adquisición de cada elemento de la competencia que (se supone) el alumno ha alcanzado. Esto permitirá que se asienten los conocimientos adquiridos de la mejor manera posible: practicando. Al mismo tiempo se creará una apreciación entre el estudiantado de la importancia de la competencia.

5. Actividades y evaluación

5.1. Tipo de actividades

No vamos a incluir en esta sección un listado exhaustivo de actividades, pues no es el objetivo de este artículo. Las actividades que se pueden realizar son muchas, y la naturaleza del mapa de la competencia invita al profesor de la asignatura a desarrollar sus propias actividades, ya que como ya se dijo anteriormente, los objetivos se escribieron de manera que sugieran la naturaleza de las actividades a realizar.

Por ejemplo, el elemento “Capacidad de síntesis” tiene como objetivos de nivel 2: *Dada una fuente y un resumen, realizar una crítica del resumen: adecuación de la longitud, ideas que ha obviado o a las que ha dado demasiada*

importancia, etcétera. Esto se puede hacer de muchas maneras:

- El profesor reparte, tras la clase, un resumen escrito pero incompleto de su lección, y pide a los estudiantes que lo completen con sus apuntes (trabajo fuera de horas de clase). Posteriormente, los estudiantes intercambian su trabajo y lo evalúan con ayuda de una rúbrica.
- Los estudiantes observan en sus casas una clase (o un tema, o una demostración) grabada en vídeo. En clase se establece un debate para que acuerden los 3 puntos más importantes de la clase visionada. Puede hacerse entre toda la clase o en grupos. También puede pedirse que contesten un cuestionario con preguntas que lleven a la reflexión como: “¿Cuáles son las 3 ideas más relevantes de la presentación?”, “Indica 5 cosas que no se han hecho bien en las transparencias de la presentación”, “¿Qué tipo de gráfico hubieras utilizado en lugar del que aparece en la transparencia 12?”, dependiendo de qué otros elementos de la competencia se traten en la asignatura.
- Los estudiantes leen, con anterioridad a la clase, un resumen de la misma contestando una serie de preguntas en un cuestionario previo. Después, el profesor imparte su clase y los alumnos critican si el resumen es adecuado a para la clase recibida.

Estos ejemplos tienen un denominador común: se delega trabajo en los estudiantes, de manera que el modelo “profesor explica de manera activa ante alumnos pasivos” no tiene cabida aquí. Todas las propuestas implican que el estudiantado pase unas horas pensando en la asignatura, debido a que discuten cuáles son los temas principales o estudian un resumen antes de recibir la lección. Todo esto lleva a aprovechar mucho más las horas de contacto con el profesor, de manera que aunque se dediquen menos horas “a explicar”, se dedican más a pensar y aprender, por lo que trabajar una competencia no es una pérdida de tiempo, sino una ganancia del aprendizaje.

Esto nos lleva a una reflexión que deberíamos hacer antes de preparar las actividades: pensar en la carga de trabajo. Los créditos ECTS nos marcan las horas de trabajo que debería dedicar un estudiante medio para aprobar la asignatura.

Debemos contar con esas horas, y aprovecharlas en actividades que incidan en el aprendizaje. Pero debemos saber calcular cuántas horas ocuparán nuestras actividades, y distribuirlas de manera eficaz. Si tenemos, por ejemplo, una asignatura de 6 ECTS y contabilizamos 25 horas de trabajo por crédito, y entre 17 y 18 semanas por cuatrimestre, el estudiante debería dedicar una media de 8,5 horas a la semana a nuestra asignatura. Lo mejor es planificar *exactamente* 8,5 horas de trabajo cada semana. Eso significa que si sabemos que la semana que los estudiantes tienen un examen de nuestra asignatura, dedican más horas a estudiar, esa semana deberíamos reducir el resto de las actividades. La buena planificación es fundamental para el éxito del plan de estudios, ya que si esta no existe, las diferentes asignaturas compiten por el tiempo de los alumnos, y estos perciben que trabajan demasiado.

Igualmente, hemos de contemplar el trabajo del profesor. No nos extenderemos aquí en consideraciones al respecto, porque hay buenos trabajos al respecto, como Sánchez *et al* [4]; solo comentar que deberíamos aprovechar que los niveles 1 y 2 se pueden desarrollar con actividades que, al ser de conocimiento y comprensión, los alumnos pueden autoevaluarse o hacer evaluación entre pares con ayuda de unas buenas rúbricas. Igualmente, se pueden hacer actividades basadas en lectura, visualización o análisis de cierto material, y deberíamos pensar que en comunicación hay muchas cosas hechas, por lo que podemos utilizar material existente.

5.2. Reflexiones sobre la evaluación

Para que los estudiantes se tomen en serio las competencias transversales, es imprescindible que se evalúen. Esta evaluación puede ser una nota desligada de la “nota técnica” de la asignatura, o bien formar parte de dicha nota, como los laboratorios o el proyecto. Creemos que la segunda opción es la mejor, pero algunos profesores tienen problemas en que la evaluación de una competencia como Comunicación forme parte de la nota. A estos profesores les pedimos que mediten en muchos elementos de la competencia, como saber usar gráficas o capacidad de síntesis, y nos digan si no la están evaluando ya, aunque no lo digan explícitamente: una respuesta bien sintetizada, con buenas conclusiones, debidamente razonada o con buenas

gráficas suele obtener mejor nota que otra solución que no cumpla estos parámetros.

Para la evaluación, lo más apropiado es el uso de rúbricas, no sólo porque permite ser más objetivos, sino porque los estudiantes, si conocen la rúbrica, saben lo que les pediremos y se centrarán en ello (y si se ha tenido en cuenta en el diseño de la rúbrica, mejoraremos el aprendizaje).

Una de las cosas en las que hay que incidir es en que una vez adquirida una competencia se ha de exigir. Así pues, si los estudiantes ya han trabajado el elemento *elaboración de informes* a nivel 3, en las asignaturas posteriores debería exigirse, aunque no se evalúe. Hay muchas soluciones, en particular que la escritura correcta sea *conditio sine qua non* para la aceptación de un informe (podemos devolver un informe corregido y con nota, pero exigir una reescritura antes de aceptarlo definitivamente).

Por otro lado, no debería preocuparnos hacer actividades de nivel 3 para evaluar el nivel 2. Por ejemplo, trabajar el nivel 2 de *presentaciones públicas* no significa que los estudiantes no realicen presentaciones (como parece desprenderse de los objetivos, pues realizar presentaciones está a nivel 3), sino que los objetivos pedagógicos (y la evaluación) se centra en el nivel 2, por lo que se evaluará a los estudiantes por los análisis de las presentaciones más que por las presentaciones en sí.

Para finalizar, una última reflexión: se ha demostrado que la realimentación es un elemento fundamental en el proceso de aprendizaje. En el caso de las competencias transversales, más que fundamental es imprescindible para la buena adquisición de estas competencias.

6. Conclusiones

Una competencia transversal no se puede aprender de una vez y para siempre. Hay que practicarla lo largo de toda la carrera, y por tanto debe introducirse en el plan de estudios en diversas asignaturas que la evalúen explícitamente. Además, las asignaturas que no evalúen la competencia explícitamente deben exigir que se cumpla el nivel adquirido, para que el estudiantado practique sus destrezas.

No es fácil introducir una competencia transversal en el plan de estudios. Es interesante utilizar una herramienta como el mapa de la competencia, similar al que se ha presentado en este trabajo para la competencia Comunicación.

Agradecimientos

A la *Facultat d'Informàtica de Barcelona*, por su apoyo, y en especial a los coordinadores de competencias transversales por sus aportaciones y sus constantes críticas, todas constructivas.

Referencias

- [1] Bloom, B.S., M. D. Engelhart, E. J. Furst, W. H. Hill and D. R. Krathwohl, *Taxonomy of educational objectives: Handbook I, The cognitive domain*, NY, David McKay & Co., 1956.
- [2] Evans, R. and J. Gabriel. *Performing Engineering: How the Performance Metaphor for Engineering Can Transform Communications Learning and Teaching*. 37th Frontiers in Education Conference, pp. T3B-9/14. Milwaukee, WI, October 2007.
- [3] Reave, L. *Technical Communications Instruction in Engineering Schools: A Survey of Top-ranked US and Canadian Schools*. Journal of Bussines and Technical Communications, 18 (4): 452-490. 2004
- [4] Sánchez, F, J. J. Escribano, M. J. García, J. González, E. Millán "Ideas para reducir el trabajo del prof-EEES-or". JENUI 2010, julio de 2010, pp 301-308.
- [5] Shuman, L.J., M. Besterfield-Sacre and J. McGourty. *The ABET Professional Skills—Can They Be Taught? Can They Be Assessed?* Journal of Engineering Education, 94(1), 2005, pp. 41–55.
- [6] Stevenson, D.H. and J.A. Starkweather. *PM critical competency index: IT execs prefer soft skills*. International Journal of Project Management, 28(7), Oct. 2010, Pp 663-671
- [7] Tenopir, C. and D. W. King, *Communication Patterns of Engineers*. New York: IEEE/Wiley InterScience, 2004.

Docencia en Desarrollo Global de Software: Una Revisión Sistemática

Miguel J. Monasor, Aurora Vizcaíno, Mario Piattini

Grupo de Investigación Alarcos
Instituto de Tecnologías y Sistemas de Información
Escuela Superior de Informática, Universidad de Castilla-La Mancha
13071, Ciudad Real, España
MiguelJ.Monasor@gmail.com, {Aurora.Vizcaino, Mario.Piattini}@uclm.es

Resumen

En este artículo presentamos los resultados de una Revisión Sistemática de la Literatura en el campo de la educación y el entrenamiento de habilidades convenientes para el Desarrollo Global de Software (DGS). Nuestro objetivo consiste en recopilar y estudiar las diferentes propuestas y estrategias empleadas en este campo que sean de utilidad tanto para profesionales como para investigadores y que permitan identificar las mejores prácticas que se deben cubrir para afrontar los problemas que conlleva el DGS.

Summary

In this paper we present the results of a Systematic Literature Review in the field of Global Software Development (GSD) training and education. Our aim is to collect and study the various proposals and strategies employed in this field that may be useful for both professionals and researchers and to identify best practices that should be covered to address the problems that GSD entails.

Palabras clave

Desarrollo Global de Software, Desarrollo Distribuido de Software, Enseñanza, Educación, Revisión Sistemática de la Literatura.

1. Introducción

El Desarrollo Global de Software (DGS) es un paradigma emergente que consiste en que los miembros involucrados en el desarrollo de software permanecen geográficamente distribuidos más allá de las fronteras de un país [16]. La principal razón de su aplicación radica en que de este modo se optimizan los recursos y se

reducen los costes a través de la expansión hacia zonas más viables económicamente y donde existe una mayor disponibilidad de profesionales cualificados.

Sin embargo, este tipo de desarrollo también conlleva ciertas desventajas, principalmente debidas a la distancia que separa a los equipos, así como las diferencias temporales, culturales y de lenguaje. Estos inconvenientes frecuentemente dificultan el entendimiento entre los participantes del proyecto, especialmente cuando éstos deben usar un lenguaje común (no nativo), pudiendo surgir malentendidos que afectan a la comunicación y la coordinación del trabajo y que podrían suponer un riesgo para el proyecto [25].

Frecuentemente, los responsables de las empresas de desarrollo de software indican que los recién licenciados carecen de las habilidades necesarias para abordar los nuevos problemas que implica el DGS. Argumentan que su experiencia se limita estrictamente a proyectos relativamente cortos, y que los programas educativos no se ocupan de estos temas a un nivel apropiado [10].

Con el objetivo de proporcionar a los estudiantes experiencias reales que les permitan desarrollar las habilidades técnicas, destrezas y competencias requeridas en DGS, son necesarios nuevos contenidos teóricos y herramientas, así como coordinación entre universidades distantes [32]. Sin embargo estos requisitos rara vez son tenidos en cuenta hasta el momento debido a las dificultades que conllevan [14].

Nuestra línea de investigación se centra en el diseño de métodos y técnicas de enseñanza adecuadas para el entrenamiento de los miembros involucrados en DGS. En este trabajo presentamos los resultados de una Revisión Sistemática de la Literatura (RSL) en el campo de la educación y el entrenamiento del DGS con el

objetivo de ofrecer a profesionales, investigadores y educadores una visión rigurosa de los principales retos, estrategias y propuestas disponibles hasta la fecha.

2. Procedimiento de la Revisión Sistemática

Una RSL permite identificar, evaluar e interpretar todos los estudios relevantes disponibles en relación con una pregunta de investigación, tema o fenómeno, de acuerdo con una estrategia predefinida. En este trabajo hemos aplicado el método para realizar RSLs proporcionado por Kitchenham y Charters [18] con la intención de responder a la siguiente pregunta de investigación:

¿Cuáles son las iniciativas llevadas a cabo en relación con la educación y el entrenamiento del Desarrollo Global de Software?

El objetivo consiste en identificar los mejores procedimientos, modelos y estrategias utilizados en la formación y el entrenamiento de los ingenieros de software. Para ello, se estableció la siguiente cadena de búsqueda:

("distributed software development" OR "global software development" OR "global software engineering" OR "distributed software engineering") AND ("learning" OR "teaching" OR "education" OR "training" OR "simulation" OR "simulator").

La estrategia de búsqueda se basó en las siguientes decisiones:

- **Fuentes de búsqueda:** La cadena de búsqueda se adaptó a los siguientes motores de búsqueda:
 - o Science@Direct (www.sciencedirect.com)
 - o SpringerLink (www.springerlink.com)
 - o IEEE Digital Library (www.computer.org)
 - o ACM Digital Library (<http://portal.acm.org>)
 - o Wiley Interscience (www.interscience.wiley.com)
- **Tipo de estudios considerados:** artículos de conferencias, artículos de revistas y talleres.
- **Periodo de publicación:** Desde el año 2000. Puesto que el DGS es una tendencia reciente como se muestra en la revisión sistemática realizada en [38], donde el 74 % de los artículos encontrados eran posteriores a 2006.

2.1. Selección de estudios primarios

Los criterios de inclusión para la selección de los estudios primarios son los siguientes:

- Estudios que describen cursos de DGS en entornos universitarios o empresas.
- Estudios que describen experiencias reales, problemas o factores de éxito.
- Estudios que proponen herramientas de formación o entornos para llevar a cabo el proceso de formación.

Se excluyeron los estudios que cumplían los siguientes criterios:

- Estudios que no responden de manera rigurosa a la pregunta de investigación.
- Estudios que no aportan ninguna propuesta o información relevante relativas a la formación del DGS o que no enmarcan rigurosamente su propuesta en el contexto del DGS.

Tras la selección inicial de estudios primarios, se llevó a cabo la evaluación de su calidad en dos etapas. Primero se revisó su idoneidad, teniendo en cuenta los criterios de inclusión y exclusión, y después se llevó a cabo una revisión más precisa en paralelo con el proceso de extracción de información. Durante este proceso, se verificó la relevancia y la calidad de los estudios, teniendo en cuenta la claridad de sus métodos y propuestas.

A través del procedimiento de búsqueda (detallado en la Figura 1) se encontraron 38 estudios primarios, listados en la Tabla 1.

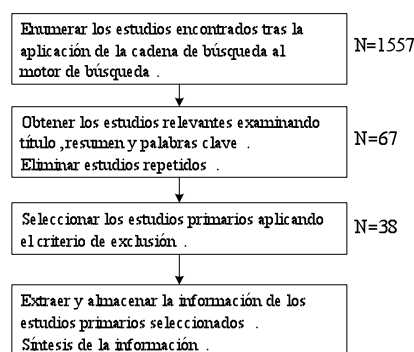


Figura 1. Proceso de selección de estudios primarios

Para cada estudio primario se aplicó un proceso de extracción de información a través de un formulario de datos predefinido con la siguiente información: título, autores, referencia, año, país

de los investigadores, origen, número de páginas, alcance, propuesta, tipo de la organización (universidad, empresa) y tamaño de la empresa, procesos cubiertos, población objetivo, fecha de la evaluación y metodología seguida.

Ref.	Fuente	Ref.	Fuente	Ref.	Fuente
[5]	IEEE	[20]	IEEE	[24]	ACM
[11]	IEEE	[17]	ACM	[23]	IEEE
[8]	IEEE	[1]	ACM	[34]	ACM
[9]	IEEE	[7]	IEEE	[39]	ACM
[21]	IEEE	[41]	IEEE	[27]	ACM
[19]	IEEE	[30]	IEEE	[12]	IEEE
[35]	ACM	[14]	IEEE	[28]	IEEE
[3]	IEEE	[26]	ACM	[40]	IEEE
[31]	IEEE	[29]	ACM	[37]	Wiley
[10]	IEEE	[22]	IEEE	[6]	ACM
[2]	IEEE	[32]	IEEE	[15]	IEEE
[13]	Springer Link	[42]	Springer Link	[4]	Springer Link
[33]	Springer Link	[36]	Springer Link		

Tabla 1. Lista de estudios primarios

3. Tendencias en la Investigación de la educación del DGS

En esta sección se analiza el contenido y las características de los estudios primarios encontrados. La metodología de los estudios primarios se clasificó atendiendo a las siguientes categorías: casos de estudio, revisiones de la literatura, experimentos, simulaciones y encuestas. También se aplicó el modelo no experimental para clasificar los estudios que presentan propuestas sin realizar pruebas experimentales.

La Figura 2 muestra que la mayoría son casos de estudio que básicamente describen experiencias en cursos universitarios. No se encontraron revisiones de la literatura en la materia.



Figura 2. Tipo de estudios analizados

La mayoría de los estudios primarios se contextualizan en un entorno universitario, y básicamente describen cómo grupos de estudiantes han llevado a cabo desarrollos conjuntos desde localizaciones distantes. A pesar de esto, es interesante destacar que también se han encontrado algunos enfoques llevados a cabo en empresas así como estudios desarrollados en colaboración entre universidades y empresas.

3.1. Tendencia de las publicaciones

Atendiendo a los principales procesos de desarrollo hacia los que se han enfocado los estudios primarios, detallados en la Tabla 2, se desprende que los mayores esfuerzos se centran principalmente en la construcción de software, diseño de software, ingeniería de requisitos y pruebas de software.

Procesos	Estudios primarios
Construcción del software	[34], [24], [5], [10], [29], [28], [31], [32], [40], [9], [1], [23],
Diseño del software	[34], [5], [10], [28], [31], [32], [40], [9], [1], [14], [20], [19]
Análisis de requisitos	[34], [5], [10], [28], [31], [32], [9], [14]
Pruebas del software	[34], [6], [5], [28], [40], [9], [15]
Gestión del proyecto	[5], [26], [28], [31], [9], [1], [42]
Elicitación de requisitos	[34], [5], [10], [28], [33], [14]
Gestión organizacional	[8], [31], [4], [36], [37]
Gestión de la configuración	[24], [29], [32], [20]
Aseguramiento de la calidad	[26], [29], [28], [13]
Documentación	[34], [29], [1]

Tabla 2. Principales procesos tratados

4. Resultados de la RSL

En esta sección, se sintetizan los desafíos, métodos y propuestas identificadas a través de la RSL, y se discuten los estudios más relevantes.

4.1. Entornos de aprendizaje

Algunos de los estudios primarios seleccionados tratan sobre entornos de aprendizaje que

proporcionan funcionalidades para la formación y el entrenamiento en actividades típicas del DGS.

iBistro [8] es un entorno que, básicamente permite a los miembros distribuidos colaborar en proyectos y aprender conceptos de gestión de proyectos, desarrollo de software y habilidades sociales. iBistro trata los problemas de comunicación entrenando a los estudiantes a través de reuniones informales en las que pueden capturar las estructuras y los conocimientos surgidos de dichas reuniones. Para ello, se apoya en una herramienta que almacena la información contextual de las reuniones y permite representarlas y navegar por su contenido.

En [35] se presenta un entorno colaborativo utilizado en un curso de desarrollo distribuido de software. Dicho entorno utiliza espacios virtuales donde se lleva a cabo la colaboración del equipo. Estos espacios virtuales pueden contener páginas con contenidos y dentro de ellas, los participantes pueden interactuar a través de diferentes canales de comunicación (como chat o correo electrónico).

En [27] se presenta una plataforma colaborativa basada en Web que facilita la comunicación y gestión de contenidos, proporcionando foros de discusión, un repositorio de archivos compartido y un calendario de proyecto. Los instructores pueden añadir módulos formativos, y los estudiantes pueden acceder a las instrucciones, hitos y resultados finales trabajando con sus compañeros de equipo para cumplir los objetivos marcados.

En la misma línea, [40] presenta una propuesta orientada a la enseñanza mediante la colaboración de equipos de estudiantes distribuidos que interactúan haciendo uso de herramientas colaborativas (incluyendo chat, pizarra compartida, herramienta de compartición de aplicaciones, herramientas de diseño de documentos UML, etc.). Los autores también presentan un software de gestión de cursos para ayudar a los profesores en tareas relacionadas con la administración de grupos y la recogida de información de las acciones del estudiante, proporcionando así un medio para evaluarlos.

Con el fin de estandarizar el trabajo diario del proyecto y evitar la ausencia de miembros en reuniones y otros problemas causados por la informalidad típica en estos entornos algunos estudios aplican la idea del **contrato de equipo**

[11], en el que se establecen unas normas relativas a la comunicación, los tiempos de respuesta y responsabilidades.

Finalmente, también hemos encontrado la aplicación de enfoques de *e-learning* como OAS!S [39]; un entorno de enseñanza virtual creado a través de la personalización de WebCT Vista, cuyo uso está ampliamente extendido en muchas universidades. Permite foros de discusión, sistemas de correo, chat y gestión de contenidos. En este sentido, también debemos mencionar la plataforma de aprendizaje OLAT (*Online Learning And Training*) [39], que proporciona características similares al anterior ofreciendo soporte a varios estándares de e-learning como IMS o SCORM.

4.2. Enseñanza del DGS en el aula

Los estudios primarios también describen clases teóricas tradicionales adaptadas a las nuevas necesidades de la Ingeniería de Software [20]. Muchos de ellos coinciden en destacar la necesidad de que los cursos se realicen de manera conjunta entre diferentes universidades de forma que los estudiantes puedan interactuar con alumnos de diferentes culturas [29]. Sin embargo, un problema común de este enfoque radica en la dificultad de lograr un nivel adecuado de coordinación y colaboración con las diferentes universidades [3].

En el Máster europeo en DGS descrito en [20] participaron varias universidades de diferentes países y básicamente se centra en las dimensiones técnicas y culturales del DGS. En este tipo de prácticas se debe tener en cuenta que los estudiantes de diferentes universidades tienen diferentes tipos de formación, aptitudes y experiencia, por lo que es importante analizar su educación previa y caracterizarlos con el fin de preparar los contenidos del curso [19].

Algunos estudios combinan la enseñanza en las aulas con prácticas y desarrollos reales. El desarrollo de las aptitudes necesarias en DGS requiere práctica en los problemas típicos que se pueden presentar en entornos reales, por lo que el enfoque más propuesto consiste en aprender a través de la práctica.

Algunas universidades tienden a organizar actividades prácticas en colaboración con universidades de diferentes países en las que los estudiantes se suelen comunicar mediante correo

electrónico, teléfono y mensajería instantánea [17] para afrontar procesos similares a los aplicados en la industria.

En la experiencia presentada en [31], participaron tres universidades de diferentes países. El estudio destaca la importancia del desarrollo de las habilidades comunicativas informales, incidiendo en que es necesario aprender a trabajar eficazmente en equipo y reaccionar rápidamente a los cambios en los requisitos, la arquitectura y la organización.

En [39] se presenta el desarrollo de dos proyectos en los que participan estudiantes de cuatro países. Durante esta experiencia se estudió el rendimiento del equipo a través de la medición de ciertos factores que afectan al trabajo colaborativo. Igualmente, en el caso presentado en [28] los estudiantes se dividieron en pequeños grupos (en algunos casos distribuidos en diferentes países), que colaboraron para desarrollar una aplicación completa.

En [13], los autores presentan un curso sobre aseguramiento de la calidad aplicado a tres universidades de diferentes países que colaboraron en el desarrollo de un software. Esta práctica permitió a los estudiantes interactuar con compañeros de diferentes culturas, y desempeñar diferentes roles en el proyecto. Los autores también sugieren que los instructores deberían desempeñar el papel de jefes de proyecto; lo que les permitiría guiar a los estudiantes a lo largo del proceso y aprovechar su experiencia para evitar ciertos problemas de coordinación.

4.3. Enseñanza del DGS en la empresa

Aunque la docencia de la enseñanza de DGS en entornos empresariales no es común, los estudios primarios, detallan algunas experiencias. El ejemplo más destacable es el presentado en [30], consistente en una iniciativa de una multinacional que aplica DGS e impartió un curso de formación práctico relacionado con prácticas comunicativas, diferencias culturales, coordinación y confianza entre los participantes. Después de la finalización de cada proyecto, los estudiantes generaban un documento sobre las lecciones aprendidas.

El mayor grado de disponibilidad de personal con experiencia en estos entornos hace posible la aplicación del concepto de **redes de aprendizaje** [13]. Dado que los instructores no pueden ser expertos en todas las áreas del DGS y que en

consecuencia no pueden cubrir todas las áreas, estas redes proporcionan una nueva forma de realizar el proceso de aprendizaje basándose en el conocimiento de un conjunto multidisciplinar de formadores con experiencia.

En [22] se describe un ejemplo de una red de aprendizaje en la que un equipo de expertos en desarrollo de software de una compañía combinaron su trabajo con actividades de formación aprovechando sus experiencias como ingenieros, jefes de proyecto, responsables de calidad, etc., en dicha compañía.

5. Competencias necesarias en DGS

Una vez analizados los enfoques utilizados en la actualidad para la enseñanza del DGS se consideró conveniente estudiar cuales eran las competencias que se pretendían desarrollar en dichos enfoques. Se obtuvieron las siguientes:

- Conocimiento de protocolos y costumbres de las diferentes culturas implicadas en la comunicación [32], [14].
- Capacidad para comunicarse eficientemente usando una terminología y lengua común [42].
- Habilidad para ganarse la confianza del equipo [42] y para resolver conflictos [27].
- Conocimiento de técnicas de negociación y redacción de contratos [10].
- Gestión de la ambigüedad e incertidumbre que derivan de los problemas comunicativos [32].
- Uso de herramientas de gestión del conocimiento, gestión de documentos y de control de versiones [8], [1].
- Capacidad para liderar las reuniones y gestionar el tiempo [22], [28].
- Habilidad para trabajar en equipo y pensar desde la perspectiva del interlocutor [11].
- Capacidad de improvisación y habilidad para la comunicación informal [31].
- Habilidad para tratar con un equipo multidisciplinar [9].

6. Conclusiones y Trabajo Futuro

En este trabajo se ha realizado una RSL con el objetivo de analizar rigurosamente los estudios relacionados con la formación y el entrenamiento del DGS que nos ha llevado a obtener las siguientes conclusiones:

Conclusión 1. En los últimos años existe un creciente interés en la docencia del DGS.

Conclusión 2. La docencia del DGS debe apoyarse en experiencias prácticas a través de las que los estudiantes puedan aprender mientras trabajan en tareas reales.

Conclusión 3. Para las universidades es difícil simular la complejidad de los entornos reales. Coordinar a estudiantes distantes y con diferentes horarios es uno de los principales desafíos.

Conclusión 4. Los instructores no pueden conocer a fondo todos los problemas que se presentan en las diferentes etapas del DGS [22], por lo que se deben especializar en un campo específico.

Conclusión 5. Los estudiantes que participan en programas de formación de DGS suelen experimentar una falta de motivación, problemas para ajustarse a los horarios y dificultades de comunicación [35], y esto se ve acentuado cuando aparecen diferencias culturales y de lenguaje [11].

Conclusión 6. Son necesarias herramientas específicas para la comunicación, colaboración y gestión de documentos. Una adecuada selección de herramientas es por tanto un aspecto clave [10].

Por otra parte, creemos que puesto que el DGS es un área extensa, este estudio podría ampliarse buscando enfoques relacionados que, a pesar de no abordar este tema directamente, se enfoquen hacia ámbitos tales como el aprendizaje de idiomas y las diferencias culturales, por lo que su estudio sería importante en un trabajo futuro.

Este trabajo se enmarca dentro del inicio de nuestra investigación relacionada con la docencia del DGS. El trabajo inmediato consistirá en analizar las deficiencias de estos enfoques para tratar de dar solución a algunas de ellas a través de la propuesta de un marco metodológico para la docencia del DGS.

Como parte de esta metodología, se pretende además desarrollar una herramienta que ofrezca soporte para que los estudiantes puedan desempeñar lo distintos roles que pueden surgir en un proyecto de DGS (por ejemplo, jefe de proyecto, analista, diseñador, implementador, etc.) de esta forma podrán enfrentarse a casi todas las posibles situaciones que pueden aparecer en este tipo de desarrollo.

La información recogida en este trabajo será también útil para diseñar escenarios de entrenamiento apropiados.

Agradecimientos

Este trabajo está financiado por el proyecto PEGASO/MAGO (Ministerio de Ciencia e Innovación MICINN y Fondos FEDER, TIN2009-13718-C02-01). También por los proyectos MEVALHE (HITO-09-126) y ENGLOBAS (PII2109-0147-8235), financiados por la Consejería de Educación y Ciencia (Junta de Comunidades de Castilla-La Mancha), y Fondos FEDER, así como por GLOBALIA (PEII11-0291-5274) (Consejería de Educación y Ciencia, Junta de Comunidades de Castilla-La Mancha) y ORIGIN (IDI-2010043 (1-5)) financiado por CDTI y FEDER.

Referencias

- [1] Adya, M., D. Nath, A. Malik, and V. Sridhar, *Bringing global sourcing into the classroom: experiential learning via software development project*, *Conference on Computer personnel research: The global information technology workforce*. 2007, ACM: St. Louis, Missouri, USA. p. 20-27.
- [2] Ahamed, S.I. *Model for Global Software Engineering Project Life Cycle and How to Use it in Classroom for Preparing Our Students for the Globalization*. in *COMPSAC '06*. 2006. Chicago, Illinois, USA.
- [3] Bellur, U., *An Academic Perspective on Globalization in the Software Industry*, in the *30th Annual International Computer Software and Applications Conference*. 2006, IEEE CS. p. 53-54.
- [4] Berkling, K., M. Geisser, T. Hildenbrand, and F. Rothlauf, *Offshore Software Development: Transferring Research Findings into the Classroom*, in *Software Engineering Approaches for Offshore and Outsourced Development*, S. Berlin, Editor. 2007: Heidelberg. p. 1-18.
- [5] Bmegge, B., A.H. Dutoit, R. Kobylinski, and G. Teubner, *Transatlantic project courses in a university environment*, in the *Seventh Asia-Pacific Software Engineering Conference*. 2000, IEEE CS. p. 30-37.
- [6] Bondi, A.B. and J.P. Ros, *Experience with Training a Remotely Located Performance Test Team in a Quasi-agile Global Environment*, in *ICGSE '09 - 2009*, IEEE CS. p. 254-261.

- [7] Bouillon, P., J. Krinke, and S. Lukosch, *Software Engineering Projects in Distant Teaching*, in *Proceedings of the 18th Conference on Software Engineering Education & Training*. 2005, IEEE CS. p. 147-154.
- [8] Braun, A., A.H. Dutoit, A.G. Harrer, and B. Brüge, *iBistro: A Learning Environment for Knowledge Construction in Distributed Software Engineering Courses*, in *Proceedings of the Ninth Asia-Pacific Software Engineering Conference*. 2002, IEEE CS. p. 197-203.
- [9] Burnell, L.J., J.W. Priest, and J.R. Durrett, *Teaching Distributed Multidisciplinary Software Development*. IEEE Softw., 2002. **19**(5): p. 86-93.
- [10] Damian, D., A. Hadwin, and B. Al-Ani, *Instructional design and assessment strategies for teaching global software development: a framework*, in *Proceedings of the 28th international conference on Software engineering*. 2006, ACM: Shanghai, China. p. 685-690.
- [11] Favela, J. and F. Peña-Mora, *An Experience in Collaborative Software Engineering Education*. IEEE Softw., 2001. **18**(2): p. 47-53.
- [12] Gotel, O., V. Kulkarni, C. Scharff, and L. Neak, *Integration Starts on Day One in Global Software Development Projects*, in *ICGSE'08*. 2008, IEEE CS. Bangalore, India. p. 244-248.
- [13] Gotel, O., V. Kulkarni, C. Scharff, and L. Neak, *Students as Partners and Students as Mentors: An Educational Model for Quality Assurance in Global Software Development*, S. Berlin, Editor. 2009: Heidelberg. p. 90-106.
- [14] Gotel, O., V. Kulkarni, C. Scharff, and L. Neak, *Working Across Borders: Overcoming Culturally-Based Technology Challenges in Student Global Software Development*, in *Proceedings of the 2008 21st Conference on Software Engineering Education and Training*. 2008, IEEE CS. p. 33-40.
- [15] Hackett, M., *Building Effective Global Software Test Teams through Training*, in *ICGSE'07*. 2007, IEEE CS. Munich, Germany. p. 293-294.
- [16] Herbsleb, J.D. and D. Moitra, *Global software development*. IEEE Software, 2001. **18**(2): p. 16-20.
- [17] Honig, W.L. and T. Prasad, *A classroom outsourcing experience for software engineering learning*, in *SIGCSE'07*. 2007, ACM: Dundee, Scotland. p. 181-185.
- [18] Kitchenham, B. and S. Charters, *Guidelines for performing Systematic Literature Reviews in Software Engineering*, in *Technical Report EBSE-2007-001*. 2007, Keele University and Durham University Joint Report.
- [19] Lago, P., H. Muccini, and M.A. Babar, *Developing a Course on Designing Software in Globally Distributed Teams*, in *ICGSE'08*. 2008, IEEE CS. Bangalore, India. p. 249-253.
- [20] Lago, P., et al., *Towards a European Master Programme on Global Software Engineering*, in *Proceedings of the 20th Conference on Software Engineering Education & Training*. 2007, IEEE CS. p. 184-194.
- [21] Liu, X. *Collaborative global software development and education*. in *COMPSAC'05*. 2005. Edinburgh, Scotland.
- [22] Lutz, B., *Training for Global Software Development in an International "Learning Network"*, in *ICGSE'07*. 2007, IEEE CS. p. 140-150.
- [23] Mead, N.R., A. Drommi, D. Shoemaker, and J. Ingalsbe. *A Study of the Impact on Students Understanding Cross Cultural Differences in Software Engineering Work*. in *33rd Annual IEEE International Computer Software and Applications Conference*. 2009. Seattle, Washington, USA.
- [24] Meneely, A. and L. Williams, *On preparing students for distributed software development with a synchronous, collaborative development platform*, in *the 40th ACM technical symposium on Computer science education*. 2009, ACM: Chattanooga, TN, USA. p. 529-533.
- [25] Monasor, M.J., M. Piattini, and A. Vizcaíno, *Challenges and Improvements in Distributed Software Development: A Systematic Review*. *Advances in Software Engineering*, 2009: p. 14.

- [26] Murphy, C., D. Phung, and G. Kaiser, *A distance learning approach to teaching eXtreme programming*. SIGCSE Bull., 2008. **40**(3): p. 199-203.
- [27] Ocker, R., M.B. Rosson, D. Kracaw, and S.R. Hiltz, *Training Students to Work Effectively in Partially Distributed Teams*. Trans. Comput. Educ., 2009. **9**(1): p. 1-24.
- [28] Petkovic, D., G. Thompson, and R. Todtenhoefer, *Teaching practical software engineering and global software engineering: evaluation and comparison*. SIGCSE Bull., 2006. **38**(3): p. 294-298.
- [29] Petkovic, D., G.D. Thompson, and R. Todtenhoefer, *Assessment and comparison of local and global SW engineering practices in a classroom setting*, in *Proceedings of the 13th annual conference on Innovation and technology in computer science education*. 2008, ACM: Madrid, Spain. p. 78-82.
- [30] Prikladnicki, R. and L. Pilatti, *Improving Contextual Skills in Global Software Engineering: A Corporate Training Experience*, in *ICGSE'08*. 2008, IEEE C S. Bangalore, India. p. 239-243.
- [31] Richardson, I., A.E. Milewski, N. Mullick, and P. Keil, *Distributed development: an education perspective on the global studio project*, in *the 28th international conference on Software engineering*. 2006, ACM: Shanghai, China. p. 679-684.
- [32] Richardson, I., et al., *Globalizing Software Development in the Local Classroom*, in *Proceedings of the 20th Conference on Software Engineering Education & Training*. 2007, IEEE CS. p. 64-71.
- [33] Romero, M., A. Vizcaíno, and M. Piattini, *Using Virtual Agents for the Teaching of Requirements Elicitation in GSD*, in *the international conference on Intelligent Virtual Agents*. 2008, Springer-Verlag: Tokyo, Japan. p. 539-540.
- [34] Rusu, A., et al., *Academia-academia-industry collaborations on software engineering projects using local-remote teams*, in *the 40th ACM technical symposium on Computer science education*. 2009, ACM: Chattanooga, TN, USA. p. 301-305.
- [35] Schümmer, T., S. Lukosch, and J.M. Haake, *Teaching distributed software development with the project method*, in *Conference on computer support for collaborative learning: learning 2005: the next 10 years!* 2005, International Society of the Learning Sciences: Taipei, Taiwan. p. 577-586.
- [36] Setamanit, S.-o. and D. Raffo, *Identifying Key Success Factors for Globally Distributed Software Development Project Using Simulation: A Case Study*, in *Making Globally Distributed Software Development a Success Story*, S. Berlin, Editor. 2008: Heidelberg. p. 320-332.
- [37] Setamanit, S.-o., W. Wakeland, and D. Raffo, *Using simulation to evaluate global software development task allocation strategies*. Software Process: Improvement and Practice, 2007. **12**(5): p. 491-503.
- [38] Silva, F.Q.B.d., C. Costa, A.C. C., and R. Prikladnicki. *Challenges and Solutions in Distributed Software Development Project Management: a Systematic Literature Review*. in *ICGSE '10*. 2010. Princeton, NJ.
- [39] Swigger, K., et al., *Structural factors that affect global software development learning team performance*, in *47th annual conference on Computer personnel research*. 2009, ACM: Limerick, Ireland. p. 187-196.
- [40] Swigger, K., et al. *Teaching Students How to Work in Global Software Development Environments*. in *International Conference on Collaborative Computing: Networking, Applications and Worksharing, 2006. CollaborateCom 2006*. 2006. Atlanta, GA, USA.
- [41] Swigger, K., et al., *A Comparison of Team Performance Measures for Global Software Development Student Teams*, in *ICGSE'09*. 2009, IEEE CS. Limerick, Ireland. p. 267-274.
- [42] Toyoda, S., M. Miura, and S. Kunifuji, *A Case Study on Project-Management Training-Support Tools for Japanese/Chinese/Indian Offshore Development Engineers*, in *Knowledge-Based Intelligent Information and Engineering Systems, S. Berlin, Editor*. 2009: Heidelberg. p. 1222-1229.

Sesión 4B:
Arquitectura de computadores

Proyecto para el diseño, montaje y administración de un Cluster de computadores por parte de los estudiantes

Fco. Javier Fernández Baldomero, Mancia Anguita López

Departamento de Arquitectura y Tecnología de Computadores

Universidad de Granada

C. Periodista Daniel Saucedo Aranda s/n

18071-GRANADA

{jfernand,manguita}@ugr.es

Resumen

En el marco de un Proyecto de Innovación Docente de la Universidad de Granada para la asignatura de 5º curso “Arquitectura de Computadores II” (ACII) en el curso 2008-09, realizamos el diseño, compra, montaje, instalación y utilización de un mini-cluster de computadores orientado a aplicaciones científicas. En el curso 2009-10, disponiendo ya del equipo montado, la actividad se ha replanteado como instalación y utilización del cluster, habiéndose concurrido al Programa de Apoyo a Docencia Práctica para la adquisición de hardware adicional con el cual explotar las prestaciones del cluster (servidor de disco, red Infiniband, etc). Esta contribución describe el método aplicado para desarrollar y evaluar la actividad, el cluster diseñado por los estudiantes y su configuración software, así como los resultados de encuestas de opinión de los estudiantes que participaron, posibles variaciones para futuras ediciones de la experiencia, y las conclusiones alcanzadas con la realización del proyecto.

Summary

In the framework of a Teaching Innovation Project of the University of Granada on the “Computer Architecture II” (ACII) subject on 5th course in the 2008-09 academic year, we designed, purchased, assembled, installed and used a mini-cluster of computers aimed at scientific applications. In 2009-10, being the system already designed, the activity has been restated as installation and use of the cluster, having applied to the Lab Teaching Support Program for the purchase of additional hardware

in order to exploit cluster performance (disk server, InfiniBand network, etc). This contribution describes the method used to perform and assess the activity, the cluster designed by the students and its software configuration, as well as the outcome of opinion polls on the students involved, possible variations for future experience editions, and the conclusions achieved with this project.

Palabras clave

Cluster, Linux, Beowulf, arquitectura de computadores, aprendizaje por proyectos.

1. Introducción

El Aprendizaje por Proyectos [8] es un método docente ampliamente reconocido por su capacidad para estimular la participación de los estudiantes y mantenerlos motivados hacia la asignatura. Como características básicas distintivas del método se pueden destacar el menor énfasis en la enseñanza mecánica y memorística para dedicarse a un trabajo más retador y complejo, el enfoque interdisciplinario (apropiado a un trabajo complejo) en lugar de orientado al área o asignatura, y el trabajo cooperativo frente al individual.

Adoptar el método puro de Aprendizaje basado en Proyectos implicaría una fuerte coordinación entre el profesorado de las distintas asignaturas implicadas [5] [9], y en último término una reforma completa en la estrategia docente de los estudios. En este trabajo hemos optado por introducir un pequeño proyecto voluntario en la asignatura ACII, con la intención obvia de mejorar la motivación de los estudiantes, y en segundo término para observar la actitud de los mismos ante lo que podría ser una práctica de

las nuevas asignaturas en los próximos planes de estudio de los Grados.

2. El proyecto y su forma de evaluación

En el contexto del Plan de Innovación Docente de la UGR en convocatoria 2008-09 [7], se nos concedió un presupuesto de 3.650€ que se puso a disposición de los estudiantes para diseñar, comprar, montar, instalar, evaluar y utilizar un pequeño cluster para la ejecución de aplicaciones científicas. Esta finalidad se justifica por el tipo de prácticas que se vienen realizando en la asignatura ACH (paralelización de aplicaciones como, por ejemplo, integración numérica, tratamiento de imágenes, etc.).

El enunciado de la actividad se proporcionó anticipadamente a los estudiantes a través de la web de los profesores [6]. Básicamente, se proponía diseñar el cluster en un plazo de 5 semanas (acabando antes de Semana Santa), para posteriormente instalarle el S.O., pasarle benchmarks (para comprobar si las prestaciones eran las esperadas según el diseño) y que diera tiempo a ejecutar en él las prácticas de la asignatura antes de que acabara el curso. Las normas de evaluación de la asignatura se alteraron de la forma indicada en la Tabla 1. Las ventajas para los estudiantes que participaran eran un menor umbral para aprobar, y un tope de nota algo mayor (11 puntos).

Actividad	resto clase		grupo cluster	
	max	umbr	max	umbr
Examen	7,0	3,5	5,0	1,5
Prácticas	3,0	1,5	3,0	1,0
Proy. Cluster			3,0	1,0
Suma	10,0	5,0	11,0	3,5

Tabla 1. Puntuación de ACH en el curso 2008-09.

Se decidió añadir 3 puntos adicionales, por el trabajo realizado en el Proyecto Cluster (2 puntos) y un cuestionario final sobre el mismo (1 punto), mantener los 3 puntos de prácticas, y prorratear a 5 puntos el examen de teoría. Obsérvese que se podrían obtener 11 puntos, y que se rebajaron los umbrales de aprobado al 30% (frente al 50% “normal”), pudiéndose aprobar holgadamente con 1.5 puntos en el examen final de Teoría.

Las encuestas del curso 2008-09 nos han revelado la buena aceptación por parte de los estudiantes de los umbrales bajos (no tanto de los

prorrates), de forma que en el curso 2009-10 hemos preferido plantear el proyecto (instalación, configuración, evaluación y explotación del cluster) como alternativa a las prácticas normales, con los mismos máximos y umbrales (Tabla 2). La alternativa se denominó “Centro de Procesamiento de Datos”, dando a entender que esta actividad u otras similares podrían practicarse en la futura asignatura de Grado de dicho nombre.

Actividad	máximo	umbral
Examen final	7,0	3,0
Test	3,5	0,7
Ejercicios	3,5	-
Prácticas	3,0	1,25
Suma	10,0	4,25

Tabla 2. Puntuación de ACH en el curso 2009-10.

Para supervisar el desarrollo de la actividad se ha utilizado el sistema SWAD de la UGR [1] [2] [3], aprovechando la zona común de archivos para entregar documentos de trabajo (Figura 1) y el foro para comunicación profesores-estudiantes (Figura 2). En el curso 08-09 se produjeron 16 trabajos y 348 *posts* (58 de los profesores).

La web de la actividad [6] sugería unas cuantas actividades que los estudiantes podían realizar autónomamente (Tabla 3), apuntaba algunas webs de fabricantes y vendedores de donde obtener información para realizar los diseños, y ha servido también para ir anotando comentarios y resumiendo las aportaciones realizadas a lo largo del proyecto.



Figura 1. Zona común de trabajos SWAD, curso 2008-09.

Asunto	Primer mensaje	Último mensaje	Nº mens.	No leídos	Escritores
Dinámica de trabajo y normas del foro	Javier 01/03/09 12:02	Fernánd... 22/06/09 19:20	10	0	3 82
Config. servidores [Pág. 1 2 ... 5 ... 8]	Luis 01/03/09 22:30	Fernánd... 19:29	78	0	15 61
Benchmarks [Pág. 1 2 3]	Ignacio 02/03/09 19:47	Javier 01/05/09 20:59	22	0	9 50
Elección procesador [Pág. 1 2 ... 4 ... 6]	Jose Al... 02/03/09 13:14	Javier 14/04/09 12:07	54	0	17 58
Procesamiento GPU [Pág. 1 2 3 4]	Francis... 04/03/09 14:20	Javier 12/04/09 23:39	34	0	11 41
Red de interconexión [Pág. 1 2]	Manuel 09/03/09 22:01	Javier 07/04/09 21:17	11	0	8 32

Figura 2. Foro SWAD usado para el diseño, curso 2008-09.

La experiencia del curso 2008-09 ha permitido aquilatar de forma más precisa la cantidad de trabajo que los estudiantes consideran apropiada para 3 puntos de calificación, de forma que en el curso 2009-10 sólo se ha planteado la instalación, configuración, evaluación y explotación del cluster ya montado. Las actividades se realizaban presencialmente en horario de prácticas y tutorías, sobre el cluster real, anotando los profesores las tareas en las que cada estudiante participaba activamente. Los estudiantes realizaron su informe [4] como documento GoogleDocs, también enlazado en la web del Proyecto [6].

3. Desarrollo del proyecto

En ambos casos, ya sea como actividad voluntaria adicional (curso 2008-09) o alternativa (2009-10) a las prácticas normales, nos ha sucedido que pocos estudiantes escogen el proyecto, razonando que implica realizar más trabajo que las prácticas normales para obtener la misma nota. En el curso 2008-09 participaron 15 estudiantes de un total de 150 (10%), y en el 2009-10, 7 de un total de 134 (5%). Como incidencias destacables, hay que comentar que en el curso 2008-09 la etapa de diseño colaborativo en el foro no convergió a un diseño consensuado, por lo que hubo que reunir a los estudiantes, agruparlos en 2 grupos (7-8 estudiantes) y encargarles que realizaran el diseño por escrito durante Semana Santa para cumplir los plazos. Paradójicamente, los dos diseños entregados fueron virtualmente idénticos. La etapa de instalación de software tuvo más incidencias, lo cual sirvió de argumento para dedicarle mayor

Actividad	Puntos
aportación de componente, precio, y características técnicas (en el foro)	0,1-0,3
rebatir o comentar aportación previa con precios, detalles técnicos o conceptos clase	0,1-0,3
encontrar defectos diseño en aportaciones	0,1-0,2
encontrar oferta similar a diseño aportado	0,1-0,2
encontrar relaciones entre documentación de componente y conceptos estudiados en clase	0,1-0,2
documentar instalación / config. de software	0,3
tarea de instalación / config. de un software	0,3
otras	según

Tabla 3. Posibles actividades del proyecto (08-09), y su puntuación aproximada.

atención (exclusiva, de hecho) en el curso siguiente, como se detalla en la sección 5. Básicamente, nos proponíamos elucidar si dichas incidencias se debían a una falta de habilidades de base, o a un exceso de tarea. Ambos motivos podrían ser también causa de la baja participación.

En el curso 2009-10 se resaltó que el proyecto consistía exclusivamente en instalar, configurar y evaluar el cluster, y administrarlo durante el periodo de explotación (las últimas semanas del curso) para que los restantes compañeros pudieran ejecutar en él los programas paralelos realizados como prácticas. Como ya se ha indicado, las actividades se han realizado presencialmente, anotando el profesor la participación de cada estudiante. Aparentemente no hubo incidencias, aunque a posteriori algunos estudiantes han mostrado discrepancias con el desarrollo del proyecto, en la encuesta anónima que se les ha realizado (no durante sesiones presenciales). Más adelante extraemos conclusiones a este respecto.

4. Cluster diseñado por los estudiantes

En el curso 2008-09 se realizó una reunión tras los diseños de Semana Santa para afinar algunos detalles, tras la cual se realizó la primera compra por un precio de 3000€ aprox. (Tabla 4). El resto del presupuesto se liquidó posteriormente, tras haber montado el cluster y comprobado su correcto funcionamiento. El colchón de 500€ hubiera permitido reponer algún componente vital si se hubiera estropeado durante el montaje. Al no surgir ninguna contingencia, se aprovechó para darle un acabado profesional al equipo (Tabla 5).

En la Figura 3 se muestra el cluster ACII tras la sesión de montaje, y en la Figura 4 con todos los componentes instalados, las puertas retiradas y los nodos parcialmente extraídos.

Componente	cant.	precio
CPU Intel Core i7-920	3	734,97
Placa Madre ASUS P6T SE	3	662,31
Mem. 6GB DDR3 1066 KVR-N7K3	3	284,97
Switch GbE TL-SG1016	1	76,99
Tarjeta GbE D-Link DGE-530T	1	18,99
cable RJ-45 Cat6 3m	4	15,08
KVM Level-One KVM-0410	1	89,00
cable KVM PS/2 1.8m	4	16,00
HDD 500GB SATA-II	3	143,97
Grabador DVD SATA	1	24,44
Lector DVD SATA	2	27,84
Fuente Alimentación 500W	3	84,87
Tarjeta VGA PCIe GF7200 256MB	3	76,77
Monitor LCD 19" LG W1941S	1	99,99
Teclado Logitech PS/2	1	7,54
Ratón Logitech óptico PS/2	1	5,40
Armario 19" 22U	1	434,90
Caja Rack 19" 4U	3	236,70
Regleta 19" 8 tomas	1	48,60
Total		3089,33

Tabla 4. Componentes básicos del cluster ACII.



Figura 3. El cluster ACII tras la sesión de montaje.

Componente	cant.	precio
Guías laterales telescópicas	3	91,11
Panel de parcheo 24x RJ-45	1	63,28
Herramienta Impacto	1	27,42
cable parcheo 1m	4	4,84
bolsa tornillería M6	1	26,26
Termostato Digital 1U	1	137,47
Regleta 19" 6 tomas	1	49,96
Pasacable 1U cepillo/peine	2	50,34
Anilla guiacable 1U horizontal	8	31,68
Anilla guiacable 1U vertical	8	31,68
Cable TwisTies 30m	1	2,96
Total		517,00

Tabla 5. Detalles de acabado del cluster ACII.

5. El software

Aunque en el curso 2008-09 diversos estudiantes escogieron (y obtuvieron puntuación por) trabajos escritos sobre instalación y configuración de determinados paquetes software, en la etapa de instalación surgieron incidencias difíciles de reconducir. ¿Cómo convencer a un estudiante de

que, si ha obtenido nota por un trabajo de instalación, él es el responsable de que funcione?



Figura 4. El cluster ACII con detalles de acabado.

Como ya se ha indicado previamente, para distinguir si esta situación se debía a un exceso de tarea o a una carencia formativa (y aprovechando que el cluster ya estaba montado), al año siguiente la actividad se simplificó, reduciéndola a la instalación, configuración, evaluación y explotación del cluster.

Para evitar la situación de que un solo estudiante pudiera bloquear el proceso de instalación, se propuso repartir el trabajo de forma que al menos dos estudiantes tuvieran base suficiente en cada tema, respondiendo los estudiantes que preferían organizarse entre ellos, comprometiéndose a que el cluster estuviera operativo en la fecha indicada.

Para evitar la situación del estudiante que hace un trabajo escrito sin sentirse comprometido a que el proceso de instalación y configuración descrito deba funcionar, o a que se deban resolver los problemas que surjan más tarde relacionados con su uso, se propuso que todo el trabajo fuera presencial, y que se fuera generando un diario o

bitácora, para disponer de una referencia que permitiera reproducir todo el trabajo realizado.

En el curso 2009-2010 los estudiantes que escogieron el proyecto realizaron la instalación del Sistema Operativo Debian con servidor DHCP y firewall IPTables, clonado de nodos con CloneZilla, paso de mensajes con Open-MPI, sistema de colas Torque, monitorización Ganglia y autenticación LDAP. No dio tiempo a probar ningún benchmark, debido fundamentalmente a la cantidad de sesiones dedicadas a instalar Torque y LDAP.

Los profesores indicaron previamente que podían aportar su experiencia con otros Sistemas Operativos, aunque si los estudiantes estaban dispuestos a organizarse entre ellos, repartirse el trabajo equitativamente, generar el diario requerido y comprometerse a tener el cluster operativo en los plazos indicados, no se les obligaría a rechazar la opción Debian.

El cluster estuvo operativo en la fecha indicada sin que se produjeran incidencias durante las sesiones presenciales. Paradójicamente, la encuesta anónima realizada posteriormente reveló disconformidades con el desarrollo del proyecto que permanecieron ocultas durante todo el cuatrimestre. Concretamente, se descubrió que los estudiantes hubieran preferido menos libertad.

6. Encuestas de opinión

En ambos cursos, 2008-09 y 2009-10, se invitó a los estudiantes a rellenar un cuestionario anónimo destinado a evaluar en qué medida se habían alcanzado los objetivos del proyecto, y si se debía mantener el próximo curso, con o sin modificaciones, o se debería eliminar. Estos objetivos del cuestionario se les comunicaban explícitamente a los estudiantes en el propio formulario. Las preguntas del cuestionario se ofrecen en la siguiente Tabla.

Pregunta	formato
¿Es el primer año cursando la asignatura?	s/n
El proceso de diseño del cluster le ha permitido aprender	-2...2
Considera que se debe mantener este trabajo en la asignatura	-2...2
Justifique 2 anteriores respuestas	libre
¿Qué es lo que más le ha gustado?	libre
¿Por qué?	
¿Qué cambiaría?	libre

En la forma de calificar al estudiante ¿qué cambiaría? ¿qué conservaría?	libre
Cualquier otro comentario que desee hacer	libre

Tabla 6. Cuestionario anónimo del Proyecto Cluster.

De los 12 estudiantes que entregaron la encuesta en el curso 2008-09, casi la mitad eran matriculados de primera vez (5 de 12), frente al 100% (6 de 6) del curso 2009-10. En la Tabla 7 se resumen las respuestas más frecuentes, junto con su frecuencia. En general, los estudiantes opinan que la actividad les ha permitido aprender y se debe mantener.

En atención a las opiniones 2008-09, en 2009-10 se redujo drásticamente la cantidad de trabajo a realizar (a la parte software exclusivamente), se

bajaron los umbrales mínimos, no se prorrateó el examen final, y se replanteó el proyecto como unas Prácticas alternativas (en sustitución de las “normales”). Los 6 estudiantes que entregaron el cuestionario anónimo opinaron mayoritariamente que debería darse más nota, siendo la puntuación injusta en comparación con las otras Prácticas.

Retrospectivamente, no se debería haber dado tanta importancia a las opiniones positivas en 2008-09 sobre la *libertad para escoger tarea*, y se debería haber intervenido frecuentemente durante las sesiones (sin darle tanta importancia al *ambiente distendido*) para recapitular sobre los hitos alcanzados y los que quedaban por cumplir.

	2008-2009					2009-2010				
Universo	12					6				
Primer año	41.7% (5 de 1 ^{er} año / 7 repetidores)					100% (6 de 1 ^{er} año)				
Aprender	1.42	(-2:1	-1:0	0:0	1:3 2:8)	1.50	(-2:0	-1:0	0:0	1:3 2:3)
Mantener	1.33	(-2:1	-1:0	0:0	1:4 2:7)	1.33	(-2:0	-1:0	0:0	1:4 2:2)
	frec.	respuesta				frec.	respuesta			
justificación	5x	aplicar teoría				2x	trabajo satisfactorio, interesante			
	4x	temas no cubiertos en otras asignaturas, motiva, interesa, es voluntario				2x	trabajo diferente, no cubierto en otras asignaturas			
	3x	aprender mediante la práctica, trabajar en grupo				2x	se aprende de los compañeros			
gustado más	6x	el clúster ha sido real foro (por su nivel, formato, ambiente)				3x	desarrollar capacidades profesión útil para un futuro empleo			
	3x	los grupos aprender sobre componentes				2x	ver el cluster funcionando ir resolviendo problemas instal. / config.			
	2x	libertad para todo (escoger tarea, proponer componentes)				2x	entorno distendido			
qué cambiaría	6x	que haya más participación mucho trabajo para tan poca nota				5x	organizar mejor: falta experiencia, material apoyo, difícil aprender que sea más guiado, menos libertad			
	4x	que quede más claro al principio los objetivos, puntuación, lista tareas, y que no es competitivo				2x	grupos más pequeños: 7→5 máx. sesiones todas en Laboratorio no en tutorías (probl. espacio)			
						2x	instalar Infiniband			
forma calificar	4x	en general, correcta				4x	más nota, injusto respecto otra Práctica			
	2x	se debe conocer la nota sobre la marcha, tras la aportación				4x	puntuar no según resultados, sino según esfuerzo / aprendizaje			
	2x	bajar más los umbrales, o quitarlos								
comentario	3x	estoy satisfecho					satisfecho con nuestro trabajo, he aprendido bastantes cosas			
	2x	espero que otros estudiantes puedan beneficiarse también					agradecer dedicación profesor y flexibilidad desarrollo trabajo			
	2x	se debe conocer la nota sobre la marcha, tras la aportación					hay que ser más realista, no da tiempo, montar 2 SO es de locos, o menos objetivos o más nota			
							obligar: Debian, Torque...			
							tanta libertad da quebraderos cabeza, obligando ahorraríamos tiempo			

Tabla 7. Resultados de la encuesta anónima.

Las calificaciones obtenidas por los estudiantes se resumen en la Tabla 8, para mostrar

que no se trata de un caso de calificación cicatera por parte del profesorado. Cubrir las competencias

de este proyecto en una asignatura con temario propio resolvería muchas de estas objeciones (participación, puntuación, organización, etc.).

curso	Notas					media
2008-09	3,0	3,0	3,0	3,0	3,0	2,41
	2,9	2,6	2,5	2,4	2,4	
	1,9	1,8	1,8	1,4	1,4	
2009-10	3,0	2,9	2,8	2,7	2,6	2,71
	2,5	2,5				

Tabla 8. Calificaciones del Proyecto Cluster.

Por tener una cierta idea de la popularidad que pudiera alcanzar la web del Proyecto Cluster, se le añadió en su día (Julio 2009) el *tracker* para las analíticas Google. El número de visitas que alcanzó en pocos meses nos hizo considerar la posibilidad de traducirla a inglés (Sept. 2009).

Ambas versiones de la web muestran distinta estacionalidad, como se puede comprobar en sus gráficas de visitas (Figura 5, Figura 6). La versión española ha llegado a acercarse recientemente a 50 visitas semanales, mientras que la inglesa llegó a 40 el pasado 2º cuatrimestre, y no se consulta tanto en los primeros cuatrimestres (15-20 visitas semanales, ver Figura 6 a izquierda y derecha del tramo con más visitas, 2º cuatrimestre 2010).



Figura 5. Informe Google Analytics de la web del Proyecto Cluster [6] (español).



Figura 6. Informe Google Analytics de la web del Proyecto Cluster [6] (inglés). El tramo de más visitas es el 2º cuatrimestre de 2010.

7. Posibles variaciones del proyecto

En vista a futuras variaciones del proyecto, se puede argumentar (mediante las encuestas

recolectadas) que los estudiantes desean una planificación rigurosa con hitos marcados en donde se obtenga puntuación que se conozca sobre la marcha, sobre un plan de trabajo previamente fijado por los profesores. Sólo un pequeño porcentaje de estudiantes prefiere un trabajo con más libertad, frente a la mayoría que preferiría cambiar la organización de la experiencia (5 de 6 en 2010). La paradójica poca satisfacción con la nota obtenida (sólo 4 de 12 en 2009, 2 de 6 en 2010, siempre 66% insatisfecho) indica que estos temas debieran cubrirse en una asignatura con temario (y calificación) propios.

La variación que quedaría por intentar sería realizar unos diarios detallados de instalación, temporizados por el profesor, para eliminar radicalmente el argumento de que es demasiado trabajo para tan poca nota. Los estudiantes que demostraran haberlos leído previamente (por ejemplo, respondiendo preguntas verbales o un test), tendrían derecho a intentar realizar la instalación y configuración descrita en ellos, en las fechas indicadas en la temporización. Se podría ofrecer la posibilidad de abandonar el proyecto durante el curso, si consideran que la nota que van obteniendo no es suficiente, y otra práctica alternativa les resultara más interesante.

8. Conclusiones

Durante los cursos 2008-09 y 2009-10 hemos conseguido poner en contacto a un total de 22 estudiantes con el tipo de supercomputador más habitual actualmente, el cluster de computadores, salvando las diferencias en cuanto a cantidad y categoría de los componentes (nodos sobremesa en lugar de servidor, prestaciones y tamaño del switch, etc.).

Las actividades realizadas han ido desde el propio diseño y montaje hasta la instalación y configuración del software de sistema y middleware asociado. Los estudiantes consideran en grado bastante alto que la actividad les ha servido de aprendizaje y se debe mantener.

La realización de este proyecto en estas fechas ha sido oportuna, ya que nos ha permitido detectar unas competencias que no se están cubriendo en nuestros actuales planes de estudio y que merecen estar presentes en los nuevos Grados, llevándonos a proponer dos nuevas asignaturas que podrían incorporar este tipo de prácticas, “*Ingeniería de*

Servidores” y “*Centro de Procesamiento de Datos*”. La primera asignatura, obligatoria de rama a impartir en 3^{er} curso, se va a dedicar al diseño, montaje, instalación y evaluación de un servidor de gama baja, y la segunda, obligatoria de la especialidad *Ingeniería de Computadores*, al diseño y evaluación de un servidor de gama media/alta. Las competencias específicas del Grado de Informática cubiertas por esta experiencia son (CVE: BOE-A-2009-12977):

- E1: Capacidad para concebir, redactar, organizar, planificar, desarrollar y firmar proyectos en el ámbito de la ingeniería en informática que tengan por objeto, de acuerdo con los conocimientos adquiridos, la concepción, el desarrollo o la explotación de sistemas, servicios y aplicaciones informáticas.
- E4: Capacidad para definir, evaluar y seleccionar plataformas hardware y software para el desarrollo y la ejecución de sistemas, servicios y aplicaciones informáticas.
- E6: Capacidad para concebir y desarrollar sistemas o arquitecturas informáticas centralizadas o distribuidas integrando hardware, software y redes.

Estas competencias son cubiertas por (entre otras) las asignaturas mencionadas anteriormente.

Los estudiantes nos han revelado lo que sucede si se intentan cubrir de manera transversal estas competencias: les supone “mucho trabajo para tan poca nota”, les causa insatisfacción si se proponen de forma voluntaria (debido a la poca participación y al agravio comparativo con la otra forma de puntuación), y les crea un conflicto de intereses, ya que si se les ofrece la posibilidad de organizar su trabajo (para evitar que consideren como imposición un SO que no es el preferido por ellos) concluirán en la encuesta final que el trabajo debe organizarse mejor, dada la falta de experiencia, e incluir material de apoyo y referencia. No habiéndose indicado explícitamente en la encuesta que echen de menos la lección magistral, interpretamos que desean los diarios detallados de instalación mencionados en la sección anterior. La experiencia adquirida nos servirá de referencia en el diseño de las prácticas de las nuevas asignaturas.

Agradecimientos

Agradecemos la ayuda del Plan de Innovación Docente de la UGR, que nos ha permitido diseñar un cluster que podremos reutilizar en cursos futuros para seguir intentando motivar a los estudiantes, así como la ayuda del Plan de Apoyo a la Docencia Práctica, que nos ha permitido adquirir la infraestructura Infiniband añadida posteriormente a dicho cluster. Agradecemos también la colaboración del proyecto de investigación TEC2010-15396 CITYC.

Agradecemos especialmente a los estudiantes Pablo Orantes, Ignacio Robles, Luis Quesada y Manuel Martín (2008-09), y Juan Pablo China y Rubén Ramos (2009-10) su valiosa colaboración y encomiable interés por el buen término del proyecto.

Referencias

- [1] Cañas, A., Calandria, D.J., Ortigosa, E.M. et al. *SWAD: Web System for Education Support*. In: Fernández-Manjón et al (Eds.): *Computers and Education: E-learning - from Theory to Practice*, 241 pages, Chapter 12, pp. 133-142, ISBN 978-1-4020-4913-2, Springer, (2007). Disponible online en <http://www.springer.com/computer/general/book/978-1-4020-4913-2>.
- [2] Cañas, A., Díaz, A.F., Prieto, A. *Sistema de servicios web de apoyo a la docencia y gestión de una asignatura*. In: *Actas de las VIII Jornadas de Enseñanza Universitaria de la Informática (JENUI'2002)*, pp. 611-614, Cáceres, 10-12 Julio 2002. Disponible online en http://bioinfo.uib.es/~joemiro/aenui/proclenui/Jen2002/Cac635_638.pdf.
- [3] Cañas, A., Ortigosa, E.M., Fernández, F.J., Anguita, M., et al. *SWAD (Sistema Web de Apoyo a la Docencia)*. In: *Actas del 6º Simposio Internacional de Informática Educativa (SIIE'04)*, Cáceres, 16-18 Noviembre 2004.
- [4] China, J.P., Ramos, R. et al (estudiantes ACII 2009-10). *Informe ACII: Puesta en marcha y configuración de un cluster con Debian 5.0*. Documento GoogleDoc disponible en <https://docs.google.com/Doc?docid=0Aane43447qy1ZGRxMmp6MjdfNWRwMjNkcWZr>.
- [5] Comellas, F., González-Cinca, R., Santamaría, E. *Simulación: Un curso innovador en los estudios de Aeronáutica*. In:

- ReVisión, vol.2, no.2, AENUI, 2009. Disponible online en <http://www.aenui.net/ojs/index.php?journal=revision&page=article&op=view&path%5B%5D=24>.
- [6] Web del Proyecto Cluster, con su enunciado: http://atc.ugr.es/~javier/docencia/Proyecto_Cluster.html. Versión inglesa cambia *Proyecto* por *Project*.
- [7] Fernández, F.J., Anguita, M. *Memoria descriptiva del Proyecto Clúster*. http://serin.ugr.es/unidad_innovacion_docente/memorias/08-08.doc. Buscador de proyectos también disponible online en la web: Convocatoria de Proyectos de Innovación Docente 2008: <http://innovacion.docente.ugr.es/pages/convocatoria-2008>.
- [8] Railsback, J. *Project-Based instruction: Creating excitement for learning*. Northwest Regional Educational Laboratory, “By Request” series, 2002. Disponible online en http://educationnorthwest.org/webfm_send/460, adaptado a español en Eduteka <http://www.eduteka.org/AprendizajePorProyectos.php>.
- [9] Valero García, M., Navarro, J.J. *FAQ sobre la adaptación de asignaturas al EEES: docencia centrada en el aprendizaje del estudiante*. In: ReVisión, vol.1, no.2, AENUI, 2008. <http://www.aenui.net/ojs/index.php?journal=revision&page=article&op=view&path%5B%5D=8>. Consultar en concreto el penúltimo párrafo en la sección §3.10.

Simbiosis entre materias troncales y transversales: Análisis de una experiencia

Miguel Garre, Antonio Navidad, Rosalía Peña
Departamentos Varios
Departamento de Ciencias de la Computación Universidad de Alcalá de Henares
{miguel.garre,antonio.navidad,rpr}@uah.es

Resumen

El primer acercamiento a la resolución de problemas supone una gran dificultad para el alumno. En nuestra universidad, este acercamiento se realiza en la asignatura de Fundamentos de la Programación (FP) en las ingenierías técnicas informáticas. La asignatura presenta un alto grado de abandono y bajo rendimiento. De otra parte, el entrenamiento en habilidades de trabajo en equipo es una de las competencias recogidas expresamente en las recomendaciones para los nuevos planes de estudio. En los últimos años, los profesores de FP, han ido realizando diferentes aproximaciones al trabajo en equipo, pensando que la colaboración entre iguales facilitaría el aprendizaje. Estas aproximaciones condujeron al profesorado a asumir formalmente el entrenamiento de los alumnos en las habilidades requeridas para trabajo en equipo. La materia convencional resulta un hilo conductor para el entrenamiento de la competencia. El equipo puede fomentar el trabajo continuado del alumno, lo que minimiza el número de abandonos y facilita un aprendizaje más crítico de la programación, incentivando en el discente la adopción de criterios metodológicos conducentes al desarrollo de código de calidad. La simbiosis de estas dos materias permite ejercitar actitudes colaborativas. La colaboración permite abordar problemas de mayor envergadura, proporcionando una visión más realista del modo en que se desarrollará el futuro ejercicio profesional del alumno.

1. Motivación

Tanto en las Jornadas de Enseñanza Universitaria de la Informática (JENU) [4,5,6,8,9,10,11,13,14], como en otros foros [1,3,7,12,15] se ha puesto de relieve la gran dificultad que encuentran los alumnos en la

primera asignatura de programación en las ingenierías informáticas. Realmente esta dificultad corresponde a un problema más general, detectado en la primera materia del currículo que implica resolución de problemas en otras titulaciones [3]. En estas asignaturas se producen unas altas tasas de abandono y también de fracaso entre los presentados. Resulta especialmente sorprendente este resultado cuando es una materia próxima a lo que pudiéramos suponer constituyen las expectativas profesionales del alumnado. Muchos autores han estudiado los factores que podrían condicionar o causar este efecto. Según Diane [7] la madurez cognitiva correlaciona menos con el éxito en la resolución de problemas, que la importancia relativa que el alumno otorga a los conocimientos teóricos y a algunos factores de su personalidad, como son la constancia y la resistencia a la frustración. Zúñiga [14] y Peña [9] coinciden en que la nota media de los cursos anteriores es el mejor predictor del rendimiento académico en esta materia. Posiblemente esto sea debido a que la nota media es función directa del hábito de trabajo del alumno. Raadt [12] relaciona el éxito, en estas materias, con el estilo de aprendizaje (profundo o superficial), descartando como relevantes la experiencia previa y capacidades innatas tales como el coeficiente intelectual, aunque reconoce que el estilo de aprendizaje está condicionado por factores de entorno y por tanto es una característica modificable. Bennedsen [1] no encuentra relación entre el desarrollo cognitivo y la habilidad de programación, ni con las calificaciones de las asignaturas de matemáticas; pero sí encuentra correlación entre las calificaciones de matemáticas y las de programación. Entendemos que su hallazgo es debido al tipo de ejercicios que propone, pues otros autores [9,14] han descartado este tipo de correlación. Willson [15] evalúa la influencia de

otros factores. La resolución de problemas comienza por la comprensión de las necesidades del cliente. En este sentido, Peña [10] intenta mejorar la capacidad lectora para facilitar la resolución. Rodríguez [13] resalta la importancia de una rápida retroalimentación al alumno sobre su desempeño. Pese a la atención que ha recibido esta problemática, no queda clara la causa, posiblemente, porque son muchos los factores involucrados. La figura 1 resume posibles causas del abandono y bajo rendimiento encontrado en estas materias.

El trabajo describe una experiencia, llevada a cabo en la universidad XX, en la que se coordina la asignatura de FP con una materia de libre configuración, de dos créditos, que bajo el título de “Habilidades de trabajo en equipo” (HTE) se imparte, en exclusiva, para los alumnos matriculados en la de programación (FP). Solapar ambos contenidos provoca una simbiosis. De una parte, los ejercicios de programación proporcionan situaciones adecuadas para ejercitar las habilidades de trabajo en equipo. Cuando se ha intentado ejercitar con ejercicios típicos de formación de equipos, los alumnos lo perciben como juegos triviales y no lo integran con su trabajo. Complementariamente, el equipo es enriquecedor para la resolución de problemas, expone a los alumnos a una mayor variedad de enfoques, entre los que deben evaluar ventajas e inconvenientes, fomentando un espíritu crítico en sus conocimientos. El apoyo del equipo permite la realización de ejercicios más ambiciosos y realistas conforme avanza el curso y se consolidan los equipos humanos. Adicionalmente, el alumnado encuentra motivador ganar maestría en técnicas de trabajo en equipo.

En el presente documento se recoge, en primer lugar, la evolución de los intentos, de los responsables de la primera materia de programación para mejorar el rendimiento académico. Estos intentos concluyeron en la formalización, en el plan de estudios, de la asignatura de libre configuración. En el resto del documento, se describe la metodología empleada, se comentan sucintamente los contenidos de la materia trasversal, y se analizan los resultados obtenidos en el segundo año de la implantación de esta metodología.

2. Evolución del enfoque

Habiendo constatado las dificultades del alumnado para superar la primera materia de programación, los profesores intuimos que:

- fijar unas horas específicas para trabajo personal fomentaría la continuidad, con lo que previsiblemente, disminuiría el índice de abandono
- afrontar un enunciado en compañía de otros compañeros, aumentaría la resistencia a la frustración
- exponer sus planteamientos a sus compañeros podría proporcionar al alumno una mayor profundidad en sus conocimientos
- tratar de entender y de mejorar los planteamientos de otros, ampliaría la perspectiva, favorecería el espíritu crítico, y simultáneamente haría patente la conveniencia de adscribirse a una metodología para el desarrollo de programas.
- La importancia de favorecer el trabajo continuado ha sido corroborada estadísticamente. Por ejemplo, un experimento de Petit [11] proporciona una amplia y precisa información de la distribución temporal del esfuerzo dedicado a la materia por cada alumno de un primer curso de programación. La confrontación del rendimiento académico respecto a la continuidad en el trabajo confirma que estos ítems están ampliamente relacionados. Los otros tres aspectos deseados los proporciona el trabajo en equipo. Los beneficios del aprendizaje entre compañeros también están bien documentados [2, por citar un ejemplo].
- Motivados por estas intuiciones, en el curso 2004-2005 se les propuso afrontar la resolución de ejercicios de programación de forma conjunta en grupos preestablecidos. Los grupos estaban dimensionados entre cuatro y seis alumnos, que se habían adherido voluntariamente al proyecto. En las horas asignadas a tutorías se reunían el profesor y un representante de cada grupo que exponía y defendía la solución aportada por su grupo, componiendo, entre todos, una

versión mejorada, que cada coordinador trasladaba a su grupo.

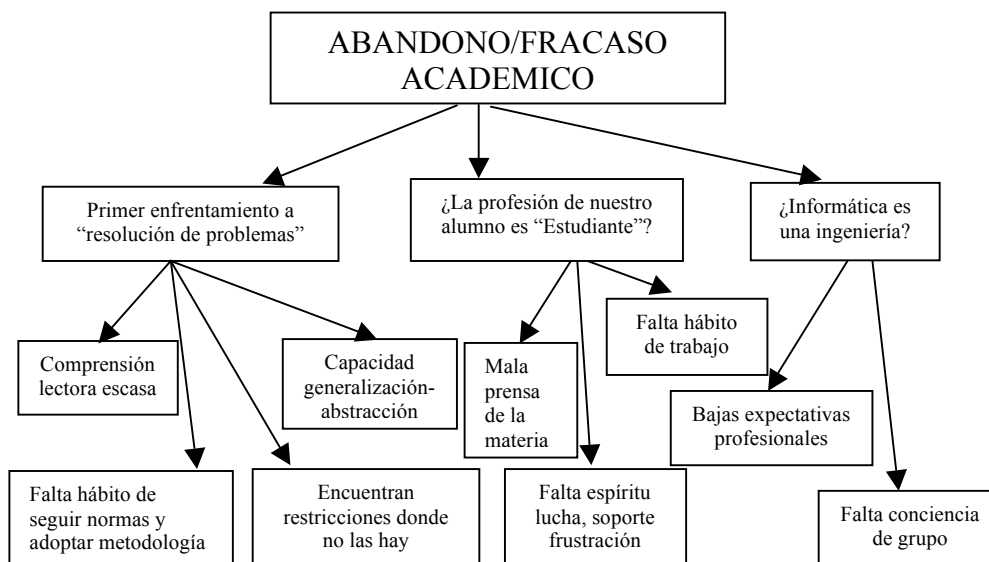


Figura 1. Posibles causas de las altas tasas de abandono y fracaso académico en Programación

Al no disponer de aulas adecuadas para el trabajo en grupo, las reuniones, tanto de cada grupo como de los coordinadores con el profesor, se organizaban en sitios públicos, de tránsito hacia las aulas. Esto es incómodo, pero aporta un beneficio inesperado:

- propicia espíritu de trabajo en los compañeros, incluso en los no participantes en el experimento.

Las primeras experiencias sugerían que mantener fijos los componentes de cada grupo, a lo largo del cuatrimestre, propicia el establecimiento de un compromiso entre los miembros, que repercute en la disminución de abandonos. Más de un 80% de los alumnos participantes se presentaban a examen, mientras que solo un 30% de los restantes lo hacía. El examen era común para alumnos participantes y no participantes en el proyecto contestándose de forma individual. En esta cuarta prueba, la calificación media del colectivo de participantes fue 1,5 puntos superior a la de los no participantes y 2,5 en el caso de los coordinadores de grupo.

En el curso 2006-2007 los alumnos pidieron abrir la sesión de discusión entre grupos a todos los componentes. De otra parte, con el fin de aumentar su conciencia de compromiso con su equipo y con el profesor se estableció un contrato, que el alumno debía de firmar a principio de curso. El alumno obtenía el derecho a participar en el proyecto siempre que se comprometiera a llevar la materia al día, a asistir a las sesiones de trabajo acordadas con el profesor y a las del equipo. Además, tenía que involucrarse en el éxito de cada uno de los componentes de su equipo.

También en la línea de reforzar los vínculos se propuso hacer la evaluación conjunta del grupo, pero esta idea fue rechazada por los alumnos, por lo que se abandonó ese año y los sucesivos.

La evaluación de los cursos 2005-2006 y 2006-2007 proporcionó nuevas reflexiones. La composición de los grupos se mantuvo a lo largo del cuatrimestre, excepto la fusión de dos grupos, por haberse producido bajas a mitad de curso. Prácticamente todos los alumnos que permanecían tras la semana 9 (sobre 14) se presentaron a examen, pero mientras que en

unos grupos había un rendimiento notable y se apreciaban similitudes entre las soluciones individuales propuestas por los componentes (descomposición funcional, elección de identificadores, etc.), en otros grupos no había mejora significativa en la evaluación, respecto a los alumnos que no habían participado en el proyecto y no se apreciaba influencia entre los componentes. Es decir, unos grupos habían funcionado y otros no. Teníamos que enseñarles a evolucionar desde el concepto de grupo al concepto de equipo de trabajo, pero necesitábamos aprender nosotros a fomentar esta transición y se pidió ayuda al gabinete de orientación psicopedagógica de la universidad. Un profesional del gabinete supervisaba esporádicamente el funcionamiento de los grupos y apreció la conveniencia de propiciar una formación específica en habilidades de trabajo en equipo en los discentes.

El curso académico 2007-2008, la Escuela aprobó la creación de una materia de libre configuración que se impartió también en el curso 2008-2009, y cuyo contenido, metodología y resultados se analizan en este trabajo.

3. Contenido

Ya que es infrecuente la formalización de la docencia de Habilidades de trabajo en equipo en nuestras titulaciones y conlleva gran variedad de aspectos, nos permitimos especificar los temas que hemos entresacado como esenciales para este primer acercamiento:

- Importancia del trabajo en equipo
 - Grupo frente a Equipo. Características y diferencias: Sinergia
 - El papel de la comunicación: Interferencia/retroalimentación
 - Organización de las reuniones
 - El papel de la comunicación no verbal
- Inteligencia emocional: Composición y aplicación en el equipo.
 - Autoconocimiento
 - Autorregulación
 - Motivación
 - Empatía
 - Habilidades sociales
- Fases en la evolución del equipo

- Formación
- Conflicto
- Estructuración
- Desarrollo
- Clausura
- Equipos eficaces frente a equipos disfuncionales
 - Tamaño del equipo y eficiencia
 - Redes de comunicación
 - El proceso de toma de decisiones
 - La negociación
 - Resolución de conflictos en las relaciones humanas
 - Cohesión
- Técnicas de trabajo en equipo
 - Variables que determinan las diferentes técnicas.

Nada especial que aportar en cuanto al contenido de la materia de FP.

4. Metodología

Precisamos de dos semanas para estructurar los equipos: recoger los contratos firmados, acordar con ellos los horarios, etc. de modo que contamos con 12 sesiones de trabajo de 2 horas semanales.

En el primer año de impartición de asignatura de libre elección el psicopedagogo preparó para cada sesión una breve exposición de los contenidos descritos y trajo juegos para ejercitar lo comentado. Los alumnos evaluaron esas sesiones como entretenidas, interesantes, pero encontraban dificultad en relacionar los juegos con su trabajo. Sin embargo, a lo largo de una sesión de trabajo en programación, surgen situaciones que propician hablar de sinergia, comunicación no verbal o resolución de conflictos. Por lo que al año siguiente, se eliminaron las presentaciones y se aprovecharon las oportunidades, surgidas de forma espontánea en el colectivo, o expresamente provocadas por el monitor, para integrar estos temas. Esto implica, que quizá alguno de los aspectos descritos en contenidos no quede cubierto en un curso concreto, pero sin embargo, lo que se aborda resulta mejor integrado.

Tres de las doce sesiones (6 horas) se dedicaron a las pruebas de evaluación continua. Adicionalmente, a fin de dar eficientemente una

pronta retroalimentación al alumno sobre su propio desempeño, tres horas lectivas de la asignatura FP se dedicaron a resolver las pruebas de evaluación continua, disponiendo cada alumno de su propio ejercicio corregido. La cuarta prueba se realizó el día del examen final.

En la primera sesión se les da un problema a resolver, se les pide que se presenten a los miembros de su equipo, y organicen lugar y horario de sus sesiones de trabajo (con frecuencia no lo hacen y queda patente la necesidad en la sesión siguiente).

En las ocho sesiones restantes resuelven proyectos de programación de los que disponen las especificaciones por adelantado y el grupo trae a la reunión su “mejor propuesta”. Mientras que los primeros proyectos los resuelven todos los miembros en conjunto; en los dos últimos proyectos abordan el enunciado, realizan la descomposición funcional y acuerdan las cabeceras de los subprogramas de forma conjunta, pero se reparten el trabajo de codificación de las implementaciones para realizarlo de forma individual en casa. En la sesión siguiente lo integran y prueban.

El monitor observa el comportamiento de los grupos; detecta desacoples. Resume al colectivo general los problemas detectados en la actividad. Unas cinco horas de las veinticuatro disponibles son ocupadas para comentar los aspectos de equipo que hemos descrito. Desde la segunda edición, esta tarea es asumida por el profesor de la materia técnica.

5. Evaluación del proceso pedagógico

Cada ejercicio académico se ha realizado una evaluación interna cualitativa y una externa cuantitativa. La valoración interna se implementa mediante un cuestionario orientado a medir el grado de satisfacción del alumnado y del profesorado implicado, así como el grado de acogida del proyecto a través de los años. Para la evaluación externa se tienen en cuenta los resultados académicos obtenidos por los alumnos en la asignatura de FP.

Fruto del análisis de ambas evaluaciones ha sido la evolución del proyecto, que se ha descrito anteriormente. Cabe destacar que la Delegación de alumnos, como organismo, se ha

involucrado activamente en la continuidad del proyecto, agradeciendo al profesorado año tras año la iniciativa y solicitando a la Escuela la oficialización de la materia transversal. Con la puesta en funcionamiento de los planes de estudio de Grado, y otras circunstancias de profesorado, el presente curso no se ha podido ofertar la asignatura de libre configuración.

En las ediciones anteriores se ofrecía, como aliciente por participación en el trabajo en grupo, un punto extra en la nota final de FP, siempre y cuando superaran el examen. La pequeña dimensión del aliciente, permite afirmar que no enmascara significativamente la tendencia que muestra el estudio del rendimiento académico. Desde que se ha formalizado la docencia de la materia transversal, se les ofrece un certificado que pueden convalidar por dos créditos de libre configuración. Llama nuestra atención que en la segunda edición ningún alumno solicitó el certificado, lo que queremos interpretar como una prueba del interés intrínseco de la actividad.

El número de alumnos participantes ha sido todos los años entre treinta o cuarenta en cada turno. El número de alumnos matriculados ha ido en descenso, mas por la paulatina salida del alto porcentaje de repetidores que por un descenso real en el nuevo ingreso. Globalmente, la participación porcentual ha aumentado, pero sigue manteniéndose por debajo de los valores deseados.

Para favorecer que se adhirieran al proyecto, en el curso 2008-2009 se ofreció a los participantes una evaluación continua a través de cuatro pruebas, en la que la última era precisamente el examen oficial de final de curso. La calificación final acordada fue:

$$\text{Max } (0,1 * p1 + 0,1 * p2 + 0,2 * p3 + 0,6 * p4, p4)$$

teniendo que sacar un mínimo de 3 sobre 10, en cada prueba, para tener derecho a continuar en el proyecto. En caso contrario, su calificación en la convocatoria ordinaria será la obtenida en p4. Esta mecánica exigió sistematizar la captura de datos que permite ahora evaluar el proyecto.

En la tabla 1 se presentan datos correspondientes al ejercicio 2008-2009 para los turnos de mañana (M) y tarde (T)¹.

Sólo 90 de los 161 alumnos matriculados firmaron el contrato, e incluso 13 de los que lo firmaron no asistieron siquiera a la primera de las reuniones.

Entendemos que la única manera de comparar objetivamente la adquisición de conocimientos en dos colectivos es someterlos a idéntica prueba (p4) en nuestra experiencia. Resaltamos que ningún alumno superó el curso con una calificación inferior a un cuatro en esta, lo que destierra los temores de que la evaluación continua pudiera ser menos significativa de los conocimientos adquiridos por el alumno que un examen final.

De los alumnos presentados a la convocatoria ordinaria superan la materia un 74% de los participantes en el proyecto (tasa de éxito), frente a un 21% entre los no participantes. En la convocatoria de septiembre la superan 41% frente a 16% respectivamente. Es decir, aproximadamente un rendimiento tres veces mayor entre los participantes en ambas convocatorias. Los porcentajes de aprobados respecto a matriculados (tasa de rendimiento) presentan una tendencia similar (42 frente a 12 y 24 frente a 16, respectivamente).

Por tanto, comparando los colectivos con independencia del turno al que asistieran, tanto la tasa de rendimiento, como la tasa de éxito es significativamente mayor entre los alumnos que participaron en el proyecto que entre los no participantes.

Desafortunadamente, pese a la sensible mejora en el rendimiento académico, las tasas de abandono y de fracaso siguen siendo preocupantes por lo que hay que seguir intentando encontrar fórmulas que faciliten el aprendizaje y aumenten la motivación,

disminuyendo el abandono y el fracaso. Es difícil encontrar el modo de llegar a todos los alumnos, ya que como se ha comentado hay muchos factores que pueden estar afectando, y además, son un colectivo poco homogéneo como lo demuestra el análisis comparativo de los datos de los alumnos de la mañana y la tarde.

Llama la atención la diferencia en la continuidad/abandono entre los dos turnos. La caída en el número de participantes es mucho mayor en la mañana que en la tarde, desde el momento inicial del proyecto. Hay 10 ($42-32=10$, $10/42*100=24\%$) alumnos, que pese a solicitar su participación, firmando el contrato, no llegaron a incorporarse al proyecto, esto supone un 24% de abandonos iniciales llamativo frente al 6% ($48-45=3$, $3/48*100=6\%$) que este fenómeno aparece en el grupo de tardes. El índice de abandono tras las sucesivas pruebas también es mayor en las mañanas que en las tardes, aunque la diferencia no es tan grande como la del abandono inicial ($(32-10)/32*100=69\%$ frente a $(45-28)/45*100=38\%$). El diferente comportamiento de los grupos puede estar influido por muchos factores, pero observamos que la edad media de los alumnos de la tarde es mucho mayor que la de los de la mañana. Muchos alumnos de la tarde están en el mercado laboral, lo cual es infrecuente en la mañana. Pensamos que la madurez, el hábito de trabajo y la mayor resistencia a la frustración son factores suficientes para justificar estas diferencias. Adicionalmente, la mayoría de alumnos de la tarde son repetidores

Posiblemente por los mismos motivos, sean los que acabamos de indicar u otros, el índice de éxito de los participantes en el proyecto en febrero es mayor en el turno de tarde que en el de mañana (49% frente a 31%). Cabe resaltar que en el turno de tarde, todos los alumnos que se presentaron a la primera prueba de la evaluación continuada (39) que no aprobaron en febrero (22) se presentaron también a la de septiembre (17). Esto conduce a pensar que el trabajo durante el curso les capacitó para afrontar en solitario la preparación de la materia durante el periodo estival.

Con la entrada de los nuevos planes de estudio se reduce el número de horas de contacto con el alumno en la primera asignatura de

¹ Donde:

- Número de alumnos en no_proyecto es el de matriculados menos el de participantes en el proyecto
- Participantes en septiembre es el número de alumnos en febrero menos superan materia en febrero
- No se proporcionan los datos parciales de los turnos para los alumnos no presentados porque oficialmente hay listas únicas.

programación, por lo que se ha reducido el temario.

En los cuestionarios de evaluación interna los estudiantes consideraban que el trabajo en equipo les había resultado útil para aprender a

programar. En esta ocasión la materia de libre configuración computó como docencia, por lo que no resultó una excesiva carga adicional para el profesorado.

	Febrero			Septiembre				Total		
	proyecto			no_proyecto	proyecto				no_proyecto	
Matriculados	M	T	M+T	M+T	M	T	M+T	M+T	161	
Firman contrato	42	48	90							
Número de alumnos	32	45	77	84	22	23	45	86		
Presentados a P1	26	39	65							
Permanencia (>= 3)	21	36	57							
Presentados P2	20	34	54							
Permanencia(>= 3)	17	29	46							
Presentados P3	17	28	45							
Permanencia(>= 3)	16	28	44							
Presentados P4	15	28	43	47	10	17	27	31		
Superan curso	10	22	32	10	3	8	11	5		58
Superan/presentados %	67	79	74	21	30	47	41	16		
Superan/matriculados %			42	12			24	6		36

Tabla 1 Estadísticas del curso académico 2008-2009

6. Conclusión

No solo es posible, sino que es conveniente que el profesorado técnico asuma el entrenamiento en competencias profesionales del alumnado. Asumir este entrenamiento no viene a mermar los conocimientos técnicos que adquirirá el alumno, si no que, por el contrario, refuerza su capacidad de aprendizaje. Concretamente, la integración del entrenamiento en habilidades de trabajo en grupo en la docencia del primer acercamiento a la programación propicia un aprendizaje más maduro, reflexivo y crítico, disminuye la aparición de vicios de programación, incentiva la adscripción a una metodología, habilitando al programador novel a realizar código de mayor calidad, al ponerse de relieve la importancia de propiedades que caracterizan la calidad del software, como son

reusabilidad, robustez, independencia de piezas de software y legibilidad, y mantenibilidad.

El apoyo de los compañeros, el compromiso establecido con ellos, disminuye el número de abandonos y aumenta las probabilidades de éxito en la materia de programación. Pese a ello, sigue siendo necesario encontrar nuevas vías de mejorar el rendimiento.

A lo largo de un cuatrimestre es posible crear la sinergia necesaria para afrontar una tarea distribuida entre los componentes de un equipo. Se les ha enfrentado también a diversas técnicas de trabajo en equipo como Braimstorming, Juego de roles y Puzzle. Han experimentado la resolución de conflictos, la necesidad del establecimiento de redes de comunicación y se han hecho conscientes del papel que juega la comunicación no verbal en las relaciones humanas.

Agradecimientos

A la Escuela por haber reconocido los créditos de la materia de Habilidades de trabajo en equipo. A la Dirección del Departamento por haberla contabilizado como carga docente del profesorado en el curso 2008-2009. Al Vicerrectorado de Profesorado (en la persona de) y al Gabinete de orientación psicopedagógica (en la persona de) por su apoyo.

Referencias

- [1] Bennedsen, J.; Caspersen, M. E. *Abstraction ability as an indicator of success for learning object-oriented programming?* ACM SIGCSE Bulletin Vol. 38 (2) pp. 39 – 43. June 2006
- [2] Boud, D.; Cohen, R.; Sampson; J. *Peer Learning and Assessment*. Assessment & Evaluation in Higher Education, Vol. 24(4), pp. 413 – 426. December 1999
- [3] Brookshire, R.; Palocsay, S. *Factors contributing the success of undergraduate Business Students in Management Science Courses*. Decision Sciences Journal of Innovative Education Vol.3(1) pp.99-108. 2005.
- [4] Casanova, A.; Garayoa, J.; Marques, F.; Vidal, V. *La relevancia de los documentos en Google: una actividad común de las asignaturas de Programación y Computación numérica*. XIII Jenui , pp. 263-268. 2007
- [5] Cernuda del Río, A. *Enseñanza por descubrimiento de los algoritmos de ordenación*. XI Jenui, pp. 214. 2005
- [6] Cernuda, A.; Hevia, S.; Suárez, M. C.; Gayo, D. *Un estudio sobre el absentismo y el abandono en asignaturas de programación*, XIII Jenui, pp. 487-494. 2007
- [7] Hall D. J.; Cegielski C. G.; Wade J.N. *Theoretical Value Belief, Cognitive Ability, and Personality as Predictors of Student Performance in Object-Oriented Programming Environments*. Decision Sciences Journal of Innovative Education, Vol. 4 (2), pp. 237. July 2006
- [8] Gómez Fernández, E.; García García, M.J.M.; Gurutze, M. *Aplicación de diversas metodologías activas en la asignatura de introducción a la programación*. XI Jenui, pp. 205-211. 2005
- [9] Peña, R.; Sánchez, I. *Análisis estadístico del rendimiento académico de una asignatura con relación a asignaturas anteriores*. XI Jenui, pp. 121. 2005
- [10] Peña, R.; Garre, M. *7 minutos para un objetivo transversal: LEA ATENTAMENTE*. XIII Jenui, pp. 545-546. 2007
- [11] Petit, J.; Roura, S. *Programación I: Una asignatura orientada a la resolución de problemas*. XV Jenui 2009. Barcelona, ISBN: 978-84-692-2758-9
- [12] Raadt M.; Hamilton, M. y Lister, R. *Approaches to learning in computer Programming Students and their effect in success*. Higher Education in a changing World. Herdsa, pp. 407-414. 2005
- [13] Rodríguez del Pino, J. C.; Díaz Roca, M. Hernández Figueroa, Z. González Domínguez, J. D. *Hacia la Evaluación Continua Automática de Prácticas de Programación*. XIII Jenui pp. 179-186. 2007
- [14] Zúnica, L.; Blesa, P. Alcover, R., Más, J., Valiente, J.M.; *Estudio del rendimiento académico de una asignatura con relación a asignaturas de cursos anteriores*. Jenui, pag 137. 2003
- [15] Wilson, B. C. *A Study of Factors Promoting Success in Computer Science Including Gender Differences*. Computer Science Education, Vol. 12(1 & 2) , pp. 141 – 164. March 2002

La evaluación personalizada como estrategia de motivación.

Vicente Arnau Llombart, Miguel Arevalillo-Herráez, José M. Claver

Departamento de Informática

Universidad de Valencia

Avda. Vicent Andrés Estellés, s/n, 46100 - Burjassot, Valencia

{vicente.arnau, miguel.arevalillo,jose.claver}@uv.es

Resumen

En el sistema educativo actual, es habitual que el estudiante concentre la mayor parte de su tiempo de estudio en las fechas cercanas a las pruebas de evaluación. Con el objetivo de fomentar el trabajo constante a lo largo del curso, exponemos una metodología docente basada en la evaluación continua y modelable por el alumno, que al mismo tiempo ayuda a evitar el plagio. Por un lado, los laboratorios se evalúan ‘in situ’ mediante una prueba que se realiza durante la sesión, en la que el alumno debe demostrar que ha preparado previamente la práctica y que ha realizado y entendido las actividades realizadas a lo largo de la sesión de laboratorio. Por el otro, en las clases de teoría y problemas, utilizamos trabajos voluntarios consistentes en problemas de dificultad variable y cuya evaluación puede, a elección del alumno, reducir de forma considerable el peso del examen final. Para fomentar el trabajo colaborativo, éstos pueden realizarse opcionalmente en grupo, aunque su corrección se realiza individualmente en horario de tutorías. Esta metodología ha sido utilizada exitosamente en dos asignaturas de la titulación de Ingeniería en Informática de la Universidad de Valencia. En el artículo se exponen los resultados académicos obtenidos, que respaldan su efectividad.

Summary

In the current Higher Education system, it is common that students concentrate most of their effort close to the examination dates. In this paper we present a teaching methodology that aims at encouraging continuous work. To this end, an adaptive evaluation model has been used. This model inherently contributes to avoid plagiarism. On the one hand, computer labs are evaluated “in place”, by means of a test that students much complete at the end of each session. In this test, the student has to demonstrate that both the pre-

lab work and the lab activities were appropriately done. On the other hand, voluntary course works are proposed during theoretical lectures. These consist of a collection of problems of variable difficulty. By handing their solution, students may considerably reduce the weight of the final exam. In addition, to encourage collaborative learning, these can be optionally solved in teams. Nevertheless, marking is always done individually, in tutorial hours. This methodology has been successfully applied in two modules of the Computer Science degree at the University of Valencia. This article presents the academic results that support its benefits.

1. Motivación

La carencia de un modelo formativo que integre las visiones académicas, profesional y vital [2] se ve reflejada en la falta de motivación académica de gran parte del alumnado que ocupa nuestras aulas. Actualmente, la mayor parte de nuestros alumnos de ingeniería se conforman con un aprendizaje superficial de los contenidos de las materias, siendo su principal objetivo conseguir aprobar, más que adquirir conocimientos que le puedan ser de utilidad en su futuro profesional.

Una de las principales causas de esta actitud es la falta de una cultura de esfuerzo continuo. Para paliar este problema, se hacen necesarias nuevas estrategias de enseñanza/aprendizaje que estimulen no solo el desarrollo intelectual, sino además el personal [6]. La aplicación de la metodología clásica de clases magistrales y un examen al final del curso como única forma de evaluar al estudiante se traduce habitualmente en bajos niveles de asistencia, malas calificaciones, y un alto número de suspensos. Por ello, se hace necesario el uso de estrategias docentes que potencien la implicación del alumno en su propio aprendizaje, en un intento por reforzar su responsabilidad, autoestima, interés y motivación [5]. Una forma de lograr estos objetivos es

mediante la recompensa inmediata de los esfuerzos realizados.

Tradicionalmente, para solucionar el problema del absentismo en los laboratorios, las prácticas se hacen de asistencia obligada. Además, esta técnica suele combinarse con una estrategia de evaluación consistente en solicitar una memoria del trabajo realizado, siendo entonces frecuentes los casos de plagio. Desafortunadamente, la realización de un examen para solucionar este último problema no suele alcanzar tampoco los resultados académicos deseados. En general, la asistencia obligatoria suele ser una técnica poco útil si no viene acompañada de otras estrategias de evaluación que fomenten la atención e interés del alumnado. Además, es importante que la propia evaluación sea de carácter formativo, se encuentre integrada dentro del proceso educativo y contribuya al desarrollo de capacidades [9].

Por otra parte, existe otro problema con alumnos que se incorporan al mundo laboral durante la realización de sus estudios, ya sea de forma continua o intermitente. Estos alumnos, generalmente, encuentran dificultades para poder someterse a una evaluación continua durante todo el curso. Por ello, suelen optar por la realización de un examen final como única prueba de evaluación, cuando la normativa de evaluación de la asignatura lo permite.

En este artículo proponemos la combinación de dos estrategias de evaluación enfocadas a garantizar el trabajo continuo y asegurar la autoría de las actividades, evitando el plagio. La primera de ellas tiene como objetivo mejorar el rendimiento del alumno en las sesiones de laboratorio, incentivando la preparación previa de las sesiones y consiguiendo que el alumno trabaje intensamente durante las mismas. La segunda propuesta pretende captar la atención de los alumnos en clase, recompensando el esfuerzo realizado al final de las sesiones teóricas. Para conseguir que además lleven la asignatura al día, les proponemos trabajos de mayor dificultad a realizar fuera del aula. Esto se combina con una flexibilización del peso del examen final, personalizándolo para cada estudiante.

El resto de este artículo se organiza de la siguiente forma. En la sección 2 describimos la metodología propuesta en detalle, presentando en primer lugar la metodología docente que venía siendo utilizada anteriormente. En la sección 3

describimos los resultados obtenidos, mostrando los beneficios que aporta el uso de nuestra estrategia docente. Finalmente, exponemos las conclusiones que se derivan de nuestro trabajo.

2. Metodología docente propuesta

En esta sección se presenta en detalle la metodología que hemos utilizado durante los 2 últimos años en la asignatura “Arquitectura e Ingeniería de los Computadores” (AIC), de cuarto curso; y durante los 7 últimos años en la asignatura “Ampliación de Estructura de Computadores” (AEC), de tercer curso, ambas pertenecientes a la Ingeniería Informática de la Universidad de Valencia.

Para facilitar la comprensión del contexto en el que se realizaron los cambios y los beneficios aportados por la metodología, comenzamos por exponer el método de enseñanza del que se venía haciendo uso antes de implantar el que se presenta. En primer lugar abordaremos la docencia y evaluación de las sesiones de laboratorio, y después la de las clases de teoría.

2.1. El laboratorio

Anteriormente, las sesiones de laboratorio se evaluaban a través de una memoria, que el alumno debía entregar en la siguiente sesión. La evaluación de estas memorias suponía el 50% de la nota de laboratorio, y un examen final de laboratorio computaba el 50% restante. Desafortunadamente, mientras que las memorias solían tener una calidad suficientemente alta, los exámenes ofrecían unas tasas de suspensos cercanas al 80%. Atribuimos este efecto a dos factores. Por un lado, al plagio, basándonos en la similitud existente entre las memorias entregadas. Por el otro, a una tendencia a descuidar la preparación del examen final por haber obtenido ya una alta calificación en las prácticas.

En la actualidad, no hay entrega de memorias, pero sí un trabajo previo fuera del aula que requiere una dedicación estimada de entre una a dos horas. Para facilitar la realización de este trabajo, el enunciado de la práctica siempre está disponible una semana antes de la sesión de laboratorio. Este enunciado incluye una breve explicación de los conceptos teóricos que serán utilizados y evaluados, junto con una completa

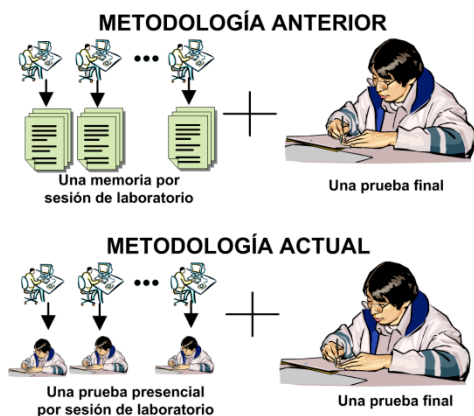


Figura 1. Evaluación de las sesiones de laboratorio. Proponemos un examen al final de cada sesión.

descripción de las actividades a realizar antes (trabajo previo) y durante la sesión. Además, al final de la sesión de laboratorio se realiza una prueba escrita con múltiples cuestiones para certificar la correcta comprensión de los objetivos planteados en la sesión de laboratorio y evaluar 'in situ' los conocimientos adquiridos. Estos exámenes se realizan por parejas y constituyen el 50% de la nota de prácticas. El 50% restante se evalúa mediante una prueba individual de aproximadamente una hora de duración que tiene lugar al final del curso, y que consiste en una selección de dos o tres preguntas sobre cada sesión de laboratorio, formuladas de forma muy similar a las cuestiones de las pruebas escritas realizadas al final de las sesiones. En la figura 1 se muestra un esquema de cómo se realizaba antes la evaluación y como se realiza ahora.

Esta propuesta es similar a la planteada en [3], pero en nuestro caso se sustituye la evaluación subjetiva por un examen objetivo al final de cada sesión que es fácil de corregir (las respuestas están tabuladas), y unas preguntas en el examen final para valorar individualmente a los alumnos.

2.2. La teoría

Previamente, en las clases de teoría aplicábamos el método clásico de la clase magistral. Explicábamos con detalle la totalidad del temario, incluyendo la realización de problemas en clase y proponíamos que se realizaran colecciones de

problemas por temas como trabajo fuera del aula. En caso de duda, los estudiantes podían asistir a tutorías para comprobar si la realización de los mismos había sido la correcta y se les guiaba en su resolución.

Aunque este método funcionó de manera satisfactoria durante los primeros años, con unas tasas de aprobados cercanas al 50%, los resultados académicos de los estudiantes fueron bajando progresivamente. En sus últimos años de utilización, la asistencia a clase del alumnado se había reducido sustancialmente, los estudiantes no resolvían las colecciones de problemas propuestos y la atención en clase había decaído notablemente. Inevitablemente, estos hechos se reflejaron en los resultados académicos. Aumentó el número de alumnos no presentados en los exámenes, y los que se presentaban obtenían unas calificaciones más bajas, disminuyendo significativamente el número de estudiantes aprobados hasta porcentajes cercanos al 30%.

Por este motivo nos planteamos probar nuevas estrategias docentes que fomentaran el trabajo continuo y su participación en las clases. Para ello, pensamos que era fundamental que el trabajo fuera del aula, ya sea individual o en grupo, tuviera un peso importante en el proceso de aprendizaje. Por este motivo, y tras analizar otras estrategias docentes como [1] [4] [7] [8], implantamos una nueva estrategia de evaluación que permitiera su personalización a cada alumno en particular. El método utilizado pretende flexibilizar la elección del alumno entre evaluación continua y examen final sin que tenga que elegir el primer día de clase entre ambas, como por ejemplo en [3]. Así, atendiendo al ritmo de las clases y de su propia disponibilidad, los alumnos pueden realizar actividades que les permitan reducir el peso del examen final.

En cuanto a las clases de teoría, intentamos incentivar la participación del alumnado y el aprendizaje continuo. Para ello, animamos (que no imponemos) a realizar diversos trabajos de dificultad y duración variable, tanto dentro como fuera del aula. Las calificaciones obtenidas en los trabajos conforman una nota acumulativa. Los trabajos cortos realizados en clase suelen tener un peso de entre 1 y 3 décimas en la nota final. En el caso de los trabajos fuera del aula, su valor está entre 2 y 5 décimas, por ser éstos de mayor

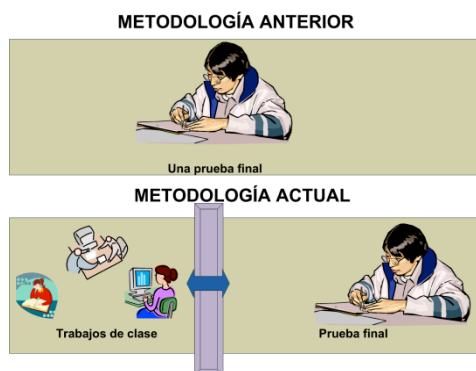


Figura 2. Cálculo de la nota de teoría en función de los trabajos puntuables opcionales realizados por los alumnos. La nota obtenida en los trabajos de clase actúa a modo de barra deslizante en el cómputo de pesos, regulando la importancia del examen final.

dificultad e implicar una mayor dedicación. Para obtener la calificación final de la asignatura, se combinan las obtenidas en los trabajos con la de la prueba final de la asignatura. Para personalizar este sistema de evaluación, el peso de la prueba final es variable, y se calcula en función de la puntuación obtenida en base a la realización de trabajos. En particular, el peso de la prueba final se computa como la diferencia entre el 10 y la calificación obtenida a través de la acumulación de puntos en base a trabajos. La fórmula aplicada para calcular la nota final (N_F) se expone en a continuación:

$$N_F = T + [E * ((10-T)/10)], \quad (1)$$

donde T es la nota de los trabajos realizados y E es la nota del examen final.

La Figura 2 muestra gráficamente cómo se calcula esta nota de teoría siguiendo el modelo anterior y el modelo actual.

Con esta forma dinámica de valorar la nota del examen se consigue que, aunque un alumno no asista a clase o no realice ninguno de los trabajos propuestos, tenga aún la posibilidad de obtener la máxima calificación en el examen final. Pero si por el contrario realiza los trabajos propuestos y por ejemplo acumula 2 puntos por este concepto, la nota del examen se re-normalizará entre 0 y 8 puntos, añadiéndose al 2 ya consolidado. De esta

forma, si un estudiante obtiene un 4 en el examen final, la nota final será de 5,2 (2 de trabajos + 3,2 del examen final). Si por el contrario, obtiene un 8 en el examen, tendrán como nota final 8,4 (2 de trabajos + 6,4 del examen).

En cuanto a los tipos de actividades se proponen, por un lado, trabajos cortos que se realizan durante las sesiones teóricas. Éstos consisten en pequeñas pruebas de entre 5 ó 10 minutos de duración que se realizan de forma individual al final de la explicación de cada tema o apartado de la asignatura. Estos trabajos, gracias a su corta extensión, son corregidos rápidamente por el profesor y entregados en la siguiente clase. Mediante esta actividad, obtenemos una realimentación inmediata sobre el proceso de aprendizaje, que nos permite corregir errores conceptuales, y además recompensamos la atención de los alumnos en el transcurso de las clases teóricas.

Por otro lado, aunque con menor frecuencia, proponemos trabajos o problemas más complejos para realizar fuera del aula y entregar en el horario de tutorías. Aunque estos trabajos se pueden realizar de forma individual o en grupo, otorgando al alumno la libertad de elegir su modo de trabajo, tanto la entrega como la evaluación debe realizarse individualmente y en horario de tutorías. La evaluación incluye una explicación, por parte del estudiante, de la forma en que ha resuelto el problema propuesto. Esta explicación no suele durar más que unos pocos minutos, y tiene el objetivo de verificar si el alumno ha comprendido realmente el trabajo realizado y asimilado correctamente los conocimientos que se pretendían adquirir. De esta manera, aunque los alumnos opten por realizar el trabajo de forma conjunta, nos aseguramos de que todos alcanzan los objetivos de aprendizaje propuestos en el trabajo. Esta afirmación se mantiene aún en el caso de alumnos que no han participado activamente en su realización e intentan aprovecharse de sus compañeros, pidiéndoles explicaciones suficientes como para poder acudir a su evaluación. En este caso, el aprendizaje de aquellos alumnos que realmente realizaron el trabajo queda reforzado mediante la técnica del aprendizaje mediante la explicación. Además, este tipo de situaciones ayuda a crear vínculos y compromisos de colaboración entre los alumnos,

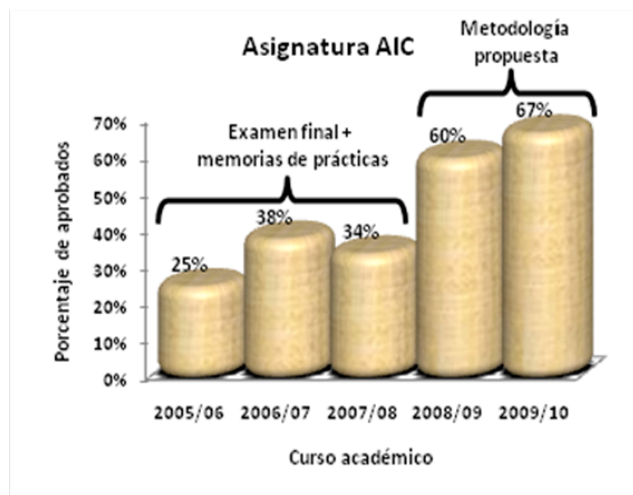


Figura 3. Porcentaje de alumnos aprobados en AIC durante los últimos 5 cursos.

que ayudan al desarrollo de aptitudes sociales útiles en su futuro profesional.

También se valora positivamente que alumnos propongan trabajos relacionados con la materia vista en clase al profesor. Estos trabajos pueden ser, desde descripciones de nuevas tecnologías desarrolladas recientemente, hasta programas o aplicaciones web que muestren conceptos presentados en las clases. Las propuestas deben ir acompañadas de un guión del trabajo a realizar, y su valoración depende de su dificultad y de cómo ponga de manifiesto los conceptos adquiridos, aunque como norma general son evaluados entre 0,5 y 1 punto.

Esta estrategia, en su conjunto, fomenta la unión entre los estudiantes y refuerza la idea de que la adquisición de conocimientos y habilidades es la meta final de todo proceso aprendizaje. Como consecuencia, los alumnos más brillantes o que más se esfuerzan son requeridos por sus compañeros para pedirles ayuda en la resolución de los trabajos planteados, complementando positivamente la labor del profesor.

3. Resultados

En esta sección se presentan los resultados académicos obtenidos al aplicar esta metodología en dos asignaturas de la Ingeniería Informática.

En la asignatura de AIC, donde veníamos aplicando el método clásico de clases descriptivas, y la memoria de prácticas y un examen final como forma de evaluación, la media de aprobados por curso (la suma de las dos convocatorias) ha pasado del 32% al 63%. Además, los resultados de las encuestas de evaluación del profesorado han mejorado sensiblemente en estos últimos años, reflejando la satisfacción del alumnado por esta nueva metodología.

A nivel subjetivo, se ha observado un aumento significativo en los niveles de atención en clase, atribuible a la posibilidad de realizar la actividad propuesta al final de la sesión y consolidar la puntuación adquirida en la misma. Además, la mejora conseguida ha quedado patente en los comentarios recibidos de alumnos que han cursado la asignatura en sus dos versiones. A este respecto, muchos alumnos nos han reconocido que la mejora más evidente la han experimentado en las sesiones de laboratorio. A modo de ejemplo, valoran positivamente que el tiempo que antes dedicaban a realizar (o presuntamente copiar) las memorias de las prácticas, ahora lo dedican a su preparación y estudio previo, afianzando así los conocimientos y habilidades técnicas adquiridas.

Actualmente, más de la mitad de los grupos de laboratorio acuden regularmente con la práctica previamente leída y suficientemente

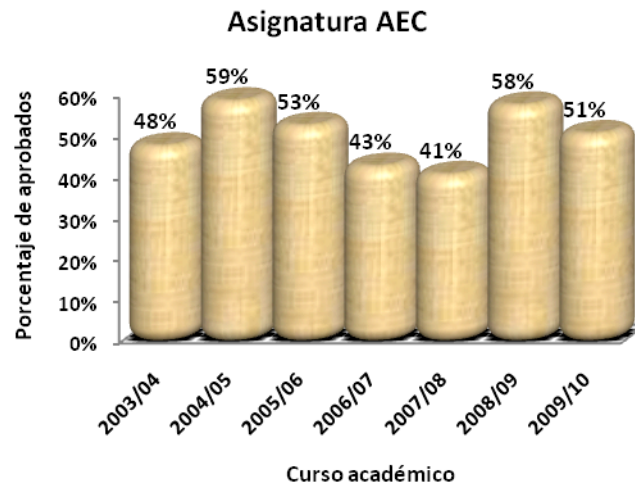


Figura 4. Media de alumnos aprobados en AEC los últimos 7 años en los que se ha aplicado la metodología propuesta.

preparada, y se ha podido observar un aumento de la actividad y atención en las sesiones de laboratorio.

En la figura 3 se muestra la evolución del porcentaje de alumnos aprobados en la asignatura de AIC, teniendo en cuenta la totalidad de los alumnos matriculados, incluidos los que no se presentan a los exámenes en ninguna de las dos convocatorias.

En la asignatura de AEC, en la que esta metodología se ha aplicado desde su implantación, los resultados son igualmente positivos. En la figura 4 se muestra, como en el caso anterior, el porcentaje de alumnos que aprueban por curso respecto al total de alumnos matriculados. Puede observarse que el número de alumnos aprobados se mantiene durante los últimos 7 años en un porcentaje medio cercano al 50% (entre el 41% y el 59%), aunque con las oscilaciones naturales entre diferentes cursos. Éste es un porcentaje bastante alto si tenemos en cuenta que aproximadamente un 25% de los alumnos matriculados no se presentan a ninguna de las dos convocatorias.

En ambas asignaturas el número de alumnos que asisten a clase ronda el 60% de los matriculados y entre ellos el 80% realiza los trabajos propuestos. Sobre este 80%, encontramos tasas de aprobados cercanas al

85%. Si comparamos la figura 3 con la 4, observamos que el promedio de alumnos aprobados en los cursos en los que la metodología propuesta fue aplicada, es menor en la asignatura de AEC que en la de AIC. Esto es debido a que muchos alumnos que fracasaban con la metodología anterior se han beneficiado de la implantación de la nueva metodología más centrada en el estudiante.

Como en la asignatura de AEC hemos aplicado la metodología propuesta desde el inicio de su implantación, el número de aprobados se ha mantenido estable, a lo largo de estos últimos 7 años.

4. Conclusiones

No cabe duda de que cualquier esfuerzo por parte del profesorado por animar la participación de los estudiantes en la clase y a trabajar en casa sobre los conocimientos adquiridos, forzando el estudio continuo de las asignaturas, se ve reflejado en el proceso de enseñanza/aprendizaje. Con la metodología que hemos presentado en este artículo motivamos al alumno a mantener la asignatura al día, para así poder realizar con éxito las pruebas que hacen en clase.

En general, los estudiantes realizan los trabajos más complejos en un plazo de una semana. Esto permite que sean rápidamente corregidos por el profesor y que sea posible dedicar parte del tiempo de alguna clase posterior a corregir errores conceptuales que se hayan podido detectar.

Hemos observado que la atención a los alumnos, la corrección de los trabajos cortos y la evaluación de los trabajos más largos, nunca ha excedido las 6 horas semanales que tenemos asignadas los profesores como horas de tutorías.

Después de varios años aplicando esta metodología, tanto los datos objetivos como los subjetivos obtenidos a partir de conversaciones directas con los alumnos corroboran su efecto positivo.

Agradecimientos

Los autores agradecen la financiación recibida desde el Vicerrectorado de Convergencia Europea y Calidad de la Universitat de València, a través de los proyectos DocenTIC y Finestra Oberta con códigos 08/DT/04/2009, 18/DT/05/2010 y 47/FO/35/2010; y desde el Ministerio de Educación y Ciencia y los fondos FEDER de la Comisión Europea, mediante los proyectos Consolider Ingenio 2010 CSD2007-00018 y CSD2006-00046, y el proyecto TIN2009-14475-C04.

Referencias

- [1] Arevalillo-Herráez, M., Benavent, X., Ferris, R., *Cambios metodológicos introducidos en la asignatura de Informática en la titulación de Matemáticas para su adaptación al Espacio Europeo de Educación Superior (EEES)*. Actas de las XV Jornadas de Enseñanza Universitaria de Informática, JENUI 2009, pp. 321-328, Barcelona, Julio 2009.
- [2] Barnnett, R., *Los límites de la competencia. El conocimiento, la educación superior y la sociedad*. Barcelona. Gedisa, 2001.
- [3] López, D., Herrero, J., Pajuelo, A. and Duran, A., *A Proposal for Continuous Assessment at Low Cost*. In 2007 ASEE/IEEE Frontiers in Education Conference (FIE 2007), pp. T1G1 - T1G6, Milwaukee, Wisconsin, USA, October 10-13, 2007.
- [4] Martínez, A., Hernández, C., Vivaracho, C.E., Simón, A., Arranz, G., Martínez, M., Prieto, O. *Introducción de metodologías activas en el aprendizaje de la informática: Experiencia del grupo GREIDI*. Actas de las XII Jornadas de Enseñanza Universitaria de Informática, JENUI 2006, pp. 347-354, Bilbao (Vizcaya), Julio 2006.
- [5] Ontoria Peña, A., *Aprendizaje centrado en el alumno: metodología para una escuela abierta*. Narcea Ediciones, 2009.
- [6] Otero, I., Achón, Z., Pérez, A., Martínez, A. *Estrategias de aprendizaje: del desarrollo intelectual al desarrollo integral*. Acción pedagógica, No. 16. pp. 194-202, 2007.
- [7] Sánchez, F., Cruz, J. L., Fernández, A., López, D., *Cómo diseñar una asignatura del EEES: de los objetivos formativos a la metodología y los contenidos*. Actas de las XII Jornadas de Enseñanza Universitaria de Informática, JENUI 2006, pp. 57-64, Bilbao (Vizcaya), Julio 2006.
- [8] Valderrama, E., Talavera, G., Montón, M., Martínez, B., Fernández, J.M., Muñoz, J., *Comparación de dos metodologías docentes utilizadas en los seminarios de Fundamentos de los Computadores*. Actas de las XIV Jornadas de Enseñanza Universitaria de Informática, JENUI 2008, pp. 181-188, Granada, Julio 2008.
- [9] Zabalza, M. A., *Evaluación de los aprendizajes en la Universidad*. A. García Valcárcel. Didáctica Universitaria. Madrid: La Muralla, 2001.

Píldoras formativas audiovisuales para el aprendizaje de Programación Avanzada

Luis Bengochea Martínez

Depto. de Ciencias de la
Computación

Universidad de Alcalá (Spain)

luis.bengochea@uah.es

Resumen

En este trabajo¹ se propone la utilización de tutoriales audiovisuales de corta duración como un nuevo recurso formativo de gran atractivo para los jóvenes estudiantes, ampliamente familiarizados con este tipo de formato similar al utilizado en YouTube.

Considerados como “píldoras formativas”, dichos tutoriales tienen como objetivo reforzar la comprensión de conceptos complejos de programación avanzada - concurrente y distribuida - que por su naturaleza dinámica pueden presentar una especial dificultad cuando solamente se utilizan materiales estáticos, como textos o diagramas. La generalización del uso de las plataformas de eLearning por los estudiantes de los nuevos grados, facilita la distribución en modo *streaming* de estos contenidos, que pueden ser visualizados en cualquier momento incluso desde dispositivos móviles.

Summary

The use of audiovisual, short-lived tutorials, is proposed here as a new educational resource very attractive for young students, widely familiar with this type of format similar to that used on YouTube.

¹ El presente trabajo ha sido desarrollado en el marco del proyecto para la integración de las tecnologías de la información y la comunicación en el proceso de enseñanza-aprendizaje, con código UAH/EV320, del programa de innovación educativa de la Universidad de Alcalá.

Considered "formative pills", these tutorials are intended to strengthen the understanding of complex concepts of advanced - concurrent and distributed - programming which by its dynamic nature could present a special challenge when only static materials as text or diagrams are used. Widespread use of e-learning platforms by the students of the new degrees facilitates the distribution mode streaming of this content that can be displayed at any time even from mobile devices.

Palabras clave

Recursos didácticos, píldoras formativas, píldoras de conocimiento, tutoriales audiovisuales, programación concurrente, *eLearning*.

1. Introducción

Casi desde el comienzo de los tiempos del cine comenzaron a rodarse documentales con fines didácticos. La utilización de cortometrajes de carácter científico se introdujo pronto en las universidades norteamericanas, como un complemento normal en la formación de los estudiantes. En España, las aulas magnas de las facultades de Ciencias disponían de medios para proyectar películas y en los años sesenta asistíamos regularmente a la visualización de filmaciones de diversos experimentos – rodados en los Estados Unidos – que se integraban en las clases magistrales como una parte más del material utilizado para el aprendizaje de la Física.

Con el paso del tiempo el formato técnico del medio - imágenes en movimiento apoyadas con sonido – utilizado con fines didácticos ha ido cambiando: película, video, dvd, internet,

YouTube, webcasts en diferido y online, etc. También a lo largo de este tiempo se han venido realizando estudios e investigaciones acerca de las oportunidades y beneficios que este medio proporciona, tanto a profesores como a estudiantes, en el proceso de enseñanza / aprendizaje.

La generalización del uso de internet y la disponibilidad de comunicaciones de gran ancho de banda ha permitido la aparición, en los últimos años, de sitios web donde se publican y almacenan ingentes cantidades de videos, en lo que ha dado en llamar el fenómeno video 2.0 [1]. En unos casos, de carácter generalista, como YouTube y en otros con orientaciones científicas, como *scivee* o didácticas, como *TeacherTube*. Y la producción de nuevos videos sigue aumentando sin cesar. En 2006 servían unos 100 millones de videos al día. En la actualidad ofrecen 1.000 millones diarios. Se han visto más de 100.000 años de video desde su nacimiento [2]. Indexadores como *blinkx* cuentan con más de 35 millones de horas de video registradas [3].

En muchos casos, las Universidades han aprovechado la oportunidad de publicación en estas plataformas abiertas, para crear canales propios donde dar a conocer noticias y eventos institucionales o incluso difundir sin restricciones de acceso material docente de alta calidad, y de esta forma aumentar su proyección y relevancia internacional.

Otras Universidades han optado por crear plataformas privadas en las que publicar los videos de producción propia, con diversos niveles de acceso a los contenidos: desde abiertos a cualquier internauta a restringidos a los estudiantes matriculados en un curso.

2. Aprendizaje con video

La mayor parte de los cursos que se imparten en las Universidades se hacen hoy en modalidad semipresencial. Profesores y estudiantes, además del aula o con independencia de ella, comparten un espacio virtual común en una plataforma de *e-learning* donde se publica y se accede a los contenidos, se realizan exámenes, se entregan trabajos y tareas colaborativas, se intercambian mensajes y se comparte información acerca del curso y de la materia objeto de estudio.

Como consecuencia de ello, surge la posibilidad de utilizar, de una manera sencilla,

contenidos didácticos en diferentes formatos, aprovechando las ventajas formativas que cada uno de ellos proporciona. Cada uno tiene características específicas que lo hacen más adecuado en algunos aspectos de la formación, y en este contexto el registro en video de una conferencia o una clase, para que puedan ser vistas de forma asíncrona en cualquier momento, aparece enseguida como una forma de contenido inherentemente asociada a la teleformación [4].

En las ingenierías relacionadas con la informática, hay estudiantes que pasan los cursos arrastrando numerosos conceptos mal entendidos y formas erróneas de resolver problemas. En cursos avanzados de programación, al tratar de resolver problemas cuya naturaleza compleja requiere un alto grado cognitivo, en ocasiones recurren a métodos inadecuados de resolución y lo que es peor, llegan a modificar sus propios conceptos inseguros, de forma que deriven en la solución por ellos propuesta [5].

La inclusión de contenidos ricos en fotografías e imágenes, videos, gráficos, animaciones, y otros elementos visuales, se ha demostrado efectiva en la enseñanza/aprendizaje de cursos de ingeniería [6], tanto para ayudar a comprender en profundidad un problema (aprendizaje), como para aplicar el conocimiento a la resolución de problemas nuevos (destreza).

3. Píldoras formativas

Las píldoras formativas, también conocidas por algunos autores (Maceiras et al. 2010) [7] como “píldoras de conocimiento”² son pequeñas piezas de material didáctico, creadas como objetos de aprendizaje de contenido audiovisual y diseñadas para complementar las estrategias tradicionales de formación y facilitar la comprensión de algunos aspectos de la materia curricular que presentan una mayor dificultad de comprensión para los estudiantes, ya sea por su hondura conceptual como por su complejidad instrumental.

La disponibilidad de una buena colección de videos de corta duración como parte del material didáctico de una asignatura, puesta a disposición de los estudiantes, permite que éstos las ejecuten y consuman de forma autónoma como complemento

² Con similar significado, se utilizan los términos en inglés “knowledge pills”, “skill pills” o “learning pills”.

formativo, mejorando así la eficacia de la transferencia de conocimientos.

Las píldoras formativas audiovisuales combinan creatividad, integración multimedia, sonido y animación. Comparten, además, el mismo lenguaje multimedia de comunicación que otros medios de publicación de contenidos audiovisuales, como “*YouTube*” tan familiares a nuestros estudiantes hoy en día.

El diseño en forma de piezas de contenido en pequeña escala, como el propuesto, con una duración de los videos de entre 5 y 15 minutos, está enfocado a integrarse en un contenido formativo más amplio en el que pueden aparecer varias piezas de esta naturaleza. Con esa duración, una píldora formativa no puede albergar una gran cantidad de información, pero a su vez debe formar una unidad autocontenida.

Dada la brevedad del video, es preciso identificar con claridad qué conceptos o técnicas, cuyo aprendizaje es susceptible de poder ser reforzado con este método, aparecen entre los que forman la unidad didáctica que se está creando.

Las píldoras formativas pueden estar ordenadas de varias formas y reutilizadas en diferentes unidades didácticas, pero sin perder su carácter de unidades con entidad propia por sí mismas.

3.1. Justificación pedagógica

En el aprendizaje multimedia el estudiante participa de tres procesos cognitivos: selección, organización e integración, identificados por Richard Mayer en los años noventa y que han generado una serie de experimentos que han desembocado en los cinco principios de cómo la multimedia puede ayudar a los estudiantes a comprender contenidos científicos. Estos principios son [8]:

- Representación múltiple. Es mejor presentar una explicación con palabras e imágenes que sólo con palabras. Los estudiantes se forman dos representaciones mentales diferentes, verbal y visual, y construyen conexiones entre ellas.
- Contigüidad. Los estudiantes comprenden mejor una explicación cuando las palabras y las imágenes son presentadas al mismo tiempo. La correspondencia entre imágenes y palabras facilita la construcción de enlaces referenciales entre ellas.

- Reparto de la atención. Las palabras deben ser presentadas en forma hablada y no escritas visualmente. Esto permite que cada discurso sea procesado de forma independiente sin que se produzca una sobrecarga entre el sistema de procesamiento visual y el auditivo.
- Diferencias individuales. Los principios anteriores son muy dependientes de diferencias individuales entre estudiantes. Fundamentalmente de su grado de conocimientos previos y de su capacidad de representación espacial.
- Coherencia. Los estudiantes aprenden mejor de un resumen coherente que resalta las palabras e imágenes más relevantes, que de un documento plagado de detalles adjuntos. Presentaciones más cortas ayudan a seleccionar la información relevante y a organizarla productivamente.

En el diseño de píldoras formativas audiovisuales, la enseñanza directa se divide en pequeños trozos. Las pequeñas unidades formativas minimizan la carga cognitiva y proporcionan la posibilidad de colocar piezas de información pertinente estratégicamente situadas dentro de situaciones complejas [9].

3.2. Ejemplos de videotutoriales

Algunas Universidades españolas tienen en marcha proyectos para producir píldoras formativas de elevada calidad. En varios casos están utilizando, para su producción, una herramienta, *Polimedia*, desarrollada en la Universidad Politécnica de Valencia en 2007, que permite la grabación de un profesor hablando junto a una pantalla en la que se muestran transparencias o videos, mientras explica una clase. Para minimizar la ocupación del archivo y facilitar su transmisión, se utiliza una resolución baja en la grabación de la imagen del profesor y otra mayor para la pantalla.

Desde el punto de vista tecnológico, las producciones creadas con *Polimedia* son videos con audio. Sin embargo, desde el punto de vista del contenido instruccional, en muchos casos el soporte fundamental consiste en una presentación de transparencias estáticas acompañadas por la figura parlante del profesor que va explicando la materia.

También pueden diferir, en cuanto a su duración, de lo que hemos visto que es una píldora

formativa, ya que en algunos casos lo que pretenden es tener la grabación de una clase completa para que los estudiantes puedan visualizarla como sustitución a la clase presencial.

Sin embargo, también sirve para crear píldoras formativas o píldoras de conocimiento, como las desarrolladas en la Universidad de Vigo para ser incluidas en cursos de posgrado de ingeniería industrial, con el mismo sentido que le estamos dando aquí. Los autores de la experiencia piloto [10] señalan así sus características: “ser objetos autónomos, estar enfocados a un tema, ser indivisibles, estar dirigidos a una audiencia específica y ser agrupables”. También otra característica a tener en cuenta es el hecho de que al aparecer la imagen del profesor hablando a la audiencia se resuelven gran parte de los problemas asociados con los derechos de autor.

Algunas universidades norteamericanas, como la de California Berkeley o la de Stanford, han optado por grabar en vídeo las clases completas de Programación impartidas por el profesor correspondiente y publicarlas en su canal de YouTube [11].

Sin embargo, a pesar de sus grandes ventajas en cuanto a la posibilidad de difusión de los cursos a estudiantes de todo el mundo, esta solución presenta varios inconvenientes, como:

- La duración es la misma que la de la clase tradicional (50 minutos), lo que hace que sea incómodo verla más de una vez si lo que se quiere es reforzar algún concepto desarrollado en esa clase.
- En el vídeo se recoge todo lo que ocurre en el aula, aunque no sea relevante para la lección (Figura 1).
- Requiere instalaciones y recursos costosos: iluminación, videocámaras profesionales, personal especializado que realice la grabación, etc.
- Elevada ocupación en disco de los archivos multimedia creados.

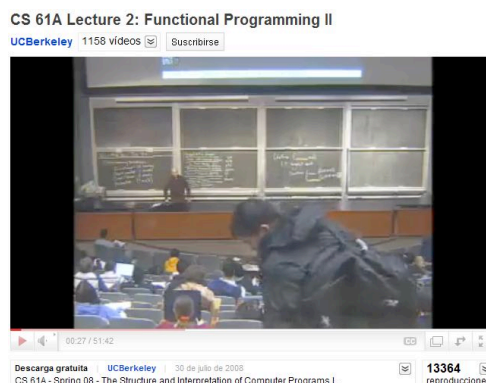


Figura 1: Clase grabada en la U.Berkeley

4. Producción de vídeos

Uno de los problemas más importantes a la hora de producir vídeos formativos, es la necesidad de disponer de medios técnicos sofisticados y de profesionales cualificados.

Esto impide producir vídeos “sobre la marcha” en respuesta a las eventualidades que ocurren durante un curso. Por ejemplo, ante una consulta planteada por un estudiante en un foro de una asignatura de programación y las respuestas aportadas por sus compañeros, el profesor puede detectar una carencia conceptual o procedimental que podría aclarar mucho mejor de viva voz, dando sonido a un vídeo donde se muestra una ventana de su ordenador en la que actúa, que escribiendo largos párrafos y pegando trozos de código.

Para poder hacer esto, no es preciso acudir a la oficina tecnológica de la Universidad a solicitar un hueco para grabar el vídeo. Basta con unas herramientas y unos conocimientos al alcance de cualquiera.

Nuestro trabajo ha consistido en estudiar las diferentes herramientas informáticas que permiten la producción de píldoras formativas con la calidad y características que se requerían, así como las posibilidades de integración en la plataforma de teleformación del Aula Virtual de nuestra Universidad.

4.1. El proyecto

El comienzo de los nuevos grados relacionados con la Ingeniería Informática supuso la generalización del uso de la plataforma de

eLearning como centro de actividades de los estudiantes. Surgió entonces el proyecto de enriquecer los contenidos publicados en ella, fundamentalmente páginas interactivas en formato HTML y documentos PDF y Powerpoint, con nuevos contenidos audiovisuales con las características de las píldoras formativas expuestas en los apartados anteriores en las que se muestra, en formato de vídeo flash, el contenido de una pantalla de ordenador en la que se van a ir alternando:

- Una presentación powerpoint o similar con contenidos conceptuales.
- Una pizarra virtual donde el profesor puede escribir o trazar esquemas que apoyen su razonamiento.
- Un entorno de programación donde se irá viendo cómo se construye el programa, y el funcionamiento dinámico de la ejecución de éste.
- Fragmentos de videos de apoyo.
- Etc.

Al mismo tiempo, se está escuchando la voz del profesor explicando los conceptos que intervienen, las acciones que va realizando y cómo éstas afectan a los resultados y en general cuantas aclaraciones sirvan para comprender mejor el aspecto de la programación tratado.

Las sesiones presenciales de los laboratorios de programación avanzada constan de una primera parte en la que el profesor expone los conceptos a desarrollar en la práctica y una segunda en la que los estudiantes realizan los programas correspondientes y exponen las dudas y dificultades con las que se enfrentan. En el caso de enseñanzas semipresenciales o totalmente virtuales, como el caso de estudios de másteres y especialización en los que figuran asignaturas con técnicas de programación, el que no haya esa primera parte de intervención del profesor para centrar los objetivos de la práctica que va a abordar el estudiante, hace que el interés por usar este tipo de material audiovisual sea mucho más relevante.

Una primera fase del proyecto ha consistido en estudiar las diferentes alternativas que existen, tanto de software libre, como propietario, con el fin de seleccionar la aplicación más apropiada para la producción del material audiovisual, teniendo en cuenta criterios como:

- Calidad de la imagen y sonido VS tamaño de los archivos generados.

- Posibilidad de difusión en forma de *streaming* desde la plataforma *eLearning* de la Universidad para que:

- a) los estudiantes puedan seguirlo sin necesidad de descargárselo a su ordenador.
- b) impedir la utilización fraudulenta de copias de los archivos descargados.

4.2. Selección de las aplicaciones informáticas

Para llevar a cabo nuestros primeros videotutoriales, estuvimos evaluando diversas aplicaciones informáticas de captura dinámica de pantallas, como fueron:

- *Adobe Captivate*
- *Camtasia Studio*
- *Snagit*

En todos los casos, aparte de otras funcionalidades, lo que nos interesaba era realizar videos en los que se capturase todo lo que aparecía en una zona de la pantalla del ordenador, al mismo tiempo que fuese grabando las explicaciones del profesor recogidas a través un micrófono.

Finalmente, por su sencillez y la relación entre la calidad del video y del sonido, frente a su ocupación en disco, nos decidimos por *Snagit* [12], creada por la empresa *TechSmith*, que también hace *Camtasia Studio*.

Los videos son producidos, en formato AVI, con un tamaño de 640x480 y 25 *frames/seg.*, son posteriormente transformados a formato Flash Video (FLV) con objeto de integrarlos en las páginas HTML del contenido de la asignatura y que se puedan visualizar en modo *streaming* a través de la plataforma virtual de la Universidad sin necesidad de que los estudiantes tengan que descargárselos a sus ordenadores.

Para la transformación de un formato a otro de video, sin pérdidas apreciables de calidad, se ha utilizado la versión libre del programa "*Any Video Converter*" [13].

4.3. Videotutoriales publicados

Finalmente, hasta la fecha se han producido varios videotutoriales, referidos a los aspectos más relevantes de la Programación Concurrente (fig 2), que permiten que los estudiantes de la asignatura puedan compartir con el tutor todo el proceso de creación de aplicaciones en las que se utilizan

diversas técnicas de sincronización de procesos concurrentes.

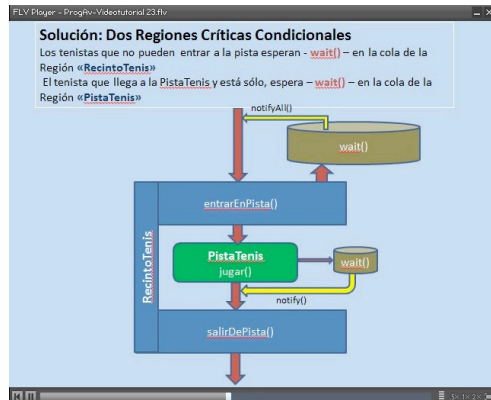


Figura 2: Videotutorial de Programación Concurrente

Los videos tienen una duración media de 12 minutos, lo que hace que se puedan volver a visualizar varias veces, en el caso de que el estudiante tuviese necesidad de reforzar la asimilación de los conceptos tratados.

También tienen una duración idónea para ser publicados en una web pública de videos, como YouTube, aunque ello debería hacerse dentro de un canal institucional, ya fuese de la propia Universidad o bien de la Escuela.

La buena calidad de imagen obtenida, permite que los estudiantes visualicen los videos en modo de pantalla completa. Esto facilita la lectura de las líneas de código de los programas que se van desarrollando en el transcurso del tutorial (fig. 3).

```

public RecintoTenis() {
    espera[0] = new Semaphore(0, true); //semáforos "justo". Los que esp
    espera[1] = new Semaphore(0, true); //saldrán en el orden que entraron
}

public int entrarEnPista(Tenista t) {
    int genero = t.sexo();
    aperturaInicial(genero);
    colaEspera.meter(t);
    try {
        espera[genero].acquire();
    } catch (InterruptedException e) {}
    if (dentro.cuantos() == 0) {
        dentro.meter(t);
        colaEspera.sacar(t);
        espera[genero].release();
        return 1;
    }
    dentro.meter(t);
    colaEspera.sacar(t);
    return 2;
}

public void salirDePista(int numZug) {
    if (numZug == 1) {

```

Figura 3: Las líneas de código son perfectamente legibles en modo de pantalla completa.

Aunque no se ha llevado a cabo ninguna encuesta que permitiese medir los beneficios que este nuevo material didáctico ha tenido sobre el aprendizaje de los estudiantes, si que se han producido algunas manifestaciones acerca de su utilidad en los foros de la plataforma.

5. Conclusiones

El material teórico de autoaprendizaje puesto a disposición de los estudiantes en un curso virtual de programación, está formado por contenidos de carácter conceptual entremezclados con muchos otros elementos de carácter procedimental. Aunque los estudiantes primerizos seguramente utilizarán todos los elementos del curso en su autoaprendizaje, es posible que otros estudiantes más avanzados en programación puedan pasarlos por alto por creer que ya los saben o por falta de interés, ocasionando errores conceptuales que repercuten en su forma de plantear y resolver los problemas de programación [5].

Píldoras formativas audiovisuales que aborden los conceptos básicos y que se encuentren distribuidas convenientemente a lo largo del material de autoaprendizaje, pueden contribuir a motivar a los estudiantes e impedir que pasen por alto dichos conceptos. La autorregulación del aprendizaje online requiere interés y motivación para fomentar la participación y conseguir el éxito de un curso y a ello pueden contribuir los elementos multimedia audiovisuales [9].

Los videotutoriales de corta duración y pequeño tamaño pueden ser visualizados en dispositivos móviles como teléfonos, PDAs y tabletas, lo que puede constituir un aliciente adicional importante a la hora de emplearlos en el diseño de un curso universitario.

La producción de píldoras formativas puede hacerse de forma autónoma por un profesor, utilizando un equipamiento informático básico y aplicaciones asequibles y fáciles de utilizar. Dado que su finalidad es puramente didáctica y su público objetivo los estudiantes de un curso de programación avanzada, no necesitan hacer uso de los medios sofisticados de producción ni de publicación institucionales que, generalmente, responden a planteamientos muy diferentes.

Este tipo de videotutoriales en forma de píldoras de conocimiento, también pueden empaquetarse, con sus metadatos convenientemente etiquetados, como objetos de aprendizaje para integrarse en los repositorios de la institución para su reutilización posterior.

Referencias

- [1] C.Snelson and P.Elison-Bowers. *Using YouTube videos to engage the affective domain in e-learning*. Research, Reflections and Innovations in Integrating ICT in Education. Edited by A. Méndez-Vilas et al. Published by FORMATEX, Badajoz, Spain (2009). VOL.3 (pp. 1235-1512). ISBN 978-84-692-1788-7.
- [2] Piergiorgio M. Sandri. *Aprender con YouTube*. La Vanguardia Digital 08/02/2010.
- [3] Webs de Video: *YouTube: Broadcast yourself* (www.youtube.com); *TeacherTube: Teach the world* (www.teachertube.com); *SciVee: Make your research known* (www.scivee.tv); *blinkx* (www.blinkx.com) visitadas:Feb2011.
- [4] Pullen,J.Mark. *Applicability of internet video in distance education for engineering*. 31 ASEE/IEEE Frontiers in Education Conference. October 10-13, 2001 Reno, NV.
- [5] L Kaczmarczyk, E.Petrick, J.P.East, and G.L. Herman *Identifying student misconceptions of programming*, Proceedings of the Forty-First ACM Technical Symposium on Computer Science Education, Milwaukee, WI, pp. 107-111, March 2010.
- [6] González,M.J.; Montero,E.; de Heredia,A.B.; Martínez,D.: *Integrating digital video resources in teaching e-learning engineering courses*. Education Engineering (EDUCON), 2010 IEEE Publication Year: 2010, Page(s): 1789–1793.
- [7] Rocio Maceiras, Ángeles Cancela y Vicente Goyanes. *Aplicación de Nuevas Tecnologías en la Docencia Universitaria*. Formación Universitaria Vol. 3(1), 21-26 (2010).
- [8] Richard E. Mayer; Roxana Moreno. *A Cognitive Theory of Multimedia Learning: Implications for Design Principles*. In "Handbook of Applied Cognition", 2nd Edition. Francis T. Durso (Editor), Wiley 2007. ISBN: 978-0-470-01534-6.
- [9] C.Snelson, & P.Elison-Bowers, *Micro-level design for multimedia-enhanced online courses*, MERLOT Journal of Online Learning and Teaching, 3, (2007).
- [10] A.Sanchez;A.Cancela;R.Maceiras;S.Urrejola. *Multimedia Productions: Knowledge Pills for University Teaching*. IADIS International Conference e-Society 2010. Avila (Spain) ISBN:978-972-8939-07-6. Pages:351-355.
- [11] Canales en YouTube: *UCBerkeley's Channel* (www.youtube.com/user/UCBerkeley); *Stanford University Channel* (www.youtube.com/user/StanfordUniversity) visitadas:Feb2011.
- [12] TechSmith. *SnagIt* (www.techsmith.com/snagit/).
- [13] Any Video Converter (www.any-video-converter.com/).

Sesión 5:
Métodos pedagógicos innovadores II

OPERA: Una herramienta soporte para el Aprendizaje Basado en Proyectos.

Pablo Fernández
Dpto. Lenguajes y Sistemas
Informáticos
Universidad de Sevilla
Avda. Reina Mercedes s/n
41012 Sevilla
pablofm@us.es

María José Jiménez
Dpto. Matemática Aplicada I
Universidad de Sevilla
Avda. Reina Mercedes s/n
41012 Sevilla
majiro@us.es

Rocío García
Dpto. Arquitectura y
Tecnología de Computadores
Universidad de Sevilla
Avda. Reina Mercedes s/n
41012 Sevilla
rocio.garcia@atc.us.es

Resumen

En los últimos años el Aprendizaje Basado en Proyectos (ABP) ha ganado popularidad en la docencia de la Ingeniería Informática por su especial adaptación a los objetivos marcados por el EEES. Sin embargo, la puesta en marcha de ABP en asignaturas suele acarrear un sobreesfuerzo derivado de la falta de adaptación específica a esta metodología en las herramientas actuales de soporte al aprendizaje.

En este trabajo presentamos la herramienta OPERA que se centra en disminuir el esfuerzo de la aplicación de ABP por parte del profesorado a través de un soporte específico para la creación de grupos y temáticas y la evaluación que se plantea habitualmente al aplicar esta metodología. En segundo lugar, la herramienta OPERA está orientada a potenciar la visibilidad de los trabajos más allá de la asignatura, mediante la creación automatizada de microportales de proyectos que se integran mediante el paradigma de “Red Social” donde los alumnos pueden interactuar entre ellos. Por último, el acercamiento mediante OPERA en la asignatura abre la posibilidad de relacionar trabajos de distinto curso académico para que el alumno, antes de iniciar su proyecto, tenga una base de conocimiento de trabajos con una temática similar al suyo y se potencie el desarrollo de un catálogo de buenas prácticas a lo largo de distintos cursos.

Summary

During the last years, Project-Based Learning (PBL) has been developed as a prominent approach in the teaching of Computer Science/Engineering. However, the actual application of PBL in a concrete subject provides

some drawbacks such as the increasing effort for the faculty and the lack of specific focus by current teaching support tools.

In this work we present a new tool specifically designed to support the teaching methodology of Projects-Based Learning with the following benefits: First, it eases the burden of managing the assignments and the evaluation of the projects; second, students can access to previous projects and interact with others in a social-web-like manner; and finally, the tool generates a public microportal for projects to boost their visibility to other students and the general public.

Palabras clave

Aprendizaje Basado en Proyectos (ABP), gestión de proyectos, redes sociales

1. Motivación

A lo largo de los últimos años, en la Escuela Técnica Superior de Ingeniería Informática de la Universidad de Sevilla, varias asignaturas han puesto en práctica la metodología de Aprendizaje Basado en Proyectos (ABP); esta experiencia ha mostrado ser de utilidad para aumentar la motivación de los alumnos y su desarrollo competencial. Sin embargo, a pesar de los importantes beneficios para el proceso de aprendizaje se han identificado dos problemas en su aplicación que se agravan en función del ratio alumnos/profesor de la asignatura: (i) Por un lado, el aumento significativo en la carga de gestión de la asignatura debido a la creación de grupos de alumnos, la creación y asignación de temáticas, la gestión de los entregables o la logística para gestionar la evaluación de los proyectos (normalmente mediante defensas en grupo).

(ii) Por otro lado, se pone de manifiesto una carencia de capitalización del conocimiento generado por los alumnos de un curso académico al siguiente. Disponer de soluciones de cursos anteriores desarrolladas por los alumnos podría constituir una fuente de referencia muy relevante y permitiría identificar un conjunto de “buenas prácticas” de desarrollo para los alumnos del curso actual. Sin embargo, actualmente la información generada durante estos años se pierde debido a la deslocalización de la misma, ya que no existe ninguna herramienta que la unifique ni permita gestionar la visibilidad de un curso a otro.

Fruto de la constatación de las carencias mencionadas, en el curso 2009/2010, se emprende la creación de la herramienta OPERA con tres objetivos principales: facilitar al profesor la gestión de una asignatura al aplicar la metodología ABP, posibilitar el desarrollo de un catálogo de buenas prácticas como recurso de referencia a lo largo de distintos cursos y, por último, aumentar la motivación de los alumnos potenciando la visibilidad de los proyectos y aplicando un acercamiento de tipo Red Social que permita la interacción entre ellos.

El resto del trabajo presenta la siguiente estructura: En la próxima sección, se introduce la metodología ABP; en la sección 3 se presentan la herramienta OPERA mostrando sus características principales; en la sección 4 se presenta la evaluación de la herramienta a través de la puesta en marcha de una experiencia piloto sobre tres asignaturas en el primer cuatrimestre del curso 2010/2011 y el análisis de una encuesta de valoración de los distintos usuarios; por último, en la sección 5 se discuten las principales conclusiones y el trabajo futuro.

2. Introducción a la metodología Aprendizaje Basado en Proyectos

El aprendizaje basado en Proyectos (ABP) es un caso particular del Problem-based Learning (PBL) que, según los expertos en la materia, se define como “un entorno constructivista de aprendizaje por excelencia” [9]. Este acercamiento ha sido empleado desde hace mucho tiempo en la enseñanza universitaria, especialmente en relación a la docencia de casos médicos. Sin embargo, en los últimos años se ha puesto de manifiesto la inquietud por el enfoque ABP dentro de los estudios de Ingeniería Informática [2,4,6,8,10,11].

Aunque el modelo ABP difiere entre diferentes escuelas, tiene tres características esenciales: los problemas como estímulo para el aprendizaje, los tutores como facilitadores, y el trabajo en grupo como estímulo para la interacción [5]. Los partidarios del ABP argumentan que este modelo tiene el potencial de preparar a los estudiantes más efectivamente para su futuro aprendizaje porque está basado en cuatro visiones sobre el aprendizaje: constructivo, auto-dirigido, colaborativo y contextual [5]. Sin embargo, los detractores del modelo llegan a las siguientes conclusiones: (a) los estudiantes ABP presentan potencialmente lagunas importantes en la base de su conocimiento cognitivo; (b) no demuestran patrones de razonamiento experto; y (c) el ABP es muy costoso [1].

Otros en una situación intermedia concluyen que el ABP tiene un efecto positivo importante en competencias pero un efecto negativo en conocimiento [12].

Por otra parte, de acuerdo a los expertos en la materia, se necesita realizar investigación orientada al diseño que permita aunar la investigación empírica educativa con el diseño basado en la teoría de los entornos de aprendizaje que nos ayuden a entender cómo, cuándo y por qué las innovaciones en educación funcionan o no en la práctica [3].

3. OPERA

Desde un punto de vista práctico, uno de los principales inconvenientes de ABP es la carga de trabajo de gestión adicional que supone para el profesorado [6] como por ejemplo la creación de grupos y temáticas, la gestión de entregables y la logística de evaluación de los proyectos.

En este contexto, pese a darse avances importantes gracias al uso de las plataformas de ayuda al aprendizaje actuales, es necesario un importante esfuerzo en conocer el modelo general de la herramienta y la configuración concreta para adaptar el modelo y hacer uso de ABP; este proceso de personalización se ve agravado en el caso de que se desee refinar el flujo de interacción entre profesores y alumnos (por ejemplo teniendo entregas en varias fases y múltiples defensas a lo largo del curso). Así mismo, desde el punto de vista de una plataforma típica de aprendizaje, los trabajos realizados por los distintos alumnos son considerados como un conjunto de archivos que,

en el caso de que se quieran utilizar como recursos de referencia deben ser procesados, catalogados y reintroducidos en la plataforma de manera manual por parte del profesor.



Figura 1. Entrada principal de OPERA

Tomando estas carencias como punto de partida, hallamos la motivación para la creación de la presente herramienta que pretende facilitar la puesta en marcha del ABP a través de un diseño ad-hoc para esta metodología en base a la experiencia de varios años de aplicación en distintas asignaturas de la Escuela Técnica Superior de Ingeniería Informática de la Universidad de Sevilla. De esta forma, el objetivo principal de la herramienta OPERA (Open Project Portal) es facilitar la puesta en marcha del enfoque ABP reduciendo la carga de gestión y configuración por parte del profesorado y potenciar los trabajos como recursos de referencia que persisten de un curso académico a otro. En este contexto, OPERA tiene un enfoque fundamentalmente distinto a las plataformas actuales debido a que da soporte a la asignatura desde una perspectiva global sin considerar cada uno de los cursos académicos como un compartimento estanco.

De manera complementaria, OPERA plantea un enfoque de red social orientado a potenciar la motivación de los alumnos y premiar la calidad de los trabajos a través del aumento de la visibilidad de los trabajos tanto en el seno de la asignatura como desde un punto de vista externo. A lo largo de la subsección siguiente se desglosan las características más importantes de la herramienta OPERA.

A la hora de desarrollar la herramienta se ha elegido un modelo de código abierto basado en la forja pública que proporciona la organización RedIris; de esta forma, la herramienta está disponible para su descarga en la dirección <https://forja.rediris.es/projects/opera/>.

Actualmente, el servidor donde se ha implantado el sistema está instalado en un servidor virtual basado en un procesador Intel Xeon X5560 2.80 GHz con 2 GB de memoria RAM y un disco duro de 11 GB; el sistema operativo instalado ha sido Red Hat Enterprise Linux 5 de 32 bits. Con respecto a la tecnología de desarrollo, se ha empleado el framework Zend PHP y en consecuencia, se requiere la instalación de un servidor web apache y un servidor de bases de datos MySQL.

3.1. Características principales

Configuración de cada curso (Fig. 2): A pesar de tratar la asignatura de forma global, pudiendo relacionar trabajos de un curso con otros anteriores, cada curso académico es configurable en cuanto a la forma en que se crearán los grupos, la asignación de trabajos, número de entregables, etc.

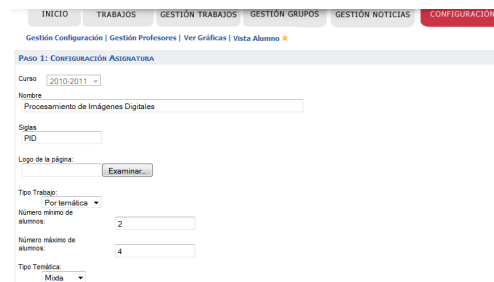


Figura 2. Configuración de un curso

Gestión de grupos de alumnos (Fig. 3): Los alumnos pueden gestionar de manera autónoma la creación y modificación de grupos hasta una fecha determinada; una vez superada esta fecha el profesor puede gestionar de forma visual (Fig. 3) los grupos creados. La herramienta permite una reorganización automática de aquellos alumnos que no se hayan inscrito en ningún grupo.



Figura 3. Captura de pantalla gestión de grupos.

Gestión de temáticas de trabajos (Fig. 4): el profesor puede asociar a cada temática una descripción, trabajos anteriores relacionados, un artículo de investigación; además puede categorizarla mediante el uso de etiquetas.

Figura 4. Captura de pantalla creación de temática.

Microportal público de proyectos (Fig. 5): Como elemento troncal, cada proyecto tiene asociado un microportal que tiene una doble función: (i) Por un lado, este portal actúa como punto de coordinación entre los distintos alumnos del grupo permitiendo que centralicen las entregas de material, seleccionen los días de defensa o realicen comentarios sobre el devenir del proyecto (ii) Por otro lado, los microportales tienen una vista parcial en la que se muestra toda la información pública (tales como la descripción, el entorno de desarrollo o videos explicativos) del proyecto a otros usuarios (o de forma anónima) potenciando la visibilidad del proyecto más allá del grupo, la asignatura o el curso académico. En este contexto de portal público, la documentación entregada por los alumnos se publica con base en la opción de configuración seleccionada por el profesor: Puede quedar expuesta desde el momento en el que se sube a OPERA, o puede ser accesible de forma anónima una vez pasada la fecha límite de entrega de todos los grupos.

Figura 5. Captura de pantalla de microportal de proyecto.

Gestión de defensas de trabajos (Fig. 6): permite elaborar un calendario de defensas de los trabajos incluyendo la posibilidad de sesiones paralelas. Los alumnos podrán inscribirse a su conveniencia en el horario que deseen manteniendo las restricciones marcadas por los profesores. Asimismo, el sistema notifica por correo electrónico si el número de sesiones está siendo copado para incluir la posibilidad de aumentar las sesiones dinámicamente.

Horario	Sesiones
12:30 - 13:00	4/4
13:00 - 13:30	4/4
13:30 - 14:00	4/4
14:00 - 14:30	3/3

Figura 6. Captura de pantalla gestión de defensas.

4. Evaluación

La herramienta ha sido probada como soporte al ABP en tres asignaturas distintas de la Escuela Técnica Superior de Ingeniería Informática. Los datos de puesta en marcha de la asignatura han sido los siguientes:

1.- En la asignatura Procesamiento de Imágenes Digitales (PID)¹ se ha utilizado OPERA para la creación de 27 grupos de trabajo en el presente curso, de los cuales 25 han sido creados por las profesoras y 2 han sido propuestos por los propios alumnos. La formación de grupos se ha realizado “por temática”, que es una de las opciones de configuración del curso. Así, se realiza una lista de propuestas por el profesor (que el alumno puede engrosar aportando su propia propuesta), cada una de ellas de trabajos sobre temáticas muy concretas (opción de temática “cerrada”, principalmente basadas en un artículo reciente de investigación), en las que el alumno se puede inscribir. Una vez han realizado el proyecto, los grupos de trabajo han subido al portal la documentación correspondiente en distintos archivos, que quedarán almacenados para su consulta pública y como referencia para los alumnos de cursos venideros.

2.- En el caso de la Ampliación de Base de Datos (ABD)² el portal OPERA se ha usado en esta asignatura para la creación de más de 60 grupos de trabajo cada uno con distinta temática y a diferencia de PID la configuración utilizada ha sido “por grupo”, de forma que cada grupo debe proponer una temática concreta para su aplicación (opción temática “abierta”). Una vez han realizado el proyecto, los grupos de trabajo han subido al portal el código de la aplicación y se han configurando una serie de días para la defensa del trabajo donde los grupos se han registrado de forma autónoma. La lista de trabajos realizados ponderados por calificación de los profesores se encuentra en el portal para su consulta pública y como referencia para futuros alumnos.

3.- En la asignatura Ingeniería del Software de Gestión 2³ (ISG2) se ha creado en torno a 19 grupos y, al igual que ABD, se ha configurado el curso mediante la opción “por grupo”. Por otro lado, en este caso se ha utilizado la opción de temática “mixta” para que tanto alumno como profesor puedan hacer propuestas.

4.1. Valoraciones de los usuarios

Para conocer la opinión que se ha creado el alumno con respecto al uso del portal, hemos realizado una encuesta de valoración, a la que se

puede acceder desde el propio portal. Se han realizado un total de 67 encuestas por parte de los alumnos. A continuación analizamos los resultados más significativos de la encuesta de valoración para cada una de las preguntas:

1. *¿Qué nivel de dificultad has encontrado en el uso de OPERA en general?*

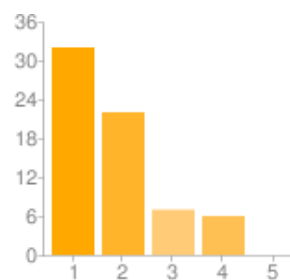


Figura 7. Dificultad de uso (1 sencilla de utilizar, 5 difícil de utilizar).

El 48% de los encuestados encontró la aplicación muy sencilla de usar. Este aspecto era altamente importante debido a que no se disponía de un tiempo para explicar la herramienta y resultó clave para el éxito de aplicación.

2. *¿Te ha resultado satisfactorio el sistema de creación de grupos y trabajos?*

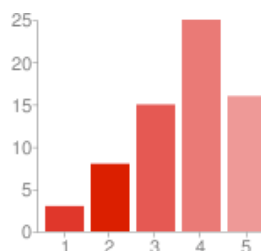


Figura 8. Sistema de creación de grupos (1 poco satisfactorio, 5 muy satisfactorio).

Aproximadamente el 40% de encuestados encontraron bastante satisfactorio (4-5) el sistema. Creemos que este resultado es debido a que la creación de grupos se realiza de forma autónoma por parte de los propios alumnos y se tiene un sistema flexible de cambio y actualización de grupos. Con base en experiencias previas de aplicación de ABP

¹ <http://opera-portal.us.es/pid>

² <http://opera-portal.us.es/abd>

³ <http://opera-portal.us.es/isg2>

en asignaturas con un alto número de alumnos se identifica este aspecto como uno de los que genera más problemas de gestión; para paliar este aspecto los profesores normalmente diseñan un sistema rígido en el que los alumnos no pueden tener flexibilidad a la hora de crear o modificar grupos.

3. *¿Ves útil que los proyectos estén disponibles de un curso a otro como referencia para futuros alumnos?*

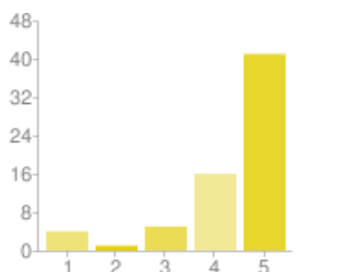


Figura 9. Visibilidad de proyectos anteriores (1 poco útil, 5 muy útil).

El 61% ha considerado muy útil la visibilidad de trabajos anteriores. En esta cuestión encontramos la mayor unanimidad desde un punto de vista de valoración positiva. Creemos que este aspecto es relevante para los alumnos debido a que, por un lado, permite disponer de un mayor número de recursos de apoyo a la hora de abordar un nuevo problema y, por otro lado, aumenta la visibilidad del trabajo realizado por parte del alumno para cursos posteriores. Este último aspecto podría ser una motivación para aumentar la calidad de los proyectos y es una de las características clave que se estableció como punto de partida en el diseño de la herramienta: desarrollar a lo largo del recorrido de una asignatura en varios cursos académicos un catálogo estructurado de proyectos de referencia que permita ser consultado como recurso de referencia por parte de los alumnos. En este sentido, cabe destacar que la consulta de proyectos se muestra ordenada por la calificación obtenida (o por la valoración de otros alumnos) permitiendo identificar el conjunto de proyectos más interesantes de forma directa.

4. *¿Ves interesante que los alumnos puedan valorar/puntuar el trabajo de otros compañeros?*

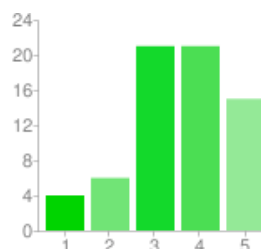


Figura 10. Valoración de proyectos de compañeros (1 poco interesante, 5 muy interesante).

Sólo el 36% de alumnos encuentran bastante interesante (4-5) la posibilidad de valorar el trabajo de los compañeros. Este aspecto consideramos que necesitaría una mejora y se han propuesto mejoras para potenciar esta motivación en la valoración entre compañeros. Una de las propuestas, ha sido la de incluir una parte de la calificación final con base en las calificaciones de los compañeros; en este sentido, a la hora de implementar este tipo de propuestas sería necesario establecer los mecanismos de control necesarios para que se garantizara un sistema de evaluación justo y penalizara la degradación del sistema mediante votos malintencionados.

5. *¿Has utilizado la integración de OPERA con Facebook?*

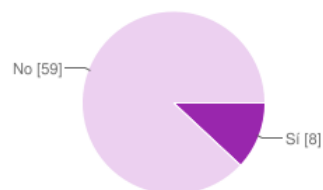


Figura 11. Cantidad de alumnos que han hecho uso de Facebook.

6. *¿Ves interesante que OPERA esté integrado con otros sistemas de redes sociales como Facebook o Twitter?*

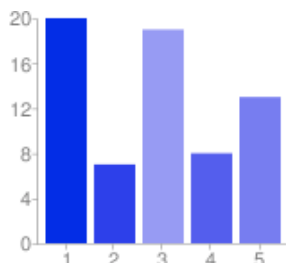


Figura 12. Integración de redes sociales (1 poco interesante, 5 muy interesante).

El 30% de los alumnos encuentran muy poco interesante (1) la integración de conocidas redes sociales. La preguntas 5 y 6 han mostrado un resultado que rechaza una de las hipótesis de partida del proyecto. El conjunto de respuestas parece apuntar a que la integración con otras aplicaciones sociales como Facebook no es motivador para los alumnos. Para poder contrastar este aspecto, sin embargo, habría que realizar un estudio de las aplicaciones sociales empleadas por el conjunto de alumnos a fin de identificar si la falta de interés es generalizada o es debido a que se ha planteado una integración con aplicaciones que no son de uso generalizado.

7. ¿Has utilizado el aprendizaje basado en proyectos (ABP) en alguna otra asignatura?

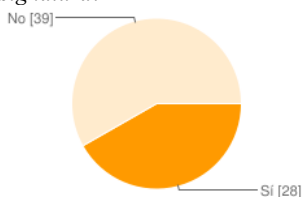


Figura 13. Cantidad de alumnos que conocían el enfoque ABP.

Para el 58% de los alumnos ésta ha sido la primera vez que utilizaban la metodología ABP. Este aspecto permite poner en relieve la importancia de que la herramienta se haya mostrado sencilla o muy sencilla para más del 80% de los alumnos (Ver Pregunta 1).

8. ¿Cómo valorarías los beneficios de usar OPERA en una asignatura con metodología ABP?

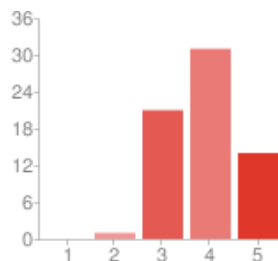


Figura 14. Beneficios de OPERA para ABP (1 poco beneficioso, 5 muy beneficioso).

El 67% opina que el uso de OPERA en una asignatura con enfoque ABP puede resultar bastante beneficioso (4-5). En este aspecto, cabe destacar que todos los alumnos tienen experiencia con el uso de WebCT como plataforma de ayuda al aprendizaje.

9. ¿Qué aspectos has encontrado más interesantes en OPERA?

La gran mayoría de respuestas recogidas hacen referencia a la comodidad y sencillez de uso, así como facilidad de acceso a proyectos anteriores.

10. ¿Qué aspectos has encontrado más negativos en OPERA?

Los comentarios negativos recogidos están relacionados, principalmente con errores puntuales de funcionalidad derivados de la fase preliminar del prototipo. A lo largo del cuatrimestre dichos errores se han subsanando en su totalidad.

De manera complementaria, siete profesores que imparten clase en las asignaturas que forman la experiencia piloto han realizado una encuesta similar enriquecida con dos nuevas cuestiones en las que se pregunta por el uso de otras herramientas y la comparación con ellas. A este respecto, los profesores valoran como “muy beneficioso” el uso de OPERA frente a otras herramientas a la hora de facilitar la puesta en marcha del ABP (la totalidad de docentes tenían experiencia en el uso de WebCT). En el resto de cuestiones, se han obtenido resultados muy similares a las aportadas por los alumnos.

5. Conclusiones

Desde el punto de vista de la aplicación de ABP, las herramientas existentes para su desarrollo tienen la necesidad de un alto grado de aprendizaje o configuración a la hora de usarlas, al ser genéricas y dar soporte a distintas estrategias pedagógicas. Esta problemática supone un incremento en la carga de gestión por parte del profesorado que disminuye la motivación a la hora de aplicar ABP. Además, las herramientas actuales están enfocadas al curso académico como contexto de aprendizaje de forma que no se permite la correlación entre el material y los proyectos generados de un curso a otro. La herramienta que presentamos consigue cubrir estas carencias con tres características principales: (i) Un sistema diseñado a medida y optimizado para ABP orientado a reducir al mínimo las necesidades de gestión por parte del profesorado; (ii) Se establece un escenario de “Red social” en el que se potencia la interacción de los alumnos y la integración con otras herramientas como Facebook o Twitter; (iii) Un mecanismo de retroalimentación de los resultados año tras año a través de un sistema de microportales públicos de proyectos que dotan de visibilidad a los mejores trabajos realizados cada año.

Hemos podido observar un grado de satisfacción bastante alto, tanto en profesores como en alumnos que han usado la herramienta en sus asignaturas. En el caso de los profesores, la creación de la aplicación ad-hoc, garantizaba de antemano los beneficios que obtendrían en su uso. Sin embargo, los alumnos también han valorado muy positivamente la aplicación, en especial la posibilidad de crear los microportales públicos.

Como trabajo futuro, planteamos una aplicación más allá de la Escuela Técnica Superior de Ingeniería Informática para comprobar la validez de la herramienta en el soporte de ABP en enseñanzas no técnicas [7]. Desde el punto de vista técnico nos gustaría estudiar la integración de OPERA con otras herramientas de amplio uso como Moodle. Desde un punto de vista metodológico, nos gustaría validar el uso de OPERA para dar soporte a escenarios de ABP complejos (similares al presentado en [2]). Por último, desde un punto de vista funcional, estamos estudiando la posibilidad de extender la herramienta para que se puedan integrar otros mecanismos de evaluación complementarios al ABP [4] y mecanismos de revisión por pares.

Agradecimientos

La puesta en marcha de OPERA ha sido posible gracias a la financiación del Vicerrectorado de Docencia de la Universidad de Sevilla concedida al proyecto de innovación docente. También, queremos agradecer a los técnicos responsables del desarrollo de la aplicación, Luis Miguel Ceballos Torres y Daniel Prieto Tagua, su arduo trabajo en el diseño, implementación y continua mejora de la herramienta, que han sido determinantes para lograr un sistema con un gran nivel de acabado.

Referencias

- [1] Albanese, M. A. & Mitchell S. *Problem-based Learning: A Review of Literature on its Outcomes and Implementation Issues*. Academic Medicine. 68 (1), 1993.
- [2] Del Canto, P. et al. . *Cómo congeniar los exámenes y los proyectos en asignaturas PBL*, Actas XII Jenui, 2006.
- [3] Design-based Research Collective. *Design-based research: an emerging paradigm for educational inquiry*. Educ. Resources, 2003
- [4] Díaz Fondón, M. A. et al. . *De la lección magistral al aprendizaje activo diseño de una actividad basada en PBL*, XIV Jenui, 2008
- [5] Dolmans, H. J. M. et al. *Problem-based learning: future challenges for educational practice and research*. Blackwell Publishing Ltd, Medical Education, 39, 2005
- [6] Estruch, V. et al. . *Aprendizaje basado en proyectos en la carrera de Ingeniería Informática*, Actas XII Jenui, 2006.
- [7] Gallego Duran, F. J. et al. . *¿Aprendizaje Basado en Proyectos? ¿Pero si mi carrera no es técnica!*, Actas XIII Jenui, 2007.
- [8] Labra Gallo, J. E. et al. . *Una Experiencia de aprendizaje basado en proyectos utilizando herramientas colaborativas de desarrollo de software libre*. Actas XII Jenui, 2006.
- [9] Moust, J. H. C. et al. *Signs of erosion: Reflections on three decades of problem-based learning at Maastricht University*, Higher Education, Springer, 2005
- [10] Reverte Bernabeu, J. R. et al. . *Aprendizaje Basado en Proyectos como modelo docente. Experiencia interdisciplinar y herramientas Groupware.*, Actas XIII Jenui, 2007.
- [11] Taboada, G. L. et al. . *Innovación docente en el EEES de cara a la práctica profesional a través del aprendizaje basado en proyectos*, Actas XVI Jenui, 2010.
- [12] Van den Bossche, P. et al. . *Does problem based learning educate problem solvers? A meta-analysis on the effects of problem based learning*, 7th EDINEB conference, 2000.

Experiencia con Webquest y herramientas Web 2.0 en la evaluación de competencias transversales

M^a Belén Vaquerizo García

Departamento de Ingeniería Civil, Escuela Politécnica Superior
Universidad de Burgos, belvagar@ubu.es

Resumen

Uno de los retos a los que nos enfrentamos en el EEES es el de usar la tecnología como herramienta para producir calidad en los resultados de enseñanza-aprendizaje e investigación.

En esta línea surgen nuevas metodologías y herramientas virtuales como apoyo a la evaluación de competencias transversales de las asignaturas.

Partiendo de nuevas metodologías docentes y protocolos de evaluación, así como de las nuevas herramientas a nuestro alcance, el objetivo es analizar, modificar y adaptar la estrategia de enseñanza-aprendizaje. Con ello se lograría optimizar la relación entre las competencias definidas para las asignaturas y las metodologías docentes y de evaluación propuestas para alcanzarlas.

El presente trabajo aporta una experiencia del uso de una Webquest y alguna herramienta Web 2.0 en una asignatura como estrategia didáctica y evaluativa de las competencias transversales.

1. Introducción

En este trabajo se muestra una experiencia del uso de una Webquest (en adelante WQ) como una metodología de aprendizaje cooperativo basada en el acceso y uso de información de la red, considerándolo así de utilidad en la adaptación al EEES. A través de ella se proporciona a los alumnos, de una manera bien estructurada y guiada, la tarea a realizar, junto con el proceso a seguir y los recursos a utilizar para su consecución.

Por tanto, se aporta una experiencia para el desarrollo y evaluación de las prácticas de una asignatura realizadas en grupo mediante el uso de esta estrategia didáctica [4,15,16], donde además se utilizan otras dos herramientas: Un Blog y una Wiki.

Ello supone una experiencia dentro de las estrategias de innovación en los métodos de

enseñanza [7,11,12], y que intenta ser coherente con el nuevo escenario del EEES en el que nos situamos [8], donde al alumno se le requieren un elevado número de horas de aprendizaje autónomo y con una participación mas activa tanto a nivel individual como grupal.

Ello requiere, por otra parte, de una utilización didáctica que se adecue a los objetivos perseguidos por el docente.

El presente trabajo se estructura del siguiente modo. A continuación, se describe en que consiste un WQ, y alguna herramienta con la que poder construirla. En el apartado 3, se describe la aplicación de esta estrategia docente en una asignatura. En el apartado 4, se muestra como se ha realizado el proceso de evaluación del alumno con la WQ. En el apartado 5, se comenta la experiencia obtenida con su uso, y se finaliza con las conclusiones obtenidas al respecto.

2. Webquest

Una WQ se puede considerar como un modelo de aprendizaje que promueve y facilita el uso de Internet y de las tecnologías de la información desde una perspectiva educativa y colaborativa [1,2,3]. Esta estrategia didáctica consta de 6 partes esenciales [5], las cuales se muestran en la siguiente figura:

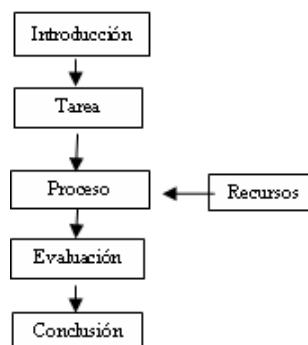


Figura 1: Etapas de la WebQuest

A continuación, se muestra una descripción de estas partes esenciales de la WQ:

1. *La Introducción.* La introducción presenta y describe a los alumnos, de una manera breve, clara y motivadora, un problema a resolver y sobre el que trabajar. Las estructuras de la introducción han de tener en cuenta el conocimiento anterior del alumno para lograr un aprendizaje cognoscitivo. En esta orientación a los alumnos sobre el trabajo que han de realizar se puede también establecer el rol que ha de seguir el alumno.
2. *La Tarea.* Aquí ya se definen claramente las tareas a realizar así como los distintos roles que tendrán los alumnos.
3. *El Proceso.* Se describen detalladamente las fases de las que constan las tareas a realizar. Para ello, se pueden usar mapas conceptuales, gráficas, tablas o cualquier otra estructura organizativa que ayude a presentar la trazabilidad y relación de las etapas. Además, todo ello se ha organizar de una manera clara, coherente y bien secuenciada. Se pueden aportar estrategias necesarias para dividir las tareas en subtarear, y describir los diversos roles, perspectivas y/o responsabilidades de los alumnos para lograr la tarea. Como ayuda para la realización de la tarea se plantea el uso de un Blog y una Wiki, consideradas como herramientas de ayuda para lograr información actual, interactividad y múltiples puntos de vista entre los alumnos.
4. *Los Recursos.* Se aportan recursos que permitan conseguir la información necesaria para que los alumnos logren la tarea. Debe haber una conexión clara y significativa entre todos los recursos. Estos recursos consisten en una lista de enlaces a páginas Web, referencias bibliográficas, etc, preparadas por el profesor para ayudar al alumno a completar la tarea y construir la WQ.
5. *La Evaluación.* Se describe a los alumnos cómo se evaluará su trabajo final. Para ello, se precisa de un modelo de evaluación donde figuren claramente los aspectos a evaluar y los criterios de evaluación para cada uno de ellos. Estos criterios pueden incluir descriptores cualitativos y cuantitativos, para ayudar a

medir lo que deben saber los alumnos y lo que deben hacer para lograr la tarea.

6. *La Conclusión.* Es la puesta en común de lo que se ha realizado. Los alumnos pueden sugerir alternativas o líneas futuras para mejorar su tarea, tratando con ello de impulsarles a la reflexión sobre lo aprendido. Debe estar estrechamente relacionada con la introducción y con todo el proceso de aprendizaje. Se podrían exponer, de manera concisa, las tareas cognitivas de los alumnos.

3. Desarrollo de la WebQuest

En esta sección se describe la experiencia del uso de la WQ como actividad didáctica de ayuda al aprendizaje cooperativo.

3.1 Herramientas para desarrollar la WQ

Existen, en la actualidad, varias herramientas que facilitan la creación y posterior publicación de WQ en Internet. Una vez creado se alojará en un servidor para que pueda ser visitado por otros usuarios. Entre estas herramientas se encuentran las dos siguientes:

- **PHP WQ:** Es una herramienta gratuita [15], para elaborar y publicar WQ muy fácilmente. Es un programa educativo pensado para realizar WQ, Miniquest y Cazas del Tesoro sin necesidad de escribir código HTML o usar programas de edición de páginas Web. El usuario puede también editar o borrar las actividades creadas por él.
- **INSTANT WQ:** Está disponible sólo en inglés. Permite elaborar y publicar en línea WQ, de forma rápida y gratuita. Esta herramienta genera, a partir de la información que introduce el docente, todos los archivos necesarios y los aloja automáticamente en un servidor Web gratuito.

Y, además, existen varias alternativas para usar la WQ. Por una parte, dentro de las posibilidades de Moodle hay más de un tipo de módulo WQ. Por otra parte, para crear y compartir fácilmente un sitio web de grupo se puede utilizar: Google Sites, una Wiki, Google Docs, etc,

Una vez decidido cómo hacer la WQ, lo siguiente es hacer el diseño educativo para lograr que la actividad sirva para el aprendizaje que se pretende [9,10].

3.2 Diseño educativo para trabajar con la WQ en la asignatura

Mediante el uso de la WQ se pretende que los alumnos transformen la información en conocimiento, y, en esta línea, les ayude a relacionar los contenidos de la materia con su futura actividad profesional. Para ello, la idea es proponerles tareas relacionadas con estas actividades profesionales, de un cierto grado de complejidad pero adecuadas a su nivel, y con un equilibrio entre el tiempo de dedicación a esa tarea y su valor en la evaluación.

Las tareas se proponen en equipo, fomentando así el trabajo colaborativo.

Para ello, se les requiere documentarse, analizar la información recopilada y reelaborarla, adquiriendo con ello destrezas profesionales propias de la disciplina en la que se encuentra.

La experiencia se ha realizado para la impartición de las prácticas de una asignatura de Ingeniería del Software, concretamente la asignatura de Análisis e Ingeniería del Software, troncal, de 9 créditos teóricos y 3 prácticos, impartida en el 2º Curso de I.T.I.G, con un número medio anual de 110 alumnos matriculados. Se ha llevado a cabo exclusivamente con los alumnos que, por diversos motivos, tenían problemas de asistencia a clase, siendo aproximadamente 40 alumnos.

Para ello, se organizaron grupos de 4 alumnos y en ellos se asignaron roles. Por cada grupo se creó un espacio de trabajo con el enlace a la WQ y a los materiales previstos que puedan necesitar para realizar la tarea.

En consecuencia, la WQ se ajustó al programa de la asignatura y a sus objetivos didácticos [13,14], y ello requirió de una clara identificación de los contenidos temáticos a cubrir. Y, todo ello en concordancia con la metodología, la didáctica y la forma de evaluación establecida previamente.

En este sentido, todo esto se ha reflejado conforme a las secciones del WQ, indicado en un documento .doc con diferentes apartados.

En primer lugar, se les muestra una introducción que les motive y les vincule con la realidad de ese problema.

A continuación, se definen las tareas a llevar a cabo, siendo muy importante enfocar lo que deben aprender los alumnos a las posibilidades que ofrece Internet para el aprendizaje. En cada tarea habrá un alumno moderador, y se les

orienta a que miembros de grupos distintos pero que han trabajado la misma parte se reúnan para aclarar sus dudas y profundizar sobre el tema de estudio.

El siguiente paso sería establecer el proceso de desarrollo. En cuanto a la elaboración de la WQ, se enumeran los pasos a seguir en la elaboración de misma. Y, además se les pide la creación de un Blog y de una Wiki, propiciando, con ello, a que los alumnos lo realicen en grupo.

Una vez establecido el proceso, se les aporta una lista de recursos que pueden usar para su realización (direcciones Web, bibliografía, etc). Y, por cada recurso aportado se aporta una pequeña aclaración de lo que se van a encontrar en él.

A continuación se concreta el método de evaluación mediante una rúbrica para que el alumno conozca de antemano los criterios de evaluación y pueda plantearse su propio aprendizaje, cumpliendo con su responsabilidad personal. Además, se les indica los aspectos y criterios a evaluar tanto a nivel individual como de grupo, incluso también a nivel de permitir su propia autoevaluación.

Y, por último, se les pide que ideen una conclusión a modo de cierre del trabajo.

Como parte del trabajo a realizar se elaboraron una serie de problemas diferentes a resolver, en la línea de experimentar diferentes formas de aprendizaje, relacionados con el análisis y diseño en el entorno O.O (materia abordada previamente en clase de teoría), y estos, bien se distribuyen aleatoriamente entre los grupos de trabajo, o bien los grupos eligen un problema de entre varios propuestos. Finalmente, todos los grupos sobre el tema elegido crean un documento explicativo, unas transparencias y un video.

Además de esos problemas, a todos los grupos se les exige de igual manera un modelado en O.O. mediante una herramienta Case de libre distribución, a elegir por ellos. Así, se fomenta que el alumno desarrolle sus competencias profesionales mas allá del trabajo sobre el tema que pretende desarrollar

Los diferentes aprendizajes en los que se pretende experimentar son: Aprendizaje basado en casos, aprendizaje basado en problemas, aprendizaje colaborativo, aprendizaje basado en proyectos, y herramientas Web 2.0 (por el uso de un Blog y una Wiki). Y, en esta línea, los

problemas a resolver se elaboran del siguiente modo:

1. Aprendizaje basado en casos: Al alumno se le plantea, en el apartado de Tarea, trabajar con un caso real que se pueda encontrar en su vida profesional, teniendo que efectuar una búsqueda de información para decidir como plantear su resolución. A partir del caso expuesto, la tarea pide buscar ciertas informaciones sobre temas relacionados con ese caso para ayudar a preparar argumentos para debatirlo con el resto de alumnos. Se ha de hacer un informe final. Así, el trabajo por medio de un caso contextualiza los contenidos y permite un aprendizaje situado.
2. Aprendizaje basado en problemas: En el apartado de Tarea se introduce un tema a investigar, y se reparte el trabajo por equipos con el fin de responder a unas cuestiones planteadas. El alumno puede revisar distintas fuentes propuestas, preparar una exposición y entregar un informe final exponiendo las conclusiones al respecto. Con ello, el alumno puede construir su propio conocimiento respondiendo a unas cuestiones básicas sobre una temática. Para ello, se han de aportar recursos básicos en el apartado de recursos y recursos mas avanzados en el apartado de proceso.
3. Aprendizaje colaborativo: Al alumno se le introduce en una temática y se le propone realizar un análisis e investigación sobre ello. En la tarea se indica que hagan una presentación y voten en clase el grupo que mejor lo ha hecho, y éste lo expondrá al resto de grupos. El trabajar en grupo y el explicar su trabajo al resto de compañeros, permite construir una pequeña comunidad de aprendizaje en torno al tema propuesto. En el apartado proceso se ha de fijar que el profesor es el orientador. Con ello, el alumno aprende a construir su propio conocimiento de forma progresiva, y esto le fomenta la toma de decisiones, algo útil para su futura vida laboral.
4. Aprendizaje basado en Proyectos: A los alumnos se les proporciona la información necesaria para elaborar un proyecto que responda a unos requisitos. Aquí es muy importante todo lo que se describa en el apartado de proceso. El alumno va a elaborar un proyecto siguiendo los pasos

Sesión 5: Métodos pedagógicos innovadores

que describe el entorno, y además ha de revisar los recursos que se proponen. Con ello se logra un buen trabajo global.

5. Empleo de un Blog y una Wiki: Se ha realizado una experiencia de aprendizaje colaborativo, donde todos los alumnos puedan participar activamente construyendo el conocimiento entre todos.

3.3 Utilización de un Blog y una Wiki

En la línea de cuidar la interacción grupal y un ámbito de intercambio de información entre alumnos de un grupo, a pesar de la no presencia física, cada grupo de alumnos se crea su propio Blog y Wiki que les permita relacionarse entre ellos. De este modo se facilita la participación, a nivel de cada grupo de alumnos, sin requerirles conocimiento entre ellos.

Mediante estas herramientas se establecen puntos de encuentro entre alumnos y profesor a nivel de cada parte del problema a resolver, y donde el tutor podrá ir respondiendo a las preguntas que los integrantes del grupo realicen sobre el material que se vaya compartiendo. De este modo, cada alumno puede expresar su propia visión sobre el tema que se plantee en cada oportunidad.

Además, se les indica una disciplina de trabajo mediante la cual y siguiendo un calendario establecido por el profesor, han de ir haciendo entregas de cada una de las partes de las que consta el problema a resolver.

Se procura que, aunque los problemas a resolver pueden tener enunciados diferentes, todos tengan un parecido grado de complejidad en su resolución.

4. Evaluación en la WebQuest

Uno de los aspectos mas importantes de la WQ es la evaluación, y ello requiere de un esfuerzo en elaborar una rúbrica con buenos descriptores y valoraciones asociadas a cada tarea.

Los criterios evaluables, tanto a nivel de grupo como individual se valoran con 3 etiquetas: Bajo, Medio y Alto, y son los siguientes:

Complejidad de las Tareas, Aportación de nuevos datos para su resolución (diagramas, ilustraciones, etc), Cantidad de Información, Calidad de Información, Organización de la Información, Documentación de las fuentes, gráficas, referencias, páginas Web o bibliografía utilizada, Análisis, Esfuerzo, Trabajo en Grupo,

Trabajo de manera ordenada, Presentación del trabajo, Uso de Recursos de Internet, Uso de otros recursos de forma acertada, Gramática correcta en la presentación (redacción).

En base a estos criterios y aspectos evaluables, a continuación se muestra un ejemplo de calificación a nivel de grupo de alumnos, y otro de calificación a nivel individual.

Ejemplos de calificaciones a nivel de grupo:

- Completaron todas las actividades aportando datos nuevos.
- Completaron las tareas y trabajaron en común realizando el trabajo.
- Completaron la mayoría de las tareas solicitando ayuda.
- Completaron algunas tareas en forma desordenada.
- No completaron la mayoría de las tareas. Trabajaron desordenados.

Ejemplos de Calificaciones individuales:

- Excelente presentación. Ha utilizado recursos de Internet y otros en forma acertada. Realizó textos coherentes sin errores de ortografía.
- Buena presentación del trabajo. Utilizó los recursos propuestos. Presenta alguna incoherencia gramatical.
- Presentó el trabajo de forma simple.

5. Experiencia obtenida

Con este planteamiento didáctico la enseñanza no se ha centrado en impartir contenidos, sino en utilizar esos contenidos para ayudar al alumno en el proceso del desarrollo de sus competencias personales y profesionales.

Se ha realizado una experiencia de aprendizaje colaborativo; los alumnos han trabajado aprendiendo entre ellos y construyendo el conocimiento entre todos y han participado de manera activa. Se ha logrado la implicación de todos los alumnos en su propio proceso de aprendizaje a través de un objetivo común, por tanto se ha conseguido el objetivo marcado.

Los resultados obtenidos permiten considerar esta estrategia como un buen recurso educativo. Estos resultados se han medido en dos vertientes, una es en cuanto al número de alumnos que han logrado superar las prácticas de la asignatura en comparación con cursos

anteriores, y, la otra vertiente es en base a unas encuestas de satisfacción realizadas a los alumnos, donde han dejado constancia de que este modelo de aprendizaje les ha facilitado la adquisición de los conocimientos. Por tanto, con ello se ha favorecido el aprendizaje del alumno. Y, por otra parte, permite aumentar el grado de motivación de los alumnos y su nivel de cooperación.

Además, también se ha favorecido el proceso de evaluación, ya que se proporciona mucha información de la actividad realizada por los alumnos. También hay que tener en cuenta que permite la evaluación de los alumnos tanto de manera colectiva como individual, así como la posibilidad de que los mismos alumnos puedan autoevaluarse.

Por otra parte, cabe indicar que supone un esfuerzo el perseguir el objetivo de que los alumnos aprendan, y el evaluar el grado de aprendizaje de los conocimientos y habilidades que se intenta enseñarles. En este sentido, el mayor esfuerzo ha sido preparar el diseño educativo, es decir, hacer que la actividad sirva para el aprendizaje que se pretende.

En esta línea, se puede decir que el planteamiento de la WQ en la asignatura ha supuesto lo siguiente:

- Cambiar el proceso de planificación: Se pasa de la unidad didáctica a la estructura WQ. En esta adaptación, los objetivos de una unidad didáctica equivaldrían a la introducción y a las conclusiones, los contenidos serían el proceso y los recursos, las actividades serían la tarea y el proceso, y la evaluación se haría mediante una rúbrica.
- Cambiar la metodología y la pedagogía: Se cambia el diseño de la enseñanza-aprendizaje y los planteamientos y roles del docente y del alumno.

Y, por otra parte, también se puede decir que el planteamiento de diferentes técnicas cooperativas, ha supuesto lo siguiente:

1. Diseñar una actividad cooperativa y valorar algunas de sus dificultades.
2. Identificar las limitaciones del aprendizaje tradicional, y, en base a ello, se han estudiado las características generales de las diferentes formas de aprendizaje cooperativo.
3. Relacionar las técnicas de aprendizaje cooperativo utilizadas en la asignatura.

Y, por último, algo importante a destacar es que mediante el enfoque de las diferentes tareas que se han propuesto en la WQ, se han podido experimentar diferentes formas de aprendizaje: Aprendizaje basado en problemas, aprendizaje colaborativo, aprendizaje basado en proyectos, aprendizaje del estudio de casos y herramientas Web 2.0 (por el uso de un Blog y una Wiki).

No obstante, cabe indicar que en el aprendizaje basado en problemas, el hecho de repartir las tareas entre equipos distintos exige que el profesor tenga un trato más individualizado hacia ellos en la resolución de las dudas que planteen.

Por todo ello, se podría justificar su uso en el contexto educativo, no solo bajo la perspectiva de los docentes sino también de los alumnos, como se indica a continuación de una manera más detallada.

Por otro lado, hay que tener en cuenta que la experiencia se ha realizado solo con una parte de los alumnos de la asignatura, y viendo el esfuerzo que al profesor le supone el control de los alumnos y las tareas, en principio no se plantea su uso con grupos mayores.

5.1 Desde el punto de vista de los alumnos:

Desde el punto de vista de los alumnos, ello conlleva los siguientes puntos favorables:

- La WQ se adecúa a los requisitos de los ECTS, para favorecer el desarrollo de sus competencias laborales.
- El alumno pasa de ser receptor de la información a recopilar información transformándolo en conocimiento significativo. Por tanto, además de orientar a los alumnos a una búsqueda de información, se favorece el desarrollo de sus competencias profesionales y habilidades [4,6] tales como su creatividad, pensamiento crítico, toma de decisiones, capacidad de procesamiento y transformación de los conocimientos adquiridos en la red, etc.
- El alumno ha de planificar su tiempo y organizar y orientar su trabajo en grupo, intercambiando información entre los componentes del grupo.
- El alumno utiliza más Internet para buscar herramientas, recursos e información. Otro aspecto derivado de ello es que el alumno

Sesión 5: Métodos pedagógicos innovadores

aprende a ser crítico respecto de la información que encuentra en Internet

- Fomenta la participación e implicación de alumnos que normalmente no participaban, con lo que favorece la participación de todos ellos.

Por consiguiente, se espera que el uso de esta estrategia didáctica, por una parte ayude al alumno a comprender la importancia del conocimiento de la materia para su futuro profesional, y, por otra parte, contribuya a desarrollar en los alumnos diferentes capacidades, tales como: aprender a analizar información, abstraer, comparar, clasificar, deducir, analizar errores, construir un sistema de ayuda, etc.

A continuación se indican las competencias transversales que el alumno puede adquirir para esta asignatura concreta:

Algunas Competencias Transversales, tales como:

- Capacidad de análisis y síntesis.
- Capacidad de organización y planificación.
- Capacidad de gestión de la información.
- Resolución de problemas.
- Toma de decisiones.
- Trabajo en equipo.
- Trabajo en un equipo de carácter multidisciplinar.
- Habilidades en las relaciones interpersonales.
- Razonamiento crítico.
- Aprendizaje autónomo.
- Adaptación a nuevas situaciones.
- Creatividad. Iniciativa y espíritu emprendedor.
- Motivación por la calidad.
- Desarrollar habilidades de aprendizaje para emprender estudios posteriores con un alto grado de autonomía.
- Planificación y gestión del tiempo.

Y, algunas competencias Específicas, tales como:

- Diseñar, desarrollar, seleccionar y evaluar, aplicaciones y sistemas informáticos, asegurando su fiabilidad, seguridad y calidad, conforme a los principios éticos y a la legislación y normativa vigente
- Valorar las necesidades del cliente y especificar los requisitos software para satisfacer estas necesidades

Y, es, en concreto, por estos objetivos que se consideran logrados de la asignatura, por lo que se justifica el uso de esta herramienta en la impartición docente de la misma.

5.2 Desde el punto de vista del profesor:

El profesor ha de planificar cómo los alumnos han de ir construyendo para ellos mismos el conocimiento.

Para ello, ha de estructurar los contenidos basándose en tareas reales que el alumno ha de resolver. Esto le exige un notable esfuerzo, pues se requiere que la WQ esté bien planificada, con propuestas bien elaboradas y con recursos significativos.

Por otra parte, la WQ le sirve de ayuda al profesor para planificar y estructurar la enseñanza de la asignatura de una manera creativa, con una mayor concreción de las tareas a realizar. Y, hay que tener en cuenta, que el tiempo y esfuerzo dedicado a la creación de una WQ se puede amortizar con la reutilización sucesiva de sus contenidos educativos.

Por consiguiente, esta metodología puede ser usada en cualquier asignatura o titulación.

5.3 Expansión a otras asignaturas / competencias:

Cabe indicar que la experiencia aportada puede ser reproducida por otros docentes en otras asignaturas diferentes.

Se ha de considerar que aunque su estructura es simple y, además, presenta significativas equivalencias con las unidades didácticas en el diseño tradicional, ello requiere readaptar y plantear las tareas a realizar por el alumno de un modo diferente:

- De cara al alumno: Se trata de lograr un dinamismo y una organización que fomente el aprendizaje constructivo centrado en el alumno, ya que ayuda a sistematizarlo.
- De cara al profesor: Le ha de servir como ayuda en la planificación temporal y en la valoración del esfuerzo por parte del alumno.

Por otra parte, aunque, fundamentalmente, las WQ están orientadas para que los alumnos trabajen en grupo, su planteamiento también admite su diseño para trabajos individuales, y, además, pueden ser diseñadas bien para una

única asignatura o bien pueden ser interdisciplinares.

Conclusiones y líneas futuras

En este trabajo se presenta una experiencia de aprendizaje colaborativo con la que se ha pretendido innovar en los métodos de enseñanza aplicados al desarrollo de las prácticas de una asignatura concreta.

La experiencia se ha realizado principalmente mediante el uso de una WQ, donde, a través de sus apartados, se explica detallada y organizadamente como desarrollar la tarea planteada, y en su proceso se requiere del uso de un Blog y una Wiki.

Por tanto, esta experiencia aporta, frente a otros trabajos, la utilización de un Blog y una Wiki como herramientas Web 2.0. para ayudar en la etapa de Proceso de la WQ. Por ello, se considera una propuesta de metodología innovadora para su aplicación en la universidad.

Como líneas futuras a tener en cuenta, con el fin de hacer un mejor uso de la WQ, podría indicarse lo siguiente:

- Impartir una primera clase presencial al alumno donde se le introduzca en el uso de entornos virtuales para que pueda trabajar de forma autónoma, y concretamente en el uso de la WQ. Esto facilitaría que el alumno tome un rol activo y se responsabilice mejor de su propio aprendizaje.
- Preparar un buen guión explicativo de absolutamente todos los aspectos que va a tener la WQ, ya que muchas de las preguntas de los alumnos son sobre contenidos en la WQ que los alumnos no llegan a leer.
- Cuidar el número de las referencias que se les ofrece y valorar si el alumno aporta alguna nueva.
- Administrar un cuestionario al mes, que incluya algunas preguntas sobre el trabajo en grupo, y comentar, posteriormente, los resultados en clase. Esto se cree que favorecería a aquellos alumnos que no se estén integrando correctamente en su grupo.
- Dividir el trabajo en mayores temas parciales.
- Proponer supuestos prácticos a desarrollar lo más realistas posibles. Hay que tener en cuenta que cuanto más realista sea la actividad propuesta al alumno, mejor serán los recursos encontrados.

- Seleccionar una sola actividad en la que todos los alumnos, no organizados por grupos, puedan trabajar en la WQ.

Referencias

- [1] Alvarez, C., López, D., Jimenez-Gonzalez, D., Alonso, J. Aprendizaje cooperativo en cursos multidisciplinares. Jenui 2009.
- [2] Cernuda, R., A y Otros.. Análisis de los hábitos de trabajo autónomo de los alumnos de cara al sistema de créditos ECTS. Jenui 2005.
- [3] Cernuda, R, A., Labra G, J.E., Fernandez L, D., Calvo S, J. *Utilización de herramientas colaborativas de desarrollo de software libre en un modelo de aprendizaje basado en proyectos*. Jenui 2006.
- [4] Johnson, D.W., Johnson, R.T., Smith, K.A. *Active learning: Cooperative Learning in the college classroom*. Edina MN: Interaction Book Company, 1991.
- [5] Lara, S. *Eficacia de la WebQuest como una metodología acorde con el nuevo EEES: Una experiencia universitaria* Revista de Ciencias de la Educación, n°210, pp135-152 (abril-junio 2007)
- [6] Novak, J.D., Gowin, D.B. *Learning how to learn*. New York. Cambridge University Press. 1984.
- [7] Quintana, A., Higuera, E. Las Webquests, una metodología de aprendizaje cooperativo, basada en el acceso, el manejo y el uso de información de la red.
- [8] Sanchez C. F. *¿Cómo serán las asignaturas del EEES?*. Jenui 2005.
- [9] Santamaría J y Otros. Potenciando el aprendizaje proactivo con ILIAS&WebQuest: aprendiendo a paralelizar algoritmos con GPUs. Jenui 2010.
- [10] Santos R y Otros. La Wiki-Webquest: Una actividad colaborativa en la asignatura
- Sesión 5: Métodos pedagógicos innovadores**
- “Nuevas Tecnologías aplicadas a la Educación”. RED. Revista de Educación a Distancia. España. ISSN Electrónico: 1578-7680.
- [11] Vaquerizo-García, M.B., Renedo-Mena, E. *Herramientas para la Elaboración de Contenidos Didácticos en el Contexto e-Learning*. Jenui 2008.
- [12] Vaquerizo-García, M.B., Renedo-Mena, E. Valero M. *Aprendizaje Colaborativo en Grupo: Herramientas Web 2.0*. Jenui 2009.
- [13] *5 Reglas para escribir un buen WebQuest*.
<http://platea.pntic.mec.es/erodri1/TALLER.htm#reglas>
- [14] Cómo elaborar una Webquest de Calidad o Realmente Efectiva. <http://www.eduteka.org/modulos.php?catx=7&idSubX=225&ida=440&art=1>.
- [15] www.aprenderenred.net.
- [16] http://phpwebquest.org/?page_id=19

Experiencia educativa para fomentar el aprendizaje autónomo a través de preguntas tipo test generadas por los alumnos

José L. Risco-Martín, Marcos Sánchez-Élez, Inmaculada Pardines

Departamento de Arquitectura de Computadores y Automática

Universidad Complutense de Madrid

C/ Prof. José García Santesmases, s/n

28040 Madrid

jlrisco@dacya.ucm.es, marcos@fis.ucm.es, inmapl@dacya.ucm.es

Resumen

Describimos en este artículo una experiencia docente que trata de incentivar el proceso de aprendizaje autónomo del alumno mediante la realización de tests con preguntas generadas por ellos. La idea es que el alumno perciba que participa de una manera directa en su calificación, ya que sus preguntas pasarán a formar parte del examen. Un conjunto de herramientas permite la interacción alumno-profesor y alumno-alumno a través del campus virtual. En este sentido, profesores y alumnos pueden ver todas las preguntas formuladas por los otros compañeros. De esta manera incentivamos el análisis crítico (comprender la calidad de las preguntas ajenas). Hemos comprobado, comparando los resultados académicos de distintos años, que el 88% de los alumnos que han incorporado a su metodología de estudio estas estrategias más activas y participativas han conseguido aprobar la asignatura.

Summary

In this paper we describe a teaching experience that tries to motivate students following an autonomous learning process. To this end, we conduct tests with questions created by the students. The idea is that students perceive that they participate in their rating in a direct way, because their questions will become part of the final test. A set of tools allows us interaction student-teacher and student-student through the virtual campus. In this way, students and teachers can see all the generated questions. Thus, we motivate the critical analysis (how to understand the quality of questions). We found, comparing to academic results of different academic years, that the 88% of the students that have included to their learning methodology these active and participative strategies, have been able to pass the course.

Palabras clave

Enseñanza universitaria, uso de nuevas tecnologías en la enseñanza, metodologías de estudio, evaluación continua, e-learning.

1. Introducción

La puesta en marcha de los nuevos grados es una buena oportunidad para plantearse nuevas metodologías docentes y de evaluación en el proceso educativo con el fin de que el alumno adquiriera no sólo los conocimientos asignados a cada asignatura sino también capacidades y destrezas transversales comunes a los estudiantes universitarios tales como el análisis y síntesis de conceptos, la gestión adecuada de la información y la mejora de la comunicación escrita.

En este artículo se presenta una experiencia docente que intenta estimular la participación del alumno en el proceso enseñanza-aprendizaje siguiendo las pautas de cambio metodológico propuestas a raíz de la adaptación de las titulaciones al marco del Espacio Europeo de Educación Superior. Para ello hemos creado una herramienta que permite a los alumnos participar en el proceso de revisión de los conocimientos adquiridos. Proponemos dar la posibilidad al alumno de plantear cuestiones de respuesta corta (tipo test) sobre la asignatura. Con las preguntas recolectadas por el alumno y otras aportadas por el profesor, se configuran diferentes pruebas tipo test a lo largo del curso. Además como parte de la motivación intrínseca [1] al alumno, las calificaciones obtenidas en las pruebas tipo test se suman a la nota obtenida en el examen final.

La idea de esta propuesta surge ante la actitud pasiva que se observa en los últimos años por parte de los alumnos. Las nuevas tecnologías de la información y las comunicaciones permiten que las clases teóricas se impartan con los últimos avances tecno-

lógicos: diapositivas, audio, video, etc. información que además se ofrece vía web al alumno. Esto convierte al alumno en un sujeto pasivo de la asignatura. Cada vez toman menos notas en clase “y por lo tanto” los conocimientos adquiridos se olvidan con suma facilidad en un plazo relativamente corto, lo que deteriora la continuidad de la asignatura. La compañía Apple Inc. se ha dado cuenta del problema de actitud pasiva por parte del alumno y actualmente presenta convocatorias en innovación educativa, buscando que los alumnos participen de forma más activa en las clases [2].

El objetivo principal de esta experiencia docente ha sido integrar un conjunto de herramientas que a través del campus virtual de la Universidad Complutense de Madrid permita desarrollar una metodología de aprendizaje autónomo a través de preguntas de test generadas por los alumnos. El resultado de los tests, como ya hemos indicado, forma parte de la calificación final de la asignatura. Con esto se consigue motivar al alumno a la participación. Como resultado hemos integrado varias herramientas (una de ellas desarrollada por nosotros), compatible con cualquiera de las plataformas del campus virtual y que permite a los docentes: (1) evaluar de forma cualitativa el nivel de comprensión que ha adquirido el alumno y su capacidad para expresarse a través de su habilidad para proponer preguntas sobre la materia objeto de estudio, y (2) generar automáticamente una prueba de evaluación para calificar de forma cuantitativa los conocimientos adquiridos por el alumno.

Indirectamente estamos fomentando en el alumno la capacidad de gestionar adecuadamente la información disponible, distinguiendo lo importante de lo accesorio. El procedimiento invita a la participación no sólo a los alumnos avanzados sino también a los alumnos con menos base ya que en ningún momento nos planteamos calificar la calidad de las preguntas subidas, sino fomentar la generación de esas preguntas por parte de los alumnos.

Por otro lado, contemplamos el estudio de la eficacia del método, comparando el rendimiento académico de los alumnos que han participado con el de aquellos que no lo han hecho. Es nuestra intención aplicar la metodología propuesta en varias asignaturas de distintas disciplinas en siguientes cursos académicos.

2. Trabajo previo

En la actualidad se están produciendo grandes avances en el campo de los denominados *Learning Management Systems (LMS)*, con las contribuciones de grupos como el IMS Global Learning Consortium y sus encaminados estándares como su especificación para cuestionarios y tests *Question & Test Interoperability Specification (QTI)* [3]. En este sentido existe una gran variedad de herramientas, gratuitas o no, que ofrecen múltiples facilidades para la gestión de contenidos y creación de exámenes. Algunos ejemplos son WebCT [4], Sakai [5], OLAT [6], dotLRN [7], y Moodle [8]. De hecho, muchas Universidades tienen implantado alguna de las herramientas anteriormente citadas como base de su campus virtual. Sin embargo, ninguna de ellas admite por defecto la creación de preguntas por parte del alumno, que es lo que se persigue siguiendo la metodología propuesta.

Para facilitar la evaluación de los alumnos y dotar de mayor interacción alumno-profesor se han creado algunas herramientas específicas de evaluación en línea, como por ejemplo iTest [9] o MegaTest [10]. No obstante, estas herramientas siguen sin proporcionar facilidades para que los alumnos puedan subir preguntas para su posterior corrección por el profesor. Sí que hay algunas herramientas comerciales para mantenimiento de bases de datos de problemas, como por ejemplo cramster.com [11], o texercises.com [12]. En ambos casos los estudiantes pueden aportar problemas y soluciones en un formato especial (flash o texto, en Cramster, y latex, en Texercises). Posteriormente, los profesores o expertos pueden corregir las soluciones (en ambas plataformas) o evaluar la calidad de los problemas (en el caso de Texercises). No obstante, estas herramientas, además de ser de pago, son demasiado genéricas para el propósito de la metodología propuesta, que consiste fundamentalmente en que el alumno pueda enviar preguntas que posteriormente puedan ser incluidas en un examen.

Hay que recalcar que el objetivo de la experiencia propuesta no es competir con las herramientas mencionadas anteriormente, sino mostrar que se puede conseguir una mayor participación del alumno en el proceso de aprendizaje y aumentar así el grado de atención en el aula a lo largo de todo el curso académico. Conseguimos con ello una mayor asimilación de conocimientos, tal como se deriva de los resultados experimentales.

3. Metodología

En esta sección desarrollamos la metodología aplicada, así como las distintas decisiones que nos llevaron a la misma.

3.1. Recolección de preguntas

Tal como se ha mencionado en la Sección 2, existe un amplio abanico de herramientas de e-learning disponibles. Sin embargo, ninguna de ellas permite que el usuario con rol de alumno pueda incorporar sus propias preguntas. Para solucionar este problema nos propusimos varias alternativas:

- *Que el alumno configure sus preguntas en un archivo de texto plano:* En este caso creábamos una actividad en el campus virtual (recogida de fichero). El alumno subía entonces un archivo de texto plano indicando las preguntas, respuestas, y entre las respuestas la que era correcta. Esta alternativa generaba diversas dificultades. La primera, y más obvia, es que los alumnos no seguían las indicaciones del profesor para el formato de las preguntas. Por lo que el profesor, en el momento de recolectar todas las preguntas, tenía que dedicar una gran cantidad de tiempo en corregir y unificar archivos para la posterior publicación de las mismas en el campus virtual.

- *Que el alumno configure sus preguntas en un archivo con formato GIFT:* El formato GIFT, desarrollado por la comunidad Moodle, permite usar cualquier editor de texto para escribir preguntas de opción múltiple. Con una sintaxis muy sencilla, la principal ventaja de GIFT es que un banco de preguntas en este formato se puede publicar automáticamente en el campus virtual (en nuestro caso basado en Moodle). Aparecieron sin embargo dos inconvenientes: (1) el alumno volvía a cometer numerosos errores al seguir el formato GIFT que tenía que corregir (de nuevo) el profesor, y (2) el formato GIFT, aunque ampliamente aceptado en la comunidad Moodle, no lo es tanto en WebCT o Sakai.

- *Que el alumno configure sus preguntas usando Hot Potatoes:* Esta fue la solución definitiva. Por un lado, el alumno no cometía errores de formato, ya que Hot Potatoes proporciona una interfaz gráfica (ver Figura 1). Además, casi todas las herramientas de e-learning incluyen aplicaciones (propias o de terceros) para convertir el formato de Hot Potatoes a su propio formato (y viceversa). De esta manera,

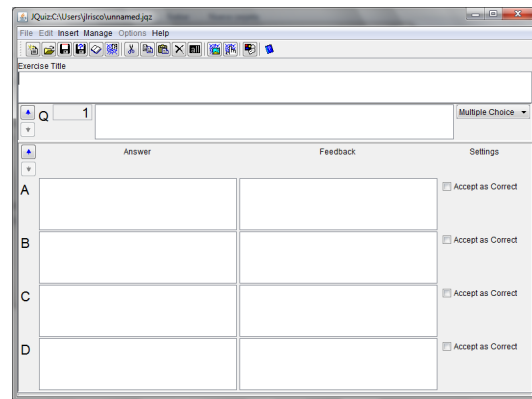


Figura 1: Interfaz gráfica de Hot Potatoes para insertar preguntas

el profesor veía reducida su tarea a unir los distintos archivos recibidos de los alumnos.

3.2. Publicación de preguntas

La siguiente fase consiste en publicar las preguntas aportadas por los alumnos en el campus virtual. Esta fase se realiza varios días antes de la realización del test definitivo. De esta forma, los alumnos pueden practicar con las preguntas subidas por sus compañeros y repasar temas.

Una vez recolectados todos los archivos de los alumnos en la fase anterior, el profesor tiene que unirlos en un único archivo de Hot Potatoes. A partir de aquí consideramos dos opciones: (1) importar desde el campus virtual todas las preguntas o (2), publicar el banco de preguntas como un test en formato HTML. Para ello utilizamos la opción de Hot Potatoes *Make a Standard Exercise*, que genera un archivo HTML que después se puede publicar en el campus virtual. Utilizamos (2) por su sencillez, dado que en este punto no es necesario evaluar los resultados de los alumnos, simplemente deseamos que los alumnos puedan practicar (y a su vez aprender) utilizando las preguntas subidas por sus compañeros. La idea es que el alumno perciba que participa de una manera directa en su calificación viendo que sus preguntas pasan a formar parte de las posibles preguntas de examen. Es también interesante que los alumnos puedan ver todas las preguntas formuladas por los otros compañeros, porque de esa manera se incentiva el análisis crítico (comprender

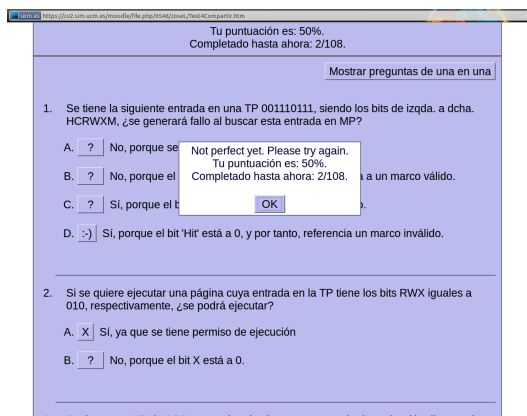


Figura 2: Banco de preguntas publicado como recurso HTML en el campus virtual

la calidad de las preguntas ajenas). También se fomenta la gestión adecuada de la información, ya que dependiendo de la cantidad de preguntas propuestas por los compañeros el alumno se puede ver obligado a aplicar alguna estrategia selectiva que le permita un mejor aprovechamiento de la información disponible. Desde la recepción de preguntas hasta la fecha del test en aula se debe disponer de varios días para ir corrigiendo los errores cometidos por los alumnos y configurar el test definitivo.

La Figura 2 muestra el banco de preguntas publicado en el campus virtual como recurso HTML. En la figura se puede apreciar que el total de preguntas recibidas es 108, y que el usuario ha respondido correctamente la primera pregunta y está fallando en la segunda. En este caso se puede responder múltiples veces, aunque esta característica (y otras) se pueden editar desde Hot Potatoes al generar el test.

3.3. Realización del test en el aula

La fase final del método propuesto se basa en la realización del test en aula. Una vez corregidas las preguntas de los alumnos, el profesor inserta algunas preguntas propias en la base de datos. En esta fase se pueden eliminar de forma manual algunas preguntas formuladas por los alumnos, atendiendo al control de calidad de las mismas, su variedad y la dificultad final. Además, tras estudiar la complejidad del banco de preguntas final, evalúa cuántas preguntas debería contener el test final (la duración de la prueba será

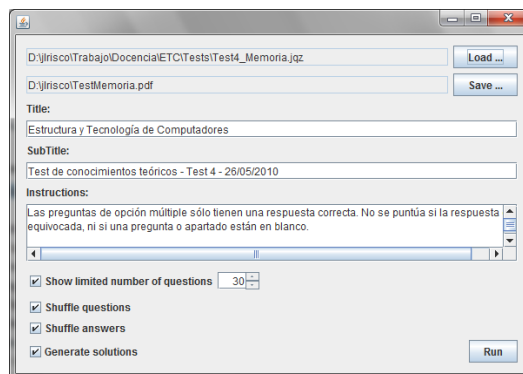


Figura 3: Generador de tests en formato PDF.

inferior a 50 minutos, que es el tiempo que disponemos para una clase). El procedimiento habitual consiste en elaborar un test de unas 30 a 40 preguntas. Para ello, se seleccionan las preguntas de forma totalmente aleatoria. En este sentido, es probable que el test incluya una pregunta del profesor que no se haya hecho pública en HTML (“probable” salvando proporciones de preguntas alumno/profesor en la base de datos final).

Uno de los problemas encontrados fue la gran cantidad de tiempo requerida para seleccionar preguntas aleatorias y (principalmente) darle formato al test con el editor de textos deseado. Para ahorrar tiempo decidimos desarrollar una herramienta que, dado un banco de preguntas en formato Hot Potatoes, generara un archivo en formato *Portable Document Format (PDF)*.

La Figura 3 muestra una captura de pantalla de la herramienta desarrollada. En primer lugar, se selecciona el banco de preguntas del que se quiere generar el test (Test4_Memoria.jqz en la Figura). Después, se elige el archivo en formato PDF donde se quiere guardar el test generado. Además, como ilustra la Figura 3, permite especificar un título para el test, así como un subtítulo e instrucciones para el mismo. Finalmente, se da la opción de generar un subconjunto de preguntas (30 en este caso), la opción de barajar las preguntas y las respuestas dentro de cada pregunta, y la de generar al final del documento una tabla con las respuestas correctas del test.

Antes de desarrollar esta herramienta comprobamos que desde el campus virtual, y dado un banco de preguntas, no se permite generar tests en un for-

mato “amigable” para su posterior resolución en papel. Esta herramienta nos ahorró mucho tiempo en la edición y maquetación de los tests.

4. Resultados

Durante el curso académico 2009-2010, se aplicó la metodología propuesta en el segundo cuatrimestre de la asignatura de Estructura y Tecnología de Computadores, asignatura troncal de segundo curso de Ingeniería Técnica en Informática de Sistemas con una carga docente total de 15 créditos, 7.5 créditos por cuatrimestre.

Con el propósito de realizar un estudio detallado de la metodología, se llevó un control de asistencia a clase. Para ello al menos un día a la semana el profesor pasaba lista. A partir de este momento consideraremos que un estudiante no ha asistido a clase si no ha estado ningún día de clase en los que se pasó lista. En este sentido, de un total de 198 alumnos matriculados, la asistencia promedio fue de un 40%. Este dato junto con el porcentaje de estudiantes presentados al examen final (53%) viene a confirmar que un número importante de alumnos deja la asignatura (para septiembre o para el curso siguiente) aunque se hayan matriculado.

Los alumnos realizaron 4 tests repartidos a lo largo del segundo cuatrimestre siguiendo la metodología expuesta en la sección anterior. Cada uno puntuaba un máximo de 0,5 puntos. De los cuatro tests, se seleccionarían los tres con mejor puntuación, con lo cual cada alumno podría obtener una puntuación máxima de 1,5 puntos.

La calificación obtenida por el estudiante en los tests se sumó directamente a la nota del examen final que estaba puntuado sobre 10 puntos. De esta manera se pretendía motivar a los estudiantes a presentarse a las pruebas. Los estudiantes realizaron las cuatro pruebas en aula los días 24 de marzo, 29 de abril, 13 de mayo y 26 de mayo de 2010, respectivamente.

4.1. Preguntas aportadas por el estudiante

En esta sección estudiamos el número de preguntas aportadas por los estudiantes.

La Tabla 1 muestra los datos obtenidos en el proceso de aportación de preguntas, tanto para cada prueba como los datos totales. En dicha tabla, P es el

número total de preguntas subidas al campus virtual, P⁺ representa el número total de preguntas aportadas por los estudiantes que fueron al menos una vez a clase. En el mismo sentido, P⁺/A⁺ representa el número de preguntas enviadas por alumno que asistió alguna vez a clase.

Como se puede apreciar, el número total de preguntas P es creciente. Esto es debido principalmente a que los estudiantes se mostraron más participativos según avanzaba el curso y además a que a partir de la segunda prueba ya disponíamos de las herramientas mencionadas para automatizar el proceso, lo que hacía más sencilla la participación del estudiante y la gestión de las preguntas por parte del profesor. Cabe destacar que del total de preguntas enviadas, el 96,47% pertenecen a alumnos que han ido alguna vez a clase. De hecho, como veremos a continuación, participaron principalmente aquellos alumnos que asistieron con asiduidad a clase.

4.2. Resultados de las pruebas

En esta sección presentamos los resultados obtenidos tras la realización de las cuatro pruebas en aula. La Tabla 2 muestra la configuración de las pruebas finales realizadas.

El número de estudiantes que subieron preguntas no fue excesivamente alto, un 46,5 en promedio (ver Tabla 2), pero que representa un 59% de los alumnos que asistieron alguna vez a clase. Cabe resaltar que el número de alumnos que se presentaron a los tests es bastante alto superando en promedio el número de alumnos que asistieron alguna vez a clase (representa el 43,2% de los alumnos matriculados). Lo que nos indica: todos los alumnos que asistían a clase estaban interesados al menos en realizar los tests y algunos alumnos seguían la asignatura aunque no asistieran a clase por lo que se presentaban a los tests.

Como se puede apreciar en la misma tabla, el porcentaje de aprobados es alto (70,46% en promedio). Esto se debe fundamentalmente a tres razones: (1) una parte de las preguntas de cada prueba estaba constituida por preguntas enviadas por los alumnos, que no entrañaban mucha dificultad, (2) todo el banco de preguntas creado por los alumnos estaba disponible unos días antes de la realización de la prueba en aula y (3) el hecho de disponer del banco de preguntas y que éstas fueran enviadas por los alum-

#	P	P ⁺	%	P ⁺ /A ⁺
1	92	92	100,00 %	0,68
2	113	113	100,00 %	0,83
3	151	151	100,00 %	1,11
4	211	191	90,52 %	1,40
Total	567	547	96,47 %	4,02

Tabla 1: Envío de preguntas para cada prueba

	1	2	3	4	Promedio
Nº alumnos que subieron preguntas	43	45	47	51	46,5
Presentados	90	81	81	90	85,5
Aprobados	63	63	63	52	60,25
Nota media (sobre 10)	6,56	6,02	5,51	4,84	5,73

Tabla 2: Resultados obtenidos en las pruebas.

nos motivaba a los alumnos a prepararse los tests. Nos gustaría resaltar la tercera razón, ya que aunque algunas de las posibles preguntas estuvieran disponibles con anterioridad, éstas superaban la centena, por lo que el estudio del test requería a los alumnos tiempo extra de estudio en casa. Finalmente, la Tabla muestra las calificaciones promedio obtenidas en las pruebas (sobre 10). En todas las pruebas hay alumnos con calificación 10 y alumnos con calificación 0.

Durante el cuatrimestre de aplicación de este método, fuimos preguntando a los estudiantes sus impresiones al respecto. En general, todos los estudiantes estaban contentos con el método ya que obtenían puntuación extra en un formato de evaluación continua. Sin embargo, la mayoría de estudiantes proponía valorar también tanto las preguntas enviadas por los alumnos como la calidad de las mismas. Además, sugirieron dotar de mayor probabilidad de aparecer en la prueba aquellas preguntas con mayor calidad. Como era de esperar, las del profesor no deberían contar, porque estas últimas no eran publicadas con el resto unos días previos a la prueba.

Nosotros, con la realización de las pruebas, notamos un beneficio mucho mayor. Una de las características de la asignatura Estructura y Tecnología de Computadores es que de un tema a otro se van utilizando los conceptos de temas anteriores. Los profesores pudimos observar que al final del cuatrimestre, en el transcurso de los últimos temas, los estudiantes recordaban conceptos impartidos en los primeros

temas.

4.3. Resultado académico final

En esta sección mostramos las calificaciones finales de junio de 2010, comparando aquellos alumnos que realizaron las distintas pruebas con los que no. La Tabla 3 muestra la comparativa de alumnos presentados, y dentro de estos, el número de aprobados y su porcentaje. El examen final fue aprobado por 44 alumnos (nota igual o superior a 5). La siguiente columna de la Tabla 3 muestra el número de alumnos presentados a las pruebas y el número de alumnos que las aprobaron, dato ya comentado en la sección anterior. La columna *Fin. + Pru.* muestra los alumnos que se presentaron al examen final y que además habían realizado tres o más pruebas, requisito mínimo para que la nota de las pruebas se le sumara a la nota del examen final. Como se puede apreciar en la Tabla el 68.29% de los alumnos que se presentaron al examen final y aprobaron esta parte de la asignatura, habían realizado las pruebas. Finalmente, la columna *Fin. + Pru. + Preg.* muestra los alumnos que se presentaron al examen final, que hicieron al menos 3 tests y que subieron preguntas para alguno de los tests, obteniéndose un resultado muy satisfactorio: el 88% de estos alumnos aprobaron esta parte de la asignatura.

Observando de nuevo la Tabla 3, debemos mencionar que en la columna *Fin.+Pru.* hay más estudiantes aprobados que en la columna *Final* (56 vs. 44). Esto es debido a que ya se ha computado la no-

	Final	Pruebas	Fin.+Pru.	Fin.+Pru.+Preg.
Presentados	105	85,5	82	50
Aprobados	44	60,25	56	44
% Aprob.	41,90 %	70,47 %	68,29 %	88 %

Tabla 3: Resultados obtenidos en las pruebas.

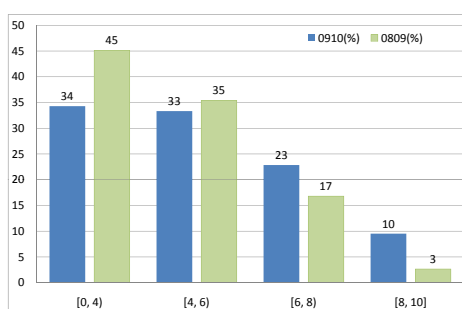


Figura 4: Comparación de notas finales en distintos cursos académicos

ta extra por la realización de los tests, es decir, hay 12 estudiantes con nota inferior a 5 pero superior a 4 que aprobaron la asignatura debido a la nota de los tests.

La Figura 4 muestra una comparativa entre las calificaciones del segundo cuatrimestre de la asignatura en el curso 2009-2010 (105 alumnos), en el que se ha aplicado esta metodología, con el curso 2008-2009 (113 alumnos), donde se realizaban los test pero sin participación de los alumnos en su elaboración. Se observa que al aplicar esta metodología el número de alumnos con notas buenas (superiores a 6) crece en un 65 % pasando del 20 % al 33 % de los alumnos presentados al examen.

La Figura 5 muestra una comparativa entre las calificaciones de los 105 alumnos del segundo cuatrimestre de la asignatura (A(%)) con el primer cuatrimestre de la misma (B(%)) en el curso 2009-2010. En el primer cuatrimestre se realizaban los test, pero sin participación de los alumnos en su elaboración. Se observa que al aplicar esta metodología el número de alumnos con notas buenas casi se triplica, pasando del 15 % al 41 % de los alumnos presentados al examen. Mientras que se reduce a menos de la mitad el número de alumnos que aprueban justo o se

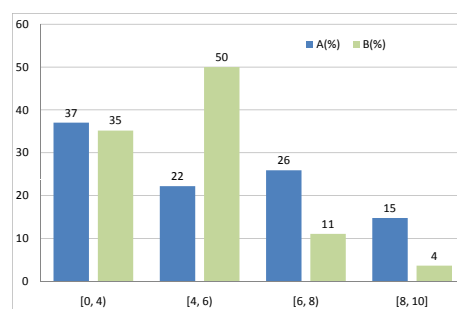


Figura 5: Comparación de notas finales en el mismo año académico, distinto cuatrimestre

quedan cerca de aprobar (de un 50 % a un 22 %).

5. Conclusiones y trabajo futuro

Nuestra experiencia como docentes en los últimos años nos ha movido a intentar buscar metodologías de aprendizaje que puedan motivar a los alumnos a participar de manera activa en el desarrollo de la asignatura.

La realización de tests a lo largo del curso está bastante extendida para que alumnos y profesores puedan examinar los objetivos docentes. Nosotros proponemos que las preguntas del test no sean *impuestas* por el profesor, si no propuestas por los alumnos, fomentando así su capacidad de síntesis y análisis. No obstante, la gestión de las preguntas enviadas por los alumnos puede ser una ardua tarea. En esta línea hemos integrado un conjunto de herramientas que facilita al profesor la labor de generar estos tests a partir de las preguntas recibidas.

Por un lado, esta experiencia nos ha demostrado que los alumnos están dispuestos a realizar el trabajo pasivo (realizar los tests), ya que se presentaron más alumnos de los que asistían con asiduidad a cla-

se. Por otro, lado los alumnos están menos dispuestos a realizar el trabajo activo, ya que sólo un 59% de los alumnos que asistían a clase subieron preguntas. Sin embargo, el resultado final de la experiencia demuestra que la mayoría de los alumnos que aprueban la asignatura son aquellos que participaron en el trabajo activo (un 88% en nuestro caso). Además, el porcentaje de alumnos que aprueban con mayor calificación (notables y sobresalientes) aumenta considerablemente con respecto a la metodología tradicional de realización de tests.

Con respecto al trabajo futuro, este curso académico estamos aplicando la metodología propuesta un total de 11 profesores en 8 asignaturas distintas. Además, estamos planificando el desarrollo de una herramienta más completa que permita a los alumnos no sólo contribuir con la aportación de preguntas, sino con la evaluación de la calidad y dificultad de las preguntas enviadas por sus compañeros. Esto permitirá satisfacer algunas de las demandas solicitadas por los estudiantes: que las preguntas de mayor calidad tengan más probabilidad de aparecer en el test, y algunas de las demandas del profesorado: que el profesor sepa cuáles son los conceptos más difíciles de asimilar.

Agradecimientos

Este trabajo está financiado por la Universidad Complutense de Madrid a través del Proyecto de Innovación y Mejora de la Calidad Docente PIMCD 196/2010.

Referencias

- [1] E. L. Deci and R. M. Ryan, *Intrinsic Motivation*. John Wisley & Sons, Inc. All, 2010.
- [2] “Apple distinguished educators,” <http://ade.apple.com/application/europe/es>, 2011.
- [3] F. Lazarinis, S. Green, and E. Pearson, “Engineering an interoperable multimedia assessment authoring and runtime environment conforming to IMS QTI,” *Int. J. Adv. Media Commun.*, vol. 3, pp. 260–276, July 2009. [Online]. Available: <http://portal.acm.org/citation.cfm?id=1568658.1568660>
- [4] T. Adeyinka and S. Mutula, “A proposed model for evaluating the success of WebCT course content management system,” *Comput. Hum. Behav.*, vol. 26, pp. 1795–1805, November 2010. [Online]. Available: <http://dx.doi.org/10.1016/j.chb.2010.07.007>
- [5] R.-A. Lee and J. DePue, “Using baldrige method frameworks, excellence in higher education standards, and the Sakai CLE for the self assessment process,” in *Proceedings of the 38th annual fall conference on SIGUCCS*, ser. SIGUCCS '10. New York, NY, USA: ACM, 2010, pp. 165–170. [Online]. Available: <http://doi.acm.org/10.1145/1878335.1878377>
- [6] S. Arnold and J. Fisler, “OLAT: The Swiss Open Source Learning Management System,” in *Proceedings of the 2010 International Conference on e-Education, e-Business, e-Management and e-Learning*, ser. IC4E '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 632–636. [Online]. Available: <http://dx.doi.org/10.1109/IC4E.2010.76>
- [7] A. Cavalli, S. Maag, S. Papagiannaki, and G. Verigakis, “From UML models to automatic generated tests for the dotLRN e-learning platform,” *Electron. Notes Theor. Comput. Sci.*, vol. 116, pp. 133–144, January 2005. [Online]. Available: <http://dx.doi.org/10.1016/j.entcs.2004.02.072>
- [8] L.-H. Kuo, T.-H. Tseng, H.-J. Yang, and H.-H. Yang, “Design a Moodle synchronous learning activity,” *WTOS*, vol. 9, pp. 409–421, April 2010. [Online]. Available: <http://portal.acm.org/citation.cfm?id=1853805.1853813>
- [9] N. Jøglar, D. Martín, J. M. Colmenar, and I. Martínez, “iTest: An Online Tool for Assessment and Self-Assessment in Mathematics,” in *Proceedings of the 2009 11th IEEE International Symposium on Multimedia*, ser. ISM '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 663–668. [Online]. Available: <http://dx.doi.org/10.1109/ISM.2009.56>
- [10] “Megatest,” <http://www.zipposoft.com/>, 2011.
- [11] C. van de Sande, “Free, open, online, mathematics help forums: the good, the bad, and the ugly,” in *Proceedings of the 9th International Conference of the Learning Sciences - Volume 1*, ser. ICLS '10. International Society of the Learning Sciences, 2010, pp. 643–650. [On-

line]. Available: <http://portal.acm.org/citation.cfm?id=1854360.1854442>

[12] “Texercises,” <http://www.texercises.com/>.

Sesión 6A:
Evaluación del alumnado III

Modelo para la evaluación de competencias: una experiencia aplicada a Fundamentos de Programación

Jesús Serrano-Guerrero, Francisco P. Romero,
Emilio Fdez-Viñas, José A. Olivas

Departamento de Tecnologías y Sistemas de Información
Universidad de Castilla-La Mancha
Paseo de la Universidad 4
13071 Ciudad Real

{jesus.serrano, franciscop.romero, emilio.fernandez,
joseangel.olivas}@uclm.es

Resumen

Este trabajo presenta un modelo para la evaluación únicamente de competencias, dejando de lado la evaluación de contenidos tan utilizada en años anteriores. Este modelo se presenta de forma genérica, tal y como podría aplicarse a cualquier asignatura, para luego ser aplicado de forma real sobre una asignatura concreta, en este caso, Fundamentos de Programación impartida en la Escuela Superior de Informática de Ciudad Real [3]. El modelo estructura todos los elementos evaluables en distintos grupos (bloques, actividades y asignatura) y asigna una serie de pesos a cada una de las competencias que intervienen en cada uno de estos elementos. La asignación de pesos permite flexibilizar la importancia de cada competencia dependiendo de los objetivos marcados para cada asignatura. La gran aportación lograda por este modelo ha sido la transmisión del concepto de competencia al alumnado y una mejor comunicación entre el profesorado y los estudiantes en términos de competencias.

Summary

This work presents a model for evaluating skills, forgetting the previously used content-based evaluation. This model is presented first in a generic way, suitable for any subject; then, it is applied to a concrete one, Programming Foundations, at the Escuela Superior de Informática from Ciudad Real. This model organizes all assessed elements into three categories (blocks, activities and subject) and assigns a weight for each skill which participates

in them. The assignment of each importance weight determines the objectives proposed for each subject. The great advantage of this model is transmission of the concept “skill” to the students. Also the communication between teachers and students and related to skills has been improved.

Palabras clave

Modelo de evaluación, competencias, Fundamentos de Programación, bloques evaluables, actividades evaluables

1. Motivación

Según el Espacio Europeo de Educación Superior (EEES), el objetivo final de la educación es la consecución de una serie de competencias que permitan al estudiante estar preparado para su incorporación al mundo laboral [1].

De acuerdo al REAL DECRETO 1393/2007, de 29 de octubre, por el que se establece la ordenación de las enseñanzas universitarias oficiales [4], los planes de estudio para acceder a un título deben garantizar la adquisición de una serie de competencias además de los conocimientos típicos sobre contenidos por parte de los estudiantes. Uno de los principales puntos de dicho decreto afirma que las competencias, tanto transversales como específicas, deben poder ser “evaluables”. Así pues, es necesario garantizar que cada una de las actividades evaluables llevadas a cabo en una asignatura esté dirigida a la adquisición de una serie de competencias de entre todas las planificadas para dicha asignatura.

Cada competencia no tiene por qué tener la misma importancia que el resto de competencias. Además las competencias no tienen por qué

adquirirse necesariamente ni en una asignatura concreta ni en un curso concreto, por lo que no es necesario hacer el mismo hincapié en todas las competencias que se trabajen en una asignatura, y por consiguiente, la adquisición de cada competencia no tiene por qué aportar en la misma proporción a la nota final de la asignatura.

El concepto de competencia cada vez tiene un mayor peso en guías docentes y memorias de grado; sin embargo, su aplicación y sobre todo su transmisión a los alumnos es una laguna sobre la que queda mucho camino por recorrer. Aunque cada vez más intentos tratan de acercar al alumno al concepto de competencia [2], no es fácil olvidar el concepto de examen o trabajo y el resultado de nota numérica final. Por tanto, es necesario un modelo en el cual el alumno olvide totalmente que su objetivo cuando estudia es adquirir una nota, sino transmitir al profesor qué competencias van siendo adquiridas de entre las propuestas en la guía docente.

Con el fin de cubrir las necesidades antes comentadas, este trabajo presenta un modelo candidato a ello así como un ejemplo de su aplicación a una clase de Fundamentos de Programación.

El resto del artículo está organizado como sigue: la sección 2 presenta el modelo para evaluar las competencias mientras que la sección 3 incluye un caso de estudio aplicado a la asignatura de Fundamentos de Programación. En la sección 4 se incluyen algunos resultados obtenidos tras la experiencia realizada y ciertas conclusiones a las que se ha llegado tanto desde el punto de vista del profesorado como del alumnado.

2. Modelo de evaluación basado en competencias

El modelo propuesto intenta evitar en todo momento el uso de los conceptos como examen o nota final de un examen, y los cambia por el concepto de actividad formativa evaluable y el de grado de consecución de una competencia. El concepto de nota final de la asignatura únicamente aparece al finalizar el curso, una vez hayan sido analizadas todas las competencias adquiridas o no por cada estudiante. Así el modelo propuesto consta de una serie de elementos evaluables como son:

- Competencias

- Bloques temáticos
- Actividades
- Nota final

A continuación se detalla la función de cada uno de los elementos enumerados anteriormente.

1. Competencias.

Dentro de la Memoria de Grado de la Escuela Superior de Informática de Ciudad Real se encuentra una serie de competencias que deben tenerse en cuenta a la hora de diseñar la guía docente de cada asignatura. Obviamente hay dos tipos de competencias: verticales y horizontales. Estas competencias se irán alcanzando a lo largo de las actividades evaluables encontradas en cada bloque temático. La importancia I_c de cada competencia C es relativa, no todas tienen la misma importancia, es decir, se pueden establecer una relación de grado entre ellas, las habrá más o menos importantes (ver Tabla 1). Este factor determinará cómo afectan a la nota final de la asignatura.

	C1	C2	C3	C4	Total
Asignat.	10	15	20	5	100

Tabla 1. Importancia de cada competencia para la asignatura

Así pues, matemáticamente diremos que se tiene un número n de competencias verticales CV y un número m de competencias horizontales CH representados de la siguiente manera:

$$CV = \{CV_1, CV_2, CV_3, \dots, CV_n\}$$

$$CH = \{CH_1, CH_2, CH_3, \dots, CH_m\}$$

2. Bloques temáticos

Una asignatura se compone de una serie t de bloques $B = \{B_1, B_2, \dots, B_t\}$ que son evaluables por separado dado que presentan una serie de contenidos bien definidos según la guía docente de la asignatura. Es decir, cada bloque está compuesto por una serie de unidades didácticas a trabajar a lo largo del curso. A lo largo de cada bloque temático se pretende alcanzar una serie de competencias bien definidas en la guía docente.

Cada competencia de cada bloque temático tiene asignado un porcentaje de la importancia que conlleva alcanzarla I_B , así la suma total de los porcentajes debe ser, obviamente, 100% (ver Tabla 2).

3. Actividades evaluables

Con el fin de poder evaluar si un alumno alcanza o no las competencias mencionadas anteriormente por cada bloque temático, están programadas una serie de z actividades evaluables $A = \{A_1, A_2, \dots, A_z\}$. Cada una de estas actividades cubre una serie de competencias en un cierto grado y pueden ser evaluables conforme a una nota numérica. La nota de cada competencia cubierta en una actividad A se denota por el símbolo: NC_A . Cada actividad pretende cubrir una serie de competencias; sin embargo, no todas tienen la misma importancia, este hecho se refleja mediante un grado de importancia I_A (ver Tabla 3).

3. Nota final

La nota final de la asignatura se calcula en función de las notas obtenidas en cada competencia de cada actividad, el peso asignado a cada competencia dentro de cada bloque temático así como el peso asignado a cada competencia dentro de la asignatura.

El modelo matemático propuesto también deja la posibilidad de que pudiera darse más importancia a cubrir las competencias de un tipo que a las otras por si los docentes por alguna cuestión lo consideraran necesario, como hacían la mayoría de los profesores antes de implantarse el método Bolonia, cuando sólo se valoraban las competencias verticales. Aunque nosotros damos igual importancia a ambas partes, pero poniendo como restricciones adicionales una nota mínima en cada una de las competencias con el fin de que en cierto grado todas las competencias se garanticen. Así pues, el cálculo de la nota final NF desde el punto de vista de las notas referentes a las competencias verticales (NV) y de las horizontales (NH) sería:

$$NF = 0,5 * NH + 0,5 * NV$$

Si se sumaran los pesos asignados a cada tipo de competencia, verticales y horizontales, y los resultados de las sumas no fueran las mismas, entonces el profesor habría dado más importancia a alguno de los tipos de competencias (ver Tabla 4).

	Horizontales	Verticales	Total
Asignatura	50	50	100

Esta situación podría ser útil en caso de que por ejemplo, el coordinador de curso decidiera que una/s determinada/s asignatura/s potenciaran una serie de competencias muy por encima de las otras con algún propósito docente específico. Por ejemplo, si se quisiera que los alumnos de primero potenciaran su capacidad lectora y de síntesis o el dominio de otros idiomas, las competencias X e Y podrían ser potenciadas muy por encima del resto con el fin, no sólo de que adquieran una serie de conocimientos acerca de informática sino también con el fin de dotarles de una serie de capacidades que no suelen adquirir en la secundaria y que serán fundamentales para el resto de cursos del Grado en Informática.

Los profesores deben puntuar con una nota numérica NC cada una de las competencias correspondientes en cada una de las actividades realizadas, así pues para cada alumno debería rellenarse la tabla 3 en las casillas que corresponda.

De forma general podría decirse que las notas medias correspondientes a las competencias horizontales y verticales se calculan de acuerdo a la siguiente fórmula:

$$NH = \sum_{i=1}^n I_{Ci} * \left(\sum_{j=1}^l I_{Bj} * \left(\sum_{k=1}^z I_{Ak} * NC_{ijk} \right) \right)$$

Sin embargo, con el fin de exigir que cada competencia se alcance en un cierto grado, se exige un mínimo en cada una de las competencias alcanzadas [5], un mínimo por cada bloque y un mínimo por cada actividad. De forma general, si el mínimo que se asigna a cada actividad es lo suficientemente restrictivo, se garantiza que los mínimos de bloque y competencia se alcancen sin problema alguno, así pues el algoritmo definitivo para calcular cualquier tipo de competencia, horizontal o vertical, con restricciones quedaría como muestra el Algoritmo 1, en este caso, suponemos que calculamos sólo las competencias horizontales.

Tabla 4. Importancia de cada tipo de competencias dentro de la asignatura

Bloques/ Competencias	B1	B2	B3	Total
C1	10	10	5	100
C2	5	30	20	100
C3	10	20	15	100
....	100

Tabla 2. Importancia de cada competencia para cada bloque

Actividades/ Competencias	A1	A2	A3	A4	A5	A6	...	Total
C1	10	10	5	100
C2	5	12	20	100
C3	6	8	15	100
C4	4	7	15	100
.....	100

Tabla 3. Importancia de cada competencia para cada actividad

```

NH ← 0, NotaBloque ← 0, NotaActividades ← 0
Para cada competencia horizontal i
  NotaBloque ← 0
  Para cada bloque j
    NotaActividades ← 0
    Para cada prueba evaluable k
      Si  $NC_{ijk} > \text{MinimoActividad}_k$ 
        NotaActividades ← NotaActividades +  $I_{Ak} * NC_{ijk}$ 
      Sino
        NH ← Supenso y fin
    Fin_si
  Fin_para

  Si NotaBloque >  $\text{MinimoBloque}_j$ 
    NotaBloque ← NotaBloque +  $I_{Bj} * \text{NotaActividades}$ 
  Si_no
    NH ← Supenso y fin
  Fin_si
Fin_para
Si NotaBloque >  $\text{MinimoCompetencia}_i$ 
  NH ← NH +  $I_{ci} * \text{NotaBloque}$ 
Si_no
  NH ← Supenso y fin
Fin_si
Fin_para
Devolver NH

```

Algoritmo 1. Cálculo de la nota final para cálculo nota con restricciones

	CV1	CV2	CV3	CV4	CV5	CV6	CH1	CH2	CH3	CH4	CH5	CH6
Bloque 1	20	20	20	0	0	0	33	33	33	33	33	33
Bloque 2	30	30	30	50	50	50	33	33	33	33	33	33

Bloque 3	50	50	50	50	50	50	34	34	34	34	34	34
Total	100	100	100	100	100	100	100	100	100	100	100	100

Tabla 5. Importancia de cada competencia para cada bloque de la asignatura Fundamentos de Programación

Si comparásemos un modelo de evaluación tradicional basado en contenidos utilizado en años anteriores con el aquí propuesto, veríamos que ciertas variables se mantienen pero cambian su enfoque dado que se centran en la evaluación de las competencias. Así pues, los pesos asignados en nuestra evaluación no basada en competencias serían:

- Importancia del bloque dentro de las asignaturas.
- Importancia de la actividad dentro del bloque.
- Importancia del tipo de actividad dentro de la asignatura.

Las competencias verticales y horizontales utilizadas fueron las definidas en la guía docente de la asignatura y que fueron sacadas de la Memoria de Grado de la Escuela Superior de Informática de C. Real aprobada por la ANECA.

Esta asignatura se compone de 5 unidades temáticas y esas 5 unidades se agrupan por su contenido y complejidad en 3 bloques temáticos:

1. Bloque 1:

- Tema 1. Elementos de programación y lenguajes

- Tema 2. Introducción a la programación

2. Bloque 2:

- Tema 3. Programación estructurada y modular

3. Bloque 3

- Tema 4. Estructuras de datos

- Tema 5. Recursividad

Evidentemente la asignatura tiene un carácter incremental, es decir, cada tema requiere de un conocimiento exhaustivo de los temas anteriores, por lo que el dominio de los últimos temas implica el dominio de los primeros temas. Así pues, parece lógico que ciertas competencias de los últimos bloques tengan un mayor peso que las del primero, y por esa razón la distribución que hicieron los profesores de la asignatura fue la siguiente. Los pesos utilizados en cada bloque se muestran en la Tabla 5.

En nuestro caso vamos a suponer que las competencias, tanto verticales como horizontales,

tienen la misma importancia como se muestra en la Tabla 4.

En concreto cada bloque temático (B) de la asignatura consta de 4 tipos de actividades evaluables (A):

- Ejercicios en casa (E): Consisten en la resolución de algún problema, generalmente en papel sobre el que se suele dar un corto espacio de tiempo, de un día para otro, y que se resuelven en clase. Un ejemplo podría ser la implementación de algún método que realice cualquier función simple o la realización un trabajo sobre algún tipo de algoritmo concreto buscando información en algún libro recomendado. Dependiendo de dicha complejidad, el trabajo puede realizarse de forma individual o en grupos reducidos.
- Ejercicios en clase (G): Consiste en la resolución de una serie de ejercicios en papel durante una hora como máximo.
- Tests (T): Cuestionario de preguntas cortas sobre algún tema con una serie de respuestas a elegir y que se realiza en media hora.
- Prácticas (P): Consiste en la implementación de un programa informático de cierta envergadura para el que se dan entre 2 y 3 semanas. Las prácticas se realizan generalmente por parejas.

La importancia asignada a cada una de las competencias alcanzables en cada actividad se muestra en la Tabla 7. La presencia de un cero indica que esa competencia no es objetivo de la prueba realizada. La tabla anterior refleja cómo las competencias son evaluadas a lo largo de todo el curso, por eso, la suma a final de curso de cada competencia es 100. Sin embargo, si quisiéramos evaluar cada prueba, es decir, que el alumno supiera qué nota ha obtenido en una determinada prueba, habría que considerar una visión vertical de la tabla, y lógicamente los pesos no serían los mismos.

Finalmente, la importancia de cada competencia para la asignatura, divididas entre horizontales y verticales, puede observarse en la

Tabla 6 (el símbolo Cx indica la competencia x de cada tipo: horizontal (CHx) y vertical (CVx)).

	C1	C2	C3	C4	C5	C6	Total
Horizontales	20	10	10	20	20	20	100
Verticales	10	10	20	20	20	20	100

Tabla 6. Importancia de cada competencia para la asignatura

3.1. Metodología seguida

La metodología seguida para poder aplicar el modelo de evaluación propuesto fue la siguiente. Al comenzar el curso el profesor trató de explicar cuál es el objetivo de las competencias dentro del nuevo Espacio Europeo de Educación Superior. Una vez transmitida esta idea, se pasó a intentar transmitir cuáles eran las competencias que habían sido seleccionadas para esa asignatura y cuáles eran los beneficios que podrían suponerles de cara a su futura incorporación al mercado laboral. También se comentó qué aspectos iban a ser tenidos en cuenta para poder evaluar cada competencia, especialmente las horizontales que podían parecer menos objetivas.

Antes de realizar cada actividad evaluable, el profesor explicaba qué se pretendía lograr con cada prueba, qué competencias se pretendían cubrir y cuál sería la forma de demostrar el grado de consecución de cada una de ellas. Una vez realizada cada prueba, el profesor colgaba en la herramienta Moodle el grado de consecución de cada una de las competencias marcadas para que el alumno supiera en qué grado iba alcanzando cada una de ellas.

Es obvio que cada competencia tiene una calificación por lo que es prácticamente imposible olvidar este concepto; sin embargo, siempre se procuró hablar de grado de consecución de competencias en lugar de aprobado o suspenso de competencias con el fin de cambiar, poco a poco, la mentalidad del alumnado sobre la nueva forma de evaluación.

4. Resultados

A continuación se comentan algunas de los resultados y conclusiones obtenidas a lo largo de esta experiencia.

El número de alumnos del grupo elegido de la asignatura es 27 aunque sólo 25 han realizado de forma regular todas las pruebas necesarias para aprobar la asignatura. De estos 25 alumnos, el número de aprobados fue 13, la mayoría de ellos repetidores o gente procedente de ciclos superiores con conocimientos ya previos sobre programación. La nota media de todos los alumnos fue 4,2.

La competencia vertical más aprobada fue CV2 y la más suspendida fue CV4. Mientras que la competencia horizontal más aprobada fue CH1 y la más suspendida fue CH6. El conocimiento del inglés con respecto a la bibliografía propuesta es un hándicap bastante grande que los profesores de la asignatura achacan principalmente a deficiencias en la Educación Secundaria Obligatoria.

Hay que destacar el hecho de que prácticamente los resultados fueron bastante similares aunque el número de aprobados en este caso fue 11 en lugar de 13. Los dos alumnos que menos nota habían obtenido en la evaluación por competencias hubieran suspendido en la evaluación por contenidos, sin embargo, las competencias horizontales les dan un plus que hace que su nota sea algo superior.

Actividades / Competencias	T1	E1	G1	P1	T2	E2	G2	P2	T3	E3	G3	P3	Total
CV1	0	0	0	10	0	10	10	10	0	20	20	20	100
CV2	5	5	5	5	8	8	8	8	12	12	12	12	100
CV3	5	5	5	5	10	10	10	10	10	10	10	10	100
CV4	5	5	5	5	10	10	10	10	10	10	10	10	100
CV5	0	0	0	0	10	10	10	10	15	15	15	15	100
CV6	5	5	5	5	8	8	8	8	12	12	12	12	100
CH1	8	8	8	8	8	8	8	8	8	8	8	8	100
CH2	8	8	8	8	8	8	8	8	8	8	8	8	100
CH3	8	8	8	8	8	8	8	8	8	8	8	8	100
CH4	5	5	5	5	8	8	8	8	12	12	12	12	100
CH5	5	5	5	5	8	8	8	8	12	12	12	12	100
CH6	8	8	8	8	8	8	8	8	8	8	8	8	100

Tabla 7. Importancia de cada competencia para cada actividad de la asignatura Fundamentos de Programación

También se observa que las notas tanto de los peores como de los mejores alumnos son superiores en la evaluación basada en competencias, principalmente porque las competencias horizontales siempre añaden algo de nota que antes no se tenía en cuenta.

4.1. Comentarios hecho por los propios alumnos

Analizando todos los comentarios realizados por los alumnos, cabe destacar como más interesantes los siguientes:

- Competencias demasiado genéricas. Son difíciles de entender algunas de las competencias por lo que realmente no saben qué es lo que evalúa el profesor.
- No saben qué tienen que hacer para adquirir esa competencia, no saben cómo estudiar.
- La no existencia de una determinada nota resta competitividad. Los buenos alumnos se sienten más motivados si ven que sacan una buena nota que si ven que han alcanzado un pequeño porcentaje de una competencia. Mientras que los malos alumnos ni siquiera son conscientes de que a mitad de la asignatura ya están suspensos si no ven una nota parcial.
- Los alumnos no son capaces de explicarles a sus padres que los exámenes ahora ni se

aprueban ni se suspenden, sino que se adquieren o no una serie de competencias que alcanzar a entender para qué les servirán en el futuro.

Aunque se les han presentado las dos formas de evaluación y a pesar de empezar a entender la forma basada en competencias, creen es mucho más clara e incluso realista la forma de calificación basada en contenidos de toda la vida. Ambas pueden ser complementarias pero creen es muy difícil de valorar y entender si realmente se alcanzan o no las competencias trabajadas.

4.2. Conclusiones alcanzadas por los profesores

Los profesores participantes en el experimento hemos sacado también una serie de conclusiones interesantes de cara al planteamiento de la asignatura en cursos futuros:

- Dificultad para evaluar de forma objetiva el grado de consecución de una competencia. Algunas de ellas son demasiado subjetivas.
- Dificultad para transmitir al alumno el concepto de competencia porque algunas son difíciles incluso de entender por el propio profesorado.
- El hecho de exigir un mínimo de cada competencia hace que las competencias, especialmente las horizontales, no tiendan a

ser olvidadas una vez que se alcanza el máximo exigido para conseguirla.

- Solapamiento entre algunas de las competencias utilizadas por lo que se crean dudas sobre qué se propuso tanto en la guía docente de la asignatura como incluso en la Memoria de Grado de la propia escuela donde se trabaja. Posiblemente habrá que replantear algunas de las competencias que debe cubrir la asignatura.
- Sensación de vuelta al instituto. La corrección de las competencias horizontales crea una sensación de frustración dado que el concepto de profesor en Informática que se concebía hace unos años no incluía tareas como la corrección de la forma de redactar de los alumnos o la expresión oral.
- Sobre-esfuerzo para diseñar las actividades intentando enfocarlas a la consecución de las competencias e incluso para pensar una forma más o menos objetiva para poder evaluarlas. Estos conceptos no habían sido trabajados antes por muchos de los profesores.

En definitiva, la experiencia puede considerarse como muy positiva desde el punto de vista del profesorado porque les ha obligado a trabajar en todo momento desde el punto de vista de las competencias, cosa que no habían hecho anteriormente. Además a los alumnos se les ha inculcado una serie de ideas positivas de cara a cambiar su concepción sobre la evaluación de futuras asignaturas y sobre qué son realmente las competencias según el EEES.

5. Conclusiones

Este trabajo pretende presentar una experiencia llevada a cabo con el fin de cambiar el concepto de evaluación de contenidos por el de evaluación de competencias. Para ello, se ha propuesto en modelo que pretende trabajar en todo momento con el concepto de competencia y nunca con el de calificación de una prueba. No se tiene en ningún momento la consciencia de que se ha aprobado una prueba concreta o no, por lo que ese concepto se pierde, y como consecuencia los objetivos de los alumnos cambian y de igual forma, cambia la forma de afrontar cada actividad. Lo que deben demostrar al profesor ya no es un único resultado sino algunas capacidades más.

El sistema presenta una gran ventaja y es que permite potenciar ciertas competencias de forma objetiva porque se consideren o no necesarias por ejemplo desde la dirección de la Escuela con el fin de corregir ciertos defectos que presentan los alumnos por ejemplo o porque el mercado laboral demande con más fuerza una competencia concreta.

El modelo ha sido puesto en marcha en una clase de Fundamentos de Programación obtenidos una serie de conclusiones y resultados alentadores para continuar con su implantación y mejora.

Agradecimientos

Este proyecto está financiado gracias al proyecto FIDELIO (TIN2010-20395, MEC-FEDER), proyecto de excelencia SCAIWEB-2 (PEIC09-0196-3018, JCCM), proyecto PLINIO (POII10-0133-3516, JCCM), proyecto de Innovación Docente HIPERION (013702011, UCLM) y el proyecto HECAOT (TC20112330, UCLM).

Referencias

- [1] ANECA: Libro Blanco del Título de Grado en Ingeniería Informática, <http://www.aneca.es>
- [2] Jiménez, F., Martínez, G., Sánchez, G., Juárez, J.M. y Paredes, S. *Un sistema de evaluación basado en competencias: Ejemplo para la asignatura Tecnología de la Programación del título de Grado en Ingeniería Informática por la Universidad de Murcia*. Actas de las XV Jornadas de Enseñanza Universitaria de Informática, Jenui 2009, pp. 193-200, Barcelona, Julio 2009.
- [3] Guía docente de la Escuela Superior de Informática de Ciudad Real, <http://webpub.esi.uclm.es>.
- [4] REAL DECRETO 1393/2007, de 29 de octubre, por el que se establece la ordenación de las enseñanzas universitarias oficiales, <http://www.mec.es/>
- [5] Valderrama, E., Rullán, M., Sánchez, F., Pons, J., Cores, F., Bisbal, J. *La evaluación de competencias en los Trabajos Fin de Estudios* Actas de las XV Jornadas de Enseñanza Universitaria de Informática, Jenui 2009, pp. 405-412, Barcelona, Julio 2009

Autoevaluación y evaluación entre iguales en una asignatura de redes de ordenadores

María Cavas Toledo¹, Francisco Chicano García², Francisco Luna Valero², Luis Molina Tanco³

¹Departamento de Psicología y Metodología de las Ciencias del Comportamiento, Facultad de Psicología

²Departamento de Lenguajes y Ciencias de la Computación, E.T.S. Ingeniería Informática

³Departamento de Tecnología Electrónica, E.T.S. Ingeniería de Telecomunicación

Universidad de Málaga

29071 Málaga

mcavas@uma.es, chicano@lcc.uma.es, flv@lcc.uma.es, lmtanco@uma.es

Resumen

Con la llegada del EEES se ha fomentado el uso de la evaluación continuada como un ingrediente de la docencia de calidad que permite al alumno mantenerse puntualmente informado de su progreso en el proceso de aprendizaje. Este trabajo se centra en el uso de la autoevaluación y la evaluación entre iguales como base para conseguir esta evaluación continuada. En él detallamos la experiencia, durante los cursos 2008/09 y 2009/10, del uso de estas dos herramientas de aprendizaje en el contexto de una asignatura de redes de ordenadores de tercer curso de Ingeniería Técnica en Informática de Gestión.

Summary

The use of continuous assessment has been encouraged in the adaptation to the European Higher Education Area. It is considered an ingredient to the high quality teaching that allows the student to have a fast feedback of her/his learning progress. This work focuses in the use of self assessment and peer assessment as the basis for a continuous evaluation system. We detail our experience during the years 2008/09 and 2009/10 on using such teaching tools in the context of a subject on computer networks.

Palabras clave

Autoevaluación, evaluación entre iguales, EEES, redes de ordenadores, aprendizaje activo

1. Introducción

Con la llegada del Espacio Europeo de Educación Superior (EEES) se ha hecho necesario un cambio en las metodologías docentes y un replanteamiento

de diversos aspectos de la enseñanza universitaria, que ahora deben estar centrados en el aprendizaje del alumno. Uno de estos aspectos a los que hay que prestar especial interés es la evaluación. En el marco educativo del EEES, esta evaluación debe estar orientada de forma definitiva a los nuevos procesos de enseñanza/aprendizaje y al conjunto más amplio de conocimientos y competencias (capacidades, habilidades, comprensión, actitudes...) que en este contexto se promueven [3, 4]. Así, la evaluación debe permitir a estudiantes y profesores identificar fortalezas y debilidades en el proceso de aprendizaje. Es ampliamente admitido que la evaluación continuada es un ingrediente de la docencia de calidad que permite al alumno mantenerse puntualmente informado de su progreso en el proceso de aprendizaje (retroalimentación frecuente), y al profesor obtener información sobre el desarrollo del aprendizaje de sus alumnos. La evaluación es así entendida como evaluación formativa. No obstante, también se hace necesario encontrar herramientas factibles para la evaluación calificativa, es decir, la que usamos para determinar la calificación que acredita el nivel de aprendizaje conseguido por el alumno.

Este trabajo se centra en el uso de la autoevaluación y la evaluación entre iguales [5, 8] como base para la organización de un sistema de evaluación que proporcione, en combinación con la utilización de las nuevas tecnologías y de las plataformas de enseñanza virtual (como *moodle*), información con prontitud, proporcionando así una evaluación continuada factible. Como herramientas de aprendizaje, la autoevaluación y la evaluación entre iguales, también llamada co-evaluación, presentan múltiples virtudes, entre las que destacan [1]:

1. Los alumnos deben utilizar criterios de corrección para evaluarse a sí mismos y a sus compañeros, por lo que los han de aprender e interior-

rizar y así ajustan cada vez más sus respuestas a lo que se espera de ellos en tareas posteriores.

2. Se desarrolla el hábito de la reflexión para poder emitir juicios de valor sobre el trabajo propio o el de los compañeros.
3. Los alumnos se convierten en ayudantes del profesor para la dura tarea de evaluar las actividades del curso, bien sea con propósito formativo o con propósito calificativo.

Estas virtudes inciden directamente en algunos de los objetivos concretos que se persiguen bajo el marco del EEES, como son formar personas con capacidad para aprender de forma autónoma o desarrollar una actitud crítica frente a trabajos propios y de los demás. El alumno pasa a ser evaluador, experto, y se involucra en su propio aprendizaje.

En este trabajo se detalla la experiencia, durante los cursos 2008/09 y 2009/10, del uso de la autoevaluación y evaluación entre iguales en el contexto de la asignatura *Informática Distribuida*, una asignatura de redes de ordenadores que se ubica en el tercer curso de Ingeniería Técnica en Informática de Gestión. El haber implementado este sistema de evaluación durante dos cursos académicos nos ha permitido reconsiderar algunas de las decisiones iniciales y, con ello, ir introduciendo mejoras en su aplicación.

El resto del trabajo se estructura como sigue. La siguiente sección está dedicada a describir el contexto donde se ha desarrollado la experiencia. La Sección 3 detalla cómo se han implementado la autoevaluación y co-evaluación en la asignatura. Las secciones 4 y 5 incluyen, respectivamente, los resultados obtenidos y una discusión de los mismos. Finalmente, la Sección 6 presenta las principales conclusiones y los pasos a seguir para introducir mejoras.

2. Contexto de la experiencia

La experiencia que aquí se relata forma parte de un Proyecto de Innovación Educativa (PIE) compuesto exclusivamente por profesores noveles y cuyo objetivo principal era el de aplicar las técnicas de autoevaluación y co-evaluación como estrategias de evaluación para mejorar el proceso de enseñanza-aprendizaje. En este PIE participaron profesores de hasta tres titulaciones diferentes y se aplicaron las técnicas de auto- y co-evaluación en cuatro asignaturas distintas. Uno de los objetivos del PIE era com-

probar si las técnicas mencionadas son igualmente válidas en los distintos contextos en que fueron planteadas (distintas asignaturas y titulaciones). Las otras titulaciones son Psicología e Ingeniería de Telecomunicación.

2.1. La asignatura

La asignatura en la que se ha llevado a cabo la experiencia es una asignatura de tercer curso de Ingeniería Técnica en Informática de Gestión cuya temática es el de las redes de ordenadores. El objetivo principal de la asignatura es iniciar al alumno en los fundamentos de redes de ordenadores con el fin de adquirir la base que le permita identificarlas, implantarlas, así como desarrollar aplicaciones sobre ellas. Esta asignatura es la única obligatoria en los planes de estudios que trata el tema de las redes de ordenadores. La estructura en temas de la asignatura es la siguiente:

1. *Introducción a las redes de ordenadores.* En este primer tema se presentan definiciones y conceptos muy generales sobre redes, tales como los conceptos de protocolo, arquitectura en capas, tiempo de propagación, tiempo de transmisión, ancho de banda, modelo de referencia OSI, modelo TCP/IP, clasificación de redes, etc.
2. *La capa de aplicación.* En este tema se presenta el nivel más alto del modelo de referencia OSI y se detallan varios protocolos del nivel de aplicación importantes dentro de la arquitectura TCP/IP. Ejemplos de estos protocolos son HTTP, FTP, SMTP, POP3, etc.
3. *La capa de transporte.* Aquí se presentan varios algoritmos básicos de envío fiable de datos en redes de ordenadores, tales como la parada y espera o la repetición selectiva. Posteriormente el tema se centra en el protocolo TCP, del que se detalla la cabecera, su funcionamiento y su control de congestión. En este tema los alumnos aprenden a usar la API de Java para la programación con sockets y realizan algunas prácticas de laboratorio donde implementan un cliente y un servidor básicos.
4. *La capa de red.* Se estudian algunos de los algoritmos de encaminamiento más importantes y se presenta con detalle el protocolo IP. Se presentan los conceptos de dirección IP, máscara

de subred, etc. y algunos otros protocolos y procedimientos relacionados con esta capa, como NAT o DHCP.

5. *La capa de enlace de datos y redes de área local.* En este último tema se estudian algunos conceptos especialmente relacionados con la capa de red, tales como la detección y corrección de errores o los algoritmos de control de acceso al medio. Algunos ejemplos son ALOHA, CSMA/CD, MACAW, etc. El tema termina centrándose en dos tecnologías de red concretas: ethernet y WiFi.

El esquema temático de la asignatura sigue las sugerencias del libro de Kurose y Ross [7] y aborda el contenido de la materia centrándose en el modelo TCP/IP y comenzando la exposición por las capas altas del modelo de referencia OSI, a diferencia del enfoque más tradicional de comenzar por las capas bajas del modelo OSI. La motivación para seguir este orden es que el alumno entra en contacto rápidamente con los conceptos que a nivel de usuario tiene que manejar habitualmente. Nuestra opinión es que de esta forma se consigue atraer más su atención.

Desde el curso académico 2006/07 hasta el curso 2009/10, esta asignatura ha estado inmersa dentro de un plan piloto para la adecuación de las asignaturas al EEES. Este plan piloto tenía como objetivos la elaboración de material didáctico específico para la nueva metodología docente, el establecimiento de mecanismos que aumentasen la motivación e incentivasen el trabajo continuado por parte del alumno, el desarrollo de estrategias de evaluación que permitiesen valorar este trabajo continuado y el establecimiento de conclusiones que sirviesen de referencia como punto de partida para afrontar los nuevos grados que han comenzado en el curso 2010/11.

Como consecuencia de este proceso de adaptación y cambio, se introdujeron en la asignatura una serie de actividades para seguir con las directrices formuladas por el plan piloto y, en general, para cambiar el paradigma de enseñanza por el de aprendizaje. Estas actividades, así como la evaluación que se propuso se presentan a continuación.

2.2. Metodología docente y evaluación

En esta sección describimos la metodología docente empleada en la asignatura tras la adaptación de la misma al EEES dentro del plan piloto.

A lo largo del curso el profesor expone la materia en clases magistrales y se realizan dos sesiones de prácticas en un laboratorio. En las sesiones teóricas el alumno aprende los fundamentos de las redes de ordenadores estudiando por separado las distintas capas de los modelos de referencia. En dichas sesiones se fomenta la participación del alumno a través de preguntas en clase.

Una característica que no es posible obviar del contenido de la asignatura es que es muy dinámico. La tecnología de redes cambia muy rápidamente y aparecen nuevos servicios con mucha frecuencia. Con el objetivo de transmitir este aspecto de dinamismo a los alumnos se les anima a buscar información relacionada con la asignatura en la Web, periódicos y revistas de actualidad para fomentar la autoformación (noticias relacionadas con la asignatura). Estas noticias son entregadas al profesor en papel o colgadas en un foro a través de la plataforma de enseñanza virtual y son discutidas en clase con cierta periodicidad. Además de indicar el titular y una referencia a la noticia, el alumno debe reflexionar sobre ella y escribir un pequeño párrafo con su opinión sobre la misma. Esta actividad no requiere un gran esfuerzo por parte del alumno pero lo mantiene actualizado en temas relacionados con la asignatura.

Adicionalmente, y de forma también voluntaria, los alumnos pueden realizar un trabajo sobre un tema propuesto por los profesores o por ellos mismos (siempre con el consentimiento del profesor). Estos trabajos deben versar sobre aspectos que no se tratan en la asignatura o en los que no se profundiza. Algunos ejemplos de temas propuestos para estos trabajos son SSL, Kerberos, Bluetooth, Redes P2P, etc. Estos trabajos pueden ser realizados por una o dos personas y son entregados en dos fases. En una primera fase los alumnos buscan la información y elaboran un primer borrador del trabajo. Este borrador es revisado por el profesor, quien les da indicaciones para mejorarlo. Tras acometer las sugerencias propuestas por el profesor, los alumnos preparan la versión definitiva del trabajo y la entregan. Tras una segunda revisión de los trabajos, el profesor cita a los autores de los mismos en horario de tutorías y les realiza una entrevista para comprobar lo que han aprendido.

En las sesiones de laboratorio los alumnos utilizan los conocimientos teóricos adquiridos para desarrollar programas que se comunican a través de una

red de ordenadores. En particular, los alumnos deben implementar un sencillo servidor y un sencillo cliente que se conecta con el servidor. En otra práctica de laboratorio los alumnos aprenden a utilizar la herramienta Wireshark para el análisis de protocolos. En dicha sesión se les presenta un protocolo, tal como FTP, y haciendo uso de la herramienta deben averiguar cómo se traducen las acciones de usuario en comandos del protocolo.

El sistema de evaluación de la asignatura considera cuatro componentes (indicios de aprendizaje):

1. Participación del alumno. Tiene en cuenta la participación del alumno en la elaboración y discusión de noticias relacionadas con la asignatura y los ejercicios elaborados en clase. En este apartado es donde se incluyen las actividades de auto- y co-evaluación que se describen en la Sección 3. La puntuación máxima que se puede sacar es 1 punto.
2. Trabajos monográficos voluntarios. Estos trabajos, detallados anteriormente, pueden suponer hasta 2 puntos en la nota final.
3. Exámenes parciales. Durante el cuatrimestre se realizan dos exámenes parciales tipo test donde se evalúa el conocimiento del alumno de los conceptos teóricos que se estudian en clase. Cada uno de estos exámenes parciales puede suponer hasta 1 punto en la nota final, de modo que la máxima nota que se puede obtener por esta actividad son 2 puntos.
4. Examen final. Los alumnos deben realizar un examen final en el que la nota máxima es 10 puntos. Si un alumno no ha realizado ninguna otra actividad durante el curso deberá sacar un mínimo de 5 en este examen. Los alumnos que han realizado otras actividades sólo deben sacar un 4 para considerar el examen final aprobado.

La nota final de la asignatura es la suma de las calificaciones de las distintas actividades. Para aprobar la asignatura hay que sacar el mínimo exigido en el examen final (4 o 5 dependiendo del trabajo del alumno) y además hay que sacar una puntuación mayor que 5 en la nota final. La puntuación máxima que puede sacar un alumno es 15. A todos los alumnos que pasan de 10 se les pone 10. Esta forma de evaluación fomenta el aprendizaje continuo, ya que los alumnos pueden obtener calificaciones altas si trabajan durante el curso.

2.3. Perfil de los alumnos

La asignatura se imparte en tercer curso de Ingeniería Técnica en Informática de Gestión. Es decir, en el último curso de dicha titulación. Por lo tanto, muchos de los alumnos que cursan esta asignatura están terminando las asignaturas de la carrera y empezando a elaborar el Proyecto de Fin de Carrera (PFC). No obstante, como suele ocurrir en carreras de dificultad alta, también hay bastantes alumnos a los que aún les queda por aprobar asignaturas de cursos anteriores. Por otro lado, algunos de los alumnos de la asignatura se encuentran ya en el mercado laboral. Esto es algo bastante frecuente en los últimos cursos de las titulaciones de informática.

En general, por tanto, se trata de alumnos con una cierta madurez que se encuentran motivados (ya que les queda poco para terminar). A esto debemos añadir que el tema de las redes de ordenadores suele ser especialmente atractivo para los alumnos debido al gran protagonismo que tiene hoy en día Internet y todas las tecnologías de comunicaciones relacionadas. La asignatura protagonista de la experiencia es la única de redes de ordenadores que cursan durante la titulación. Posiblemente como consecuencia de todo esto, los alumnos se muestran muy participativos en clase y no suelen quejarse mucho del trabajo que deben realizar.

3. Descripción de la experiencia

Originalmente el enfoque de la asignatura era principalmente teórico. Un primer cambio que se decidió realizar para introducir las actividades de auto- y co-evaluación fue la inclusión de ejercicios prácticos en la asignatura. Algunos ejemplos de estos ejercicios son el cálculo del ancho de banda de una red, el cálculo de la eficiencia de un protocolo, la configuración de redes IP, el tráfico detallado de tramas entre dos máquinas, etc. Además de dedicar horas de clase a desarrollar las destrezas necesarias para la realización de este tipo de ejercicio, se introdujeron, durante el curso académico 2008/09, tres relaciones de problemas con el objetivo de usarlas como base para la aplicación de las nuevas técnicas didácticas. La incorporación de las relaciones de problemas a las actividades de la asignatura vino acompañada de una modificación del examen final. Hasta entonces el examen final estaba formado por 5 pre-

guntas de desarrollo teórico sobre el contenido de la asignatura y a partir del curso 2008/09 dos de las preguntas teóricas fueron sustituidas por problemas similares a los presentes en las relaciones. El resto de actividades (trabajos, noticias, prácticas, etc.) se mantuvieron como en el año anterior. Con respecto a la evaluación, las nuevas actividades de auto- y co-evaluación se incluyeron como parte de la nota de participación del alumno y, por lo tanto, podían suponer hasta 1 punto de la nota final.

A continuación describimos en las dos secciones siguientes en qué consistió la experiencia realizada durante el curso 2008/09 y el curso 2009/10.

3.1. Curso 2008/09

Durante este curso académico el grupo de profesores que participaban en el PIE reflexionaron sobre los cambios que debían realizar en su asignatura para incluir las actividades de auto- y co-evaluación y diseñaron dichas actividades. La ubicación en el segundo cuatrimestre de la asignatura permitió además alguna experiencia ya en esta asignatura ese mismo curso. El objetivo principal de estas actividades era exploratorio. Se pretendía conocer la opinión de los alumnos acerca de la auto- y la co-evaluación y detectar posibles problemas para mejorar su implantación. Uno de los principales temores de los profesores era que los alumnos percibieran la actividad de co-evaluación como una estrategia para liberar al profesor de sus labores docentes [8]. Por este motivo, el estudio exploratorio de este curso se centró en recabar la opinión de los alumnos sobre la co-evaluación.

La metodología empleada para esta primera actividad de co-evaluación fue la siguiente. Cada alumno debía escoger dos ejercicios de una relación de problemas y debía resolverlos. No existía ningún tipo de restricción en cuanto a la elección de los ejercicios. Posteriormente los ejercicios eran repartidos entre ellos de forma aleatoria (cada ejercicio iba, potencialmente, a un revisor distinto) y corregidos con la ayuda de una solución correcta que el profesor proporcionaba a los evaluadores. El profesor daba a cada alumno evaluador las soluciones detalladas de los ejercicios que debía corregir y debían emitir un juicio de valor acerca de la solución de su compañero. Una vez terminada la fase de evaluación, los alumnos veían la corrección que había

hecho su compañero. El profesor revisaba las evaluaciones para comprobar si se habían hecho con seriedad y sólo si era así la actividad se consideraba realizada. Todo esto se hizo con el apoyo de la herramienta *taller de moodle*.

Tras la actividad se realizó una encuesta pidiendo la opinión de los alumnos. La reacción de los alumnos ante esta actividad fue muy positiva. A todos les gustó la metodología y las únicas críticas fueron a los aspectos técnicos de su implementación (aspectos relacionados con la plataforma de enseñanza virtual) y al hecho de que algunos alumnos recibieron para evaluar las dos veces el mismo ejercicio (de compañeros distintos), con lo cual su aprendizaje era más limitado. Pudimos así comprobar que si el planteamiento de la actividad es el adecuado, la percepción de “descarga de trabajo” para el profesor no tiene por qué producirse.

Podemos destacar dos observaciones realizadas durante el transcurso de esta primera experiencia. En primer lugar, hubo muchos ejercicios que no fueron escogidos por ningún alumno (los más difíciles). Puesto que existía libertad de elección, los alumnos escogieron aquéllos ejercicios que les resultaban más fáciles de resolver. En segundo lugar, se pensó que al ser ejercicios de solución (casi) única, el hecho de proporcionar la solución correcta no ayudaba tanto en el aprendizaje. El alumno podía comparar la solución de su compañero con la proporcionada por el profesor y realizar una evaluación adecuada sin que ello implicara que había entendido correctamente todo el ejercicio. Estos dos aspectos fueron tratados en el curso siguiente, como veremos a continuación.

3.2. Curso 2009/10

Tras la experiencia con la co-evaluación en el curso anterior, durante el curso 2009/10 se diseñó de nuevo la actividad de co-evaluación teniendo en cuenta los dos aspectos mencionados en el párrafo anterior. En particular, para evitar que existieran ejercicios sin resolver se limitó el número de alumnos que podían realizar el mismo ejercicio. De esta forma, los alumnos que querían participar se veían obligados, en ocasiones, a realizar un ejercicio diferente al que originalmente habían planeado y, como consecuencia, todos los ejercicios fueron realizados por alguien. El hecho de trabajar todos los ejerci-

cios, incluyendo los más difíciles, tiene beneficio directo para todo el grupo y no sólo para quien decide escoger los ejercicios más difíciles, ya que los evaluadores de estos ejercicios tendrán que analizar la solución del compañero.

Por otro lado, en esta ocasión el profesor no aportó ninguna solución para que los alumnos evaluaran a sus compañeros, debían ser ellos los que razonaran por qué la solución era correcta o incorrecta. Esta versión de co-evaluación exige en los alumnos destrezas adicionales que no están presentes en la co-evaluación con solución. En particular, el alumno debe ser capaz de comprender, analizar y evaluar la solución del compañero. En comparación con la co-evaluación con solución, que no requiere mucho más que comprensión, esta modalidad alcanza los niveles de análisis y evaluación de la taxonomía de Bloom [2].

En este segundo año de experiencia se incluyó además una actividad de autoevaluación con el objetivo de comparar la acogida de ambas actividades por parte del alumnado. Para esta actividad de autoevaluación los alumnos debían escoger un ejercicio de la relación de problemas y, al igual que se hizo en la actividad de co-evaluación, el número de personas que podía realizar un determinado ejercicio estaba limitado para que todos los ejercicios fueran resueltos por alguien. Cuando los alumnos realizaron sus ejercicios el profesor les mandó la solución al mismo y les pidió un informe razonado de sus errores. Es este informe razonado el que se tuvo en cuenta para considerar como realizada o no la actividad.

Cronológicamente se realizó en primer lugar la actividad de autoevaluación usando la mitad de la primera relación de problemas. Una vez terminada la actividad de autoevaluación se realizó la actividad de co-evaluación usando la segunda mitad de la misma relación de problemas. En esta ocasión cada trabajo era enviado a tres evaluadores y, consecuentemente, cada evaluador tenía que revisar tres trabajos. De esta forma debían trabajar hasta cuatro ejercicios diferentes (de un total de seis posibles).

Tras realizar estas dos actividades se pasó una encuesta preguntando a los alumnos su opinión y su preferencia sobre las actividades. Los resultados de la encuesta indican que, en general, les costó más trabajo evaluar el ejercicio de sus compañeros que el suyo propio. De los 11 encuestados, 6 prefieren la co-evaluación y 5 la autoevaluación. Sin embargo,

cuando se les pregunta con cuál han aprendido más, 7 de los 10 que responden dicen que han aprendido más con la co-evaluación.

Los buenos resultados obtenidos tanto en las encuestas de opinión como en la destreza de los alumnos participantes a la hora de realizar los ejercicios prácticos en el examen final animaron al profesorado a seguir aplicando estas técnicas. En el mismo curso 2009/10 se realizaron dos actividades más de co-evaluación y la actividad ha sido exportada a otras asignaturas impartidas por los profesores.

4. Resultados

Para medir el impacto de las actividades de auto-evaluación y co-evaluación se han realizado dos encuestas anónimas que los alumnos completaron tras la realización de las actividades (y que ya han sido mencionadas en la sección anterior) y se ha estudiado el rendimiento de los alumnos en las preguntas prácticas del examen final, que eran las que se trabajaban en clase con dichas técnicas. Veremos que, en general, los resultados han sido muy positivos, pero debemos tener presente que el número de alumnos que ha participado en las actividades es reducido: 21 en el curso 2008/09 y 17 en el curso 2009/10. Al margen de las posibles consecuencias beneficiosas para el proceso de enseñanza-aprendizaje que puede tener el trabajar con un conjunto reducido de alumnos, la confianza estadística de los resultados es también menor, y los resultados pueden variar considerablemente de un año a otro.

En primer lugar vamos a comentar los resultados de la encuesta que se les hizo sobre la actividad de co-evaluación en el curso 2008/09. En dicha encuesta se les preguntaba, entre otras cosas, cuánto tiempo les había costado hacer los ejercicios, cuánto tiempo les había llevado corregir los del compañero y en qué grado estaban de acuerdo con la nota que les habían puesto. Con respecto al tiempo, afirman que les llevó entre 5 y 30 minutos realizar cada ejercicio y entre 1 y 10 corregir el del compañero. Debemos recordar que, en este caso, el profesor les daba las soluciones para que realizaran la corrección. Con respecto al acuerdo con la nota, la gran mayoría afirmó que estaba completamente de acuerdo.

En dicha encuesta se les dejó un espacio para que escribieran cualquier comentario o sugerencia que tuvieran sobre la actividad. Aquí la mayoría respon-

dió que la actividad había sido muy interesante y que habían aprendido mucho. Algunos alumnos indicaban que la actividad les había obligado a hacer ejercicios. Algunas sugerencias que realizaron fueron la posibilidad de debatir la evaluación realizada por el compañero y la adición de más ejercicios a la relación existente.

En el curso 2009/10, tras la realización de las actividades de autoevaluación y co-evaluación, se realizó una encuesta para preguntarles acerca de dichas actividades. De nuevo en esta encuesta se les preguntó por el tiempo empleado en evaluar el trabajo (propio o ajeno), por los beneficios e inconvenientes de cada técnica, por el acuerdo de la nota del compañero, por la preferencia entre las técnicas y por lo que habían aprendido con cada una de ellas. En dicha encuesta afirmaron tardar entre 2 y 15 minutos en autoevaluarse y entre 10 y 60 minutos en evaluar al compañero (en este caso no tenían la solución correcta). Con respecto a los beneficios de la autoevaluación señalaron la detección efectiva de errores y la obligación de tener que hacer ejercicios. El único inconveniente destacable que señalaron fue la posible falta de imparcialidad al realizar la evaluación. En el caso de la co-evaluación destacaron como beneficios el ver y comprender otras formas de realizar un ejercicio y la diversidad de los ejercicios que tuvieron que tratar. Los inconvenientes que observaron en esta técnica fueron la dificultad de comprender a veces lo que dice el compañero y el miedo a evaluar incorrectamente o a penalizar de forma no adecuada.

Para terminar la encuesta, los alumnos debían indicar qué actividad les había gustado más y con cuál creían que habían aprendido más. El 70% de los alumnos responde que ha aprendido más con la co-evaluación y, de entre éstos, el 40% afirma que le gustó más la autoevaluación. Por otro lado, el 20% de los alumnos responden que les gustó más la autoevaluación y afirman haber aprendido más con ella. Por último, un 10% afirma haber aprendido más con la autoevaluación pero le ha gustado más la co-evaluación.

Con respecto al rendimiento de los alumnos en las preguntas prácticas del examen final, se observó que en el curso 2008/09, primer año en el que se incorporaron estas preguntas, los alumnos obtenían una baja puntuación en ellas. Sin embargo, las puntuaciones aumentaron considerablemente en el examen del curso 2009/10. Esto sugiere que la co-evaluación sin

proporcionar la solución fomenta en el alumno más competencias que cuando se les proporciona una solución correcta.

5. Discusión

En esta sección comentamos algunos aspectos de la experiencia que consideramos de gran interés y que no se han mencionado anteriormente.

Tanto en la autoevaluación como en la co-evaluación uno de los inconvenientes señalados por los alumnos tiene que ver con los criterios de evaluación. Una forma de resolver este inconveniente es mediante el empleo de rúbricas [6], que el alumno debe usar para determinar el grado de consecución de cada uno de los objetivos identificados en los ejercicios. Estas rúbricas se pueden negociar previamente en clase para hacer a los propios alumnos partícipes de su propia evaluación.

En relación con el contexto de la experiencia, ésta se realizó en el seno de un PIE. Los profesores participantes en dicho proyecto aplicaron las técnicas en sus respectivas asignaturas de diferentes titulaciones, pero no lo hacían de forma aislada o independiente del resto. Con cierta frecuencia, los profesores se reunían para discutir acerca de las distintas actividades realizadas en las asignaturas y realizar sugerencias sobre las mismas. Cada vez que un profesor iba a realizar una actividad de auto o co-evaluación convocaba al resto y se realizaba una reunión para discutir y hacer una puesta en común sobre la forma de proceder. Además, en dichas reuniones se creaban las encuestas que posteriormente se les pasaría a los alumnos para recabar la opinión que de dichas actividades tenían.

Estas reuniones de los profesores resultaron ser muy enriquecedoras y podrían considerarse en sí mismas actividades de co-evaluación, ya que en cada reunión una actividad propuesta por un profesor era evaluada por el resto. Pensamos que, al igual que las actividades de co-evaluación fomentan una serie de competencias en el alumno por el hecho de que tenga que evaluar el trabajo de un compañero, la co-evaluación de actividades docentes, como la realizada en este trabajo, también ayuda a los profesores a mejorar su metodología docente.

6. Conclusiones

En la adaptación de las asignaturas al EEES, las herramientas de autoevaluación y co-evaluación pueden facilitar el diseño de un sistema de evaluación formativa y continuada. Estas estrategias constituyen herramientas de aprendizaje y evaluación que presentan ventajas muy beneficiosas: los alumnos ven otras formas de trabajar, desarrollan el hábito de reflexión y de identificación de errores propios y se fomenta el pensamiento crítico. Todos estos aspectos son fundamentales cuando uno de los objetivos en el EEES es desarrollar la competencia del aprendizaje autónomo.

La implementación de esta experiencia ha mostrado que las estrategias de autoevaluación y co-evaluación facilitan la evaluación formativa, permitiendo al alumno y al profesor obtener retroalimentación sobre el proceso de enseñanza-aprendizaje, pero además permiten potenciar muchas otras competencias, habilidades y actitudes que el EEES entiende como enseñanza de calidad. El alumno pasa a ser evaluador, experto y desarrolla la actitud crítica ante su propio aprendizaje.

La experiencia descrita en este artículo es un primer paso en la transformación del sistema evaluativo de la asignatura descrita. Sin embargo, quedan pendientes numerosas cuestiones. Por ejemplo, los alumnos participantes en esta experiencia manifiestan en general aprender más con la co-evaluación y tanto la auto- como la co-evaluación les resultan motivantes y provechosas. Pero no se ha analizado con rigor si el uso de estas herramientas lleva consigo un mayor éxito académico.

Referencias

- [1] A. W. Bangert, Peer Assessment: A Win-Win Instructional Strategy for Both Students and Teachers, J. Cooperation and Collaboration in College Teaching, Vol. 10, No. 2, p. 77, 2001.
- [2] B. S. Bloom. Taxonomy of Educational Objectives: The Classification of Educational Goals. Susan Fauer Company, Inc., 1956.
- [3] S. Brown, A. Glasner, Evaluar en la universidad: problemas y nuevos enfoques. Madrid: Narcea, 2003.
- [4] P. del Canto, I. Gallego, J. Mora, A. Reyes, E. Rodríguez, K. Sanjeevan, E. Santamaría, M. Valero, Evaluación entre compañeros: cómo lo hacemos en nuestros cursos de programación de ordenadores, CIDUI 2010.
- [5] P. del Canto, I. Gallego, J. Manuel López, F. Mochón, J. Mora, A. Reyes, E. Rodríguez, K. Sanjeevan, E. Santamaría, M. Valero, RED: Docencia Universitaria en el contexto de la Sociedad del Conocimiento, No. 1, Septiembre 2010.
- [6] Ideas and Rubric, Instructional Intranet, Chicago Public Schools, http://intranet.cps.k12.il.us/Assessments/Ideas_and_Rubrics/ideas_and_rubrics.html
- [7] J. Kurose and K. Ross, Computer Networking A Top Down Approach Featuring the Internet, Addison Wesley, 5th edition, 2005
- [8] M. Valero, L. M. Díaz de Cerio, Autoevaluación y co-evaluación: estrategias para facilitar la evaluación continuada. Actas del I Congreso Español de Informática, 2005.

Análisis de distintas metodologías de evaluación en prácticas de laboratorio en asignaturas de Redes de Computadores

Jaime Benjumea, Ana V. Medina, Octavio
Rivera, Enrique Dorronzolo

Departamento de Tecnología Electrónica
Universidad de Sevilla
E.T.S Ingeniería Informática
Avda. Reina Mercedes s/n
41012 Sevilla
benjumea@dte.us.es

Resumen

En este artículo se presenta una comparativa entre distintos esquemas de evaluación del alumnado en las prácticas de laboratorio de Redes de Computadores utilizados en los últimos años. Se analizan tres esquemas distintos: 1) Evaluación basada en memorias teóricas y cuestionarios experimentales, 2) Evaluación práctica a práctica basada en WebCT, 3) Evaluación final basada en WebCT. Estos distintos sistemas persiguen como objetivos aplicar técnicas de evaluación continua, lograr un sistema de calificación individualizado y, finalmente, homogeneizar las calificaciones prácticas y teóricas. Se presenta un análisis de cada uno de estos sistemas, en base al cumplimiento o no de estos objetivos usándose, para ello, los datos estadísticos de los seis últimos cursos.

Summary

This paper presents a comparison of three different schemes for evaluating students in a Computer Networks subject. For the last years, we have implemented three different systems: 1) Evaluation based on a previous work (i.e. homework) and a lab questionnaire, 2) Evaluation based on WebCT exams for each session and 3) Evaluation based on one WebCT exam for the lab sessions as a whole. Our objective is to find a system that allows us to: a) use continuous evaluation, b) achieve an individualized evaluation and c) make examination results (practical and theoretical) homogeneous. We present an analysis of these systems to find out if they comply with our objectives. In order to do so, we have used data from the last six years.

Palabras clave

Evaluación del alumnado, prácticas de laboratorio, redes de computadores, WebCT.

1. Introducción

En el ámbito de la Ingeniería Informática, las asignaturas de redes de computadores no solo son materias transversales a todas las intensificaciones de las titulaciones de grado en Ingeniería Informática, sino que cuentan con créditos obligatorios en todas ellas, especialmente en la de tecnologías de la información [1].

Esto significa que se trata, para la mayor parte de los planes de estudios, de asignaturas obligatorias en las que el número de alumnos puede llegar a ser considerablemente alto.

Durante los últimos cursos, hemos estado desarrollando distintas alternativas para la calificación de la parte práctica de dos asignaturas cuatrimestrales (obligatorias) de la titulación de Ingeniería Informática. En estos trabajos se ha analizado la posibilidad de calificación mediante corrección manual [2], mediante cuestionarios con WebCT para cada práctica [3] y mediante cuestionarios WebCT al final de todas las sesiones de prácticas [4].

En todos estos casos se ha intentado buscar un sistema que, por una parte, permita aplicar técnicas de evaluación continua y, por otra, garantice que a la hora de calificar a los alumnos, se evalúen las competencias individuales de éstos y no las de un grupo o una pareja.

Al contrario de lo que sucedía en los trabajos previos, en los que cada sistema se analizaba por

separado, en el trabajo que se presenta a continuación, se analizan las tres alternativas de forma conjunta. Además, los datos estadísticos obtenidos en los últimos seis años permiten analizar, de forma sistemática, las ventajas e inconvenientes de cada sistema.

Este trabajo está organizado como sigue: en el apartado 2 se describen las asignaturas, en el apartado 3 las debilidades del sistema original, mientras que en el apartado 4 se analiza el problema. En el apartado 5 se detallan las tres soluciones que se han adoptado en distintos cursos de las asignaturas analizadas. En el apartado 6 se hace un análisis crítico de las distintas opciones barajadas, análisis que se apoya en los datos estadísticos de seis cursos distintos (apartado 7) y su correspondiente análisis (apartado 8). Finalmente, este trabajo termina con unas conclusiones y una breve descripción de las líneas a seguir.

2. Descripción del sistema original usado en las prácticas de laboratorio

Como suele ser habitual en este tipo de asignaturas, las sesiones de prácticas de laboratorio se distribuyen a lo largo de todo el cuatrimestre, intercalándose con las clases de aula.

Anteriormente, las sesiones de prácticas de laboratorio venían consistiendo en dos horas de trabajo presencial del alumno en el laboratorio más un trabajo previo de preparación de las prácticas. El trabajo previo del alumno (cuyo propósito fundamental era el de hacer que el alumno repasara los conceptos básicos), se evaluaba mediante la corrección de una memoria teórica que el alumno entregaba al finalizar la sesión de laboratorio. El trabajo "in situ" del alumno se evaluaba mediante la corrección de una memoria práctica que el alumno rellenaba a medida que se desarrollaba la práctica y que se entregaba, junto con la memoria teórica, al finalizar la sesión de laboratorio.

En el diseño de las prácticas, tanto el estudio teórico previo como el estudio experimental debían ser realizados por los alumnos de forma individual.

Además, el estudio experimental era auto-contenido, es decir, cumplía un doble objetivo: por una parte presentaba al alumno una serie de instrucciones a seguir, de forma que las pudiera

completar de forma autónoma y a su propio ritmo. Por otra parte, el estudio experimental le planteaba al alumno una serie de cuestiones sobre las que tenía que reflexionar y responder.

Una descripción más completa de la organización de las prácticas de las asignaturas puede encontrarse en [2].

Las asignaturas objeto de estudio son Arquitectura de Redes de Computadores 1 (ARC1) y Arquitectura de Redes de Computadores 2 (ARC2), ambas asignaturas troncales de 4º curso de la titulación de Ingeniería Informática. Ambas asignaturas son cuatrimestrales (primer y segundo cuatrimestre) y no hay incompatibilidades entre las mismas. Es decir, los alumnos matriculados en una y otra asignatura son prácticamente los mismos.

3. Debilidades del sistema original

El sistema original presentaba tres debilidades: dificultad de aplicar sistemas de evaluación continua, dificultades para establecer un sistema de evaluación individualizado y, finalmente, la falta de equilibrio entre las calificaciones de las prácticas y las obtenidas por esos mismos alumnos en los exámenes de teoría.

3.1. Dificultad de aplicar sistemas de evaluación continua

Uno de los problemas detectados consistía en el hecho de que resultaba excesivamente compleja la corrección de los estudios teóricos y los experimentales. Eso, unido a un alto número de alumnos que hacían las prácticas (más de 150 alumnos realizaban 4 prácticas cada uno), hacía imposible que el alumno pudiera disponer de la calificación de cada práctica antes del final del cuatrimestre.

3.2. Dificultad para establecer un sistema de evaluación individualizado

Otro problema detectado consistía en el hecho de que los alumnos desarrollaban tanto el estudio teórico como el experimental en grupo (durante las sesiones de prácticas, de hecho, era habitual que trabajaran en parejas).

Naturalmente, el hecho de que los alumnos colaboren entre sí durante el proceso de aprendizaje no es perjudicial, más bien lo contrario. Pero el problema surgía a la hora de

evaluar a estos alumnos de forma individual porque, en la realidad, los profesores se encontraban estudios teóricos y experimentales casi idénticos.

3.3. Falta de equilibrio entre las calificaciones del examen “teórico” y las prácticas de laboratorio

En las asignaturas objeto de análisis, las prácticas de laboratorio suponen entre un 10% y un 15% de la nota final del alumno, correspondiendo la otra parte a un examen teórico en el que, normalmente, se suelen plantear supuestos prácticos.

Si bien se evalúan competencias diferenciadas, también es cierto que una y otra parte comparten aspectos comunes. Es de suponer, por lo tanto, que la diferencia entre una y otra calificación no debería ser muy grande.

Sin embargo, la sensación que existía entre los profesores de teoría y de prácticas era que esta diferencia era muy grande. Es decir, que el nivel de exigencia y de rigor en una y otra parte de la asignatura no era homogéneo.

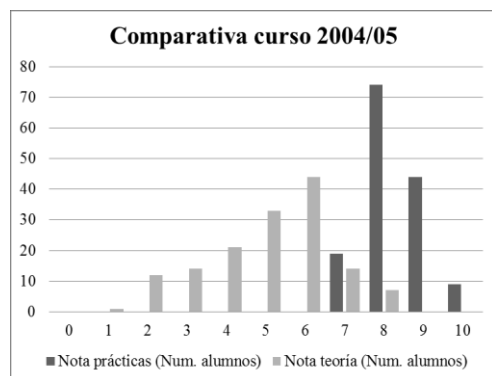


Figura 1. Comparativa calificaciones del curso 2004/05 de la asignatura ARC1

Esta apreciación se puede confirmar con los datos reales de una de las asignaturas, ARC1, cuyos datos comparativos se muestran en la Figura 1.

Como se puede observar, existe una clara diferencia entre la distribución de calificaciones de prácticas y las de teoría.

Puede apreciarse, además, que no solo las calificaciones son claramente distintas, sino que, además, su distribución es distinta. En la Tabla 1,

se muestra un análisis estadístico de los datos que aparecen en la Figura 1.

	Prácticas	Teoría
Media	7,85	4,58
Desviación estándar	0,7589	1,5434
Población	146	146

Tabla 1. Datos estadísticos curso 2004/05

Los datos obtenidos desvelan no solo que la media obtenida por los mismos alumnos en prácticas y en teoría son sustancialmente distintas sino que la desviación estándar en un caso y en otro también son distintas.

Dicho comportamiento se repetía a lo largo de los distintos cursos de esa asignatura (ARC1). Además, esto también sucedía con la asignatura distinta situada en el segundo cuatrimestre: ARC2.

Merece la pena analizar esta cuestión con detenimiento, dado que habría que justificar si esta discrepancia es natural o, por el contrario, era debida a algún problema en nuestro sistema original de evaluar las prácticas de laboratorio.

En concreto, hay dos aspectos que se deben considerar, especialmente:

- Diferencias claras en cuanto al “nivel de dificultad” exigido en una y otra parte: Podía ser esperable unos resultados mejores en las calificaciones de prácticas frente a las de teoría. Pero lo que no parece justificable es que las diferencias sean tan grandes. No parece razonable que, mientras que ningún alumno obtiene una calificación inferior a 7 en las prácticas, un tercio de los alumnos no superaran la parte teórica.
- Diferencias claras en el comportamiento de la distribución de notas: De igual forma, tampoco parece razonable que la desviación estándar en uno y otro caso sean tan diferentes. Una diferencia semejante sería justificable si se tratara de materias distintas o de alumnos distintos, pero este no es el caso. Estas discrepancias, nos llevaron a analizar dónde podría radicar el problema y, desde nuestro punto de vista, concluimos que podía ser debido a la forma en la que se calificaban las prácticas. En concreto, nuestras hipótesis son las siguientes:
 - Las diferencias entre la media en un caso y en otro pueden estar justificadas por el hecho de que ni el estudio teórico ni el experimental

eran, realmente, fruto del trabajo individual del alumno. Además, como se comenta en [3], el propio profesor con sus explicaciones durante la sesión práctica rellenaba, de facto, el cuestionario experimental al que los alumnos debían contestar.

- El hecho de que la desviación típica en la calificación de prácticas fuera sensiblemente inferior a las calificaciones de teoría, podría estar también justificado por el hecho de que los alumnos, en realidad, con calificados “en grupo” lo cual puede llevar a una homogeneización de las calificaciones.

Es decir, creemos que esa discrepancia puede estar motivada por el modo en el que se califica al alumno.

4. Análisis del problema

Por tanto, la metodología usada en las prácticas nos llevaba, principalmente, a dos problemas fundamentales.

Por un lado, la dificultad que existía con el sistema antiguo de aplicar sistemas de evaluación continua en los que el alumno pudiera observar su progreso en esta parte de la asignatura.

Por otro, la dificultad de calificar a los alumnos de forma individualizada dadas las múltiples interacciones que se producían tanto en el trabajo previo que los alumnos realizaban, como durante la práctica de laboratorio. Esto, además, podía llevar a una discrepancia entre las notas de teoría y las de prácticas que habría que analizar.

5. Sistemas utilizados

5.1. Primer sistema: Corrección manual abreviada

Uno de los primeros cambios que se realizó consistió en intentar aplicar métodos de evaluación continua en las prácticas. Dicho sistema se aplicó en la asignatura ARC1 durante el curso 2006/07, como parte de un proyecto de innovación docente [2].

Para lograr los objetivos de aquel proyecto, se procedió a modificar tanto los cuestionarios teóricos como, sobre todo, los cuestionarios experimentales. El objetivo consistía en mantener un sistema de estudios teóricos y experimentales

pero modificar éstos para que su corrección pudiera ser sistemática.

Las conclusiones de este proyecto fueron claras:

- Aplicar este sistema de evaluación continua no tenía incidencia en el abandono de los alumnos: uno de los temores a la hora de aplicar sistemas de evaluación continua era que desanimaran a los alumnos a continuar. Los resultados obtenidos demostraron que eso no ocurría.
- Dificultad a la hora de aplicar sistema de evaluación continua si se hace una corrección manual: también se constató que, a pesar de haber sistematizado la corrección de las prácticas, resultaba muy difícil llevar a cabo el objetivo de que los alumnos pudieran tener disponible su calificación antes de comenzar la siguiente sesión de prácticas (se hacían prácticas cada dos semanas).

A pesar de que los resultados obtenidos no fueron satisfactorios, sí se logró redactar unos cuestionarios experimentales que fueran fáciles de corregir. No obstante, la dificultad a la hora de calificar a los alumnos, aún existía.

Por tanto, el siguiente paso que decidimos dar consistió en convertir esos cuestionarios “fáciles de corregir” en una serie de cuestiones que se plantearan al alumno en forma de prueba objetiva y que, por tanto, se pudieran trasladar a una plataforma de enseñanza virtual como WebCT¹.

5.2. Segundo sistema: uso de WebCT con calificación práctica a práctica

Esta iniciativa, consistía, básicamente, en mantener los estudios teóricos y experimentales del curso anterior pero, en este caso, basar la calificación del alumno en un cuestionario que ellos mismos rellenaban de forma individual en los 15 últimos minutos de las dos horas que duraba cada sesión de prácticas.

Aunque este sistema se describe con detalle en [3], comentaremos a continuación algunos detalles de la misma así como las conclusiones principales a las que se llegó.

¹ El uso de WebCT no indica ninguna preferencia de una plataforma de enseñanza virtual sobre otras. Simplemente se usa esta plataforma porque es la que está oficialmente soportada en nuestra Universidad.

La metodología consiste en mantener, con las debidas modificaciones, la existencia del estudio teórico y del estudio experimental para cada una de las prácticas. Pero, a diferencia de la metodología aplicada hasta entonces, tanto las respuestas al estudio teórico como las del estudio experimental formarán parte de los apuntes personales del alumno. Es decir, el profesor de prácticas no recoge ni el estudio teórico ni el experimental.

Para poder valorar el grado de aprovechamiento individual del alumno en la práctica, todos los alumnos rellenan a través de la plataforma WebCT, un cuestionario de evaluación de 10 preguntas. El sistema se ha diseñado de forma que exista una batería suficientemente amplia de preguntas, con el fin de que los alumnos no tengan información previa del contenido del cuestionario. Los alumnos deben rellenar el cuestionario de forma individual, pero se les permite usar el material bibliográfico que deseen.

Este sistema presenta, desde el punto de vista de la calificación de los alumnos, las siguientes ventajas:

- Permite que las sesiones de laboratorio se desarrollen de una forma más dinámica con lo que se logra una mayor implicación del profesor en el proceso de aprendizaje y fomenta el aprendizaje colaborativo en los alumnos, sin que ello “contamine” el proceso de calificación (con el sistema anterior, el alumno podía trasladar la respuesta que el profesor o un compañero le acabada de dar, al cuestionario que estaba rellenando, sin entender realmente la respuesta).
- Añade un mayor grado de equidad a la hora de evaluar las prácticas dado que, al rellenarse los cuestionarios de evaluación de forma individual, reflejan directamente el grado de aprovechamiento de cada alumno, independientemente de cómo ha adquirido las competencias objeto de evaluación (por iniciativa propia, por la ayuda de otros compañeros o del profesor, etc).

Este es el sistema que se ha venido aplicando en la asignatura ARC1 desde el curso 2007/08.

5.3. Tercer sistema: uso de WebCT con calificación al final de todas las sesiones de prácticas

Los resultados satisfactorios obtenidos en la asignatura ARC1, nos plantearon aplicar la misma metodología a la otra asignatura de redes: ARC2.

Pero, a diferencia de lo que ocurre con la asignatura ARC1, decidimos aplicar una metodología ligeramente distinta. Las diferencias básicas entre este sistema y el anterior se pueden enumerar en las siguientes:

- Se realiza una única prueba de evaluación: en efecto, se sustituye cada una de las pruebas de evaluación de cada práctica por una prueba global al final del cuatrimestre (se requiere que hayan asistido a las sesiones prácticas). Esta prueba de evaluación global tiene una hora de duración, abarca todas las prácticas realizadas, y, al igual que sucedía con las pruebas por práctica, se permite que el alumno use el material bibliográfico que desee.
- El alumno, al contrario de lo que sucedía en el otro sistema, sí se puede llevar el estudio experimental a su casa, una vez realizada la sesión de prácticas.

En el siguiente apartado se justificará las razones que nos han llevado a analizar también esta alternativa.

6. Análisis de las tres soluciones

En [2] se indicaba, a modo de conclusión, que el sistema de calificación manual de las prácticas de laboratorio no era posible llevarlo a cabo con los recursos de profesorado disponibles.

Por otra parte, en lo que respecta al segundo sistema, es necesario señalar que se ha aplicado con éxito durante los últimos cuatro cursos. Este sistema permite, de forma evidente, evitar las dificultades señaladas como 3.1 (evaluación continua) y 3.2 (calificación individualizada). Falta por analizar si también logra evitar la dificultad 3.3 (calificación desequilibrada), análisis que se realizará seguidamente.

Respecto al tercer sistema, éste tiene, a priori, las mismas ventajas que el segundo sistema. No obstante, la ausencia de una calificación para cada práctica, hace que no se aplique un sistema de evaluación continua. Existen dos razones para considerar este sistema:

Por un lado las “quejas” de los alumnos respecto a un sistema en el que se les califica en cada práctica, por la presión que ello suponía sobre los alumnos (si bien es cierto que siempre fueron quejas informales).

Por otro lado, porque creemos que un sistema de evaluación global permite abarcar todas las prácticas a la vez y eso, desde nuestro punto de vista, se asemeja más a la labor profesional que desarrollarán nuestros egresados (al enfrentarles a un problema global, frente a uno parcial).

En cualquier caso, la creación de cuestionarios de autoevaluación por cada práctica, permitirían incorporar a este tercer sistema un esquema de evaluación continua.

Analizaremos, por tanto, el segundo y tercer sistema, dado que el primero no es posible llevarlo a cabo.

7. Datos estadísticos utilizados

Como ya se ha comentado reiteradamente, se han analizado dos asignaturas distintas, ARC1 y ARC2. En la primera se ha utilizado el sistema basado en evaluaciones realizadas para cada práctica, mientras que en la segunda asignatura se ha implantado el sistema de calificación global.

Ambas asignaturas pertenecen al mismo curso y, dado que se trata de asignaturas troncales, la población de referencia es similar.

Por otra parte, al recaer la coordinación de la asignatura en la misma profesora, la metodología de las clases de teoría y de laboratorio es muy parecida (salvo las diferencias de temario). El profesorado de teoría y de laboratorio ha permanecido estable durante todo el periodo observado.

Para el cálculo de la nota de teoría y de prácticas, solo se ha considerado a aquellos

alumnos que se han presentado a la parte teórica de la asignatura (ya sea mediante un sistema de evaluación continua de la parte teórica o al examen final) y que, además, han realizado todas las prácticas de laboratorio. La observación se restringe a la primera convocatoria de cada curso académico.

Se han analizado datos estadísticos de los cursos 2004/05 a 2009/10 para el caso de ARC1, mientras que para el caso de ARC2 se han usado datos desde el curso 2006/07 a 2009/10 (debido cambios metodológicos y de profesorado).

Los hitos históricos en la aplicación de estos sistemas fueron los siguientes:

- Curso 2006/07: Se implanta el primer sistema en ARC1.
- Curso 2007/08: Se implanta el segundo sistema en ARC1.
- Curso 2008/09: Se implanta el tercer sistema en ARC2.

8. Análisis de los datos estadísticos

El objetivo del análisis estadístico es el de comprobar cómo se comportan los dos sistemas “supervivientes” en relación al equilibrio entre las calificaciones obtenidas en las prácticas y en teoría.

8.1. Análisis de los datos de ARC1

Los datos estadísticos de ARC1 se muestran en la Tabla 2. Los datos de los cursos 2004/05 y 2005/06 se corresponden con los cursos en los que se aplicaba el sistema antiguo y pueden observarse las diferencias entre las calificaciones medias de las prácticas y la teoría y sus correspondientes desviaciones típicas.

	04/05		05/06		06/07		07/08		08/09		09/10	
	NP	NT	NP	NT	NP	NT	NP	NT	NP	NT	NP	NT
Media aritmética	7,85	4,58	6,86	3,68	7,09	3,52	6,54	4,10	6,07	3,98	6,61	4,35
Desviación estándar	0,76	1,54	0,94	1,97	1,02	1,59	0,97	1,56	0,99	1,71	1,02	1,72
Población	146		178		135		154		135		153	

Tabla 2. Datos estadísticos de la asignatura ARC1

En el curso 2006/07, año en el que se implantó el primer sistema, se observa que se logra obtener un comportamiento más parecido en lo que respecta a la dispersión de las notas pero, aun así, la diferencia entre las medias sigue siendo excesiva.

Durante los cursos siguientes, durante los cuales se utilizó el sistema descrito en el apartado 5.2 (WebCT, evaluación en cada práctica), se puede observar que no sólo las desviaciones estándar toman valores más cercanos sino que, además la diferencia media entre las calificaciones de prácticas y teoría se reduce a valores en torno a 2,3 puntos.

Si analizamos detenidamente la distribución de calificaciones correspondientes al curso 2009/10 (Figura 2), puede observarse que, aun siendo las calificaciones de prácticas superiores a las de teoría, las diferencias entre estas parecen más adecuadas.

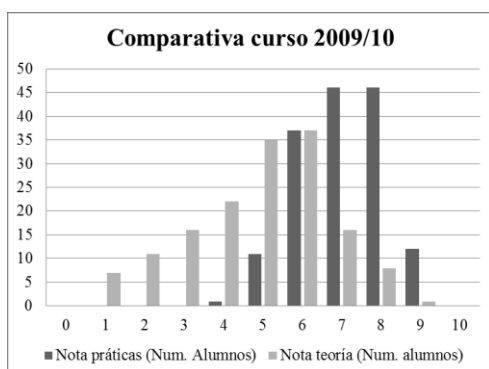


Figura 2. Comparativa calificaciones del curso 2009/10 de la asignatura ARC1

En este caso, el porcentaje de alumnos que no superan la parte teórica es similar al del curso 2004/05 pero, sin embargo, el porcentaje de alumnos que obtienen 7 o más en las prácticas se reduce desde el 100% observado en el curso 2004/05, al 65% para el curso 2009/10.

Es cierto que se obtienen peores calificaciones en las prácticas pero, desde nuestro punto de vista, la implantación del nuevo sistema de calificación de las prácticas hace que éstas sean coherentes con el nivel de exigencia o dificultad de la teoría.

Creemos que este “aumento de la dificultad” de las prácticas es beneficioso puesto que le sirve al alumno para conocer su nivel de desempeño en la asignatura. No obstante, también es cierto que no se aprecia una mejora significativa del

rendimiento de los alumnos en la parte teórica de la asignatura.

8.2. Análisis de los datos de ARC2

En la asignatura ARC2, originalmente existía el mismo sistema de calificación de las prácticas que en ARC1. Los datos estadísticos correspondientes a los cursos previos muestran un comportamiento muy parecido al observado en ARC1, con una media en prácticas de 7,10 y una desviación de 0,95, mientras que los datos de teoría era 3,93 (media) y 1,78 (desviación).

En el curso 2008/09 se puso en marcha un sistema similar al de ARC1 excepto por el hecho de que se hacía una única prueba global de prácticas. Los datos estadísticos de los cursos en los que se ha aplicado el nuevo reflejan un comportamiento similar al de ARC1.

Curso	08/09		09/10	
	NP	NT	NP	NT
Media	4,69	3,71	4,05	4,25
Desviación	1,40	1,76	1,51	1,79
Población	114		115	

Tabla 3. Datos estadísticos ARC2. Sistema nuevo

Sin embargo, si que se observa algo distinto, consistente en que las diferencias entre las calificaciones de prácticas y las de teoría son muy pequeñas, llegando incluso a ser ligeramente inferiores, tal y como es muestra en el histograma correspondiente al curso 2009/10 (Figura 3).

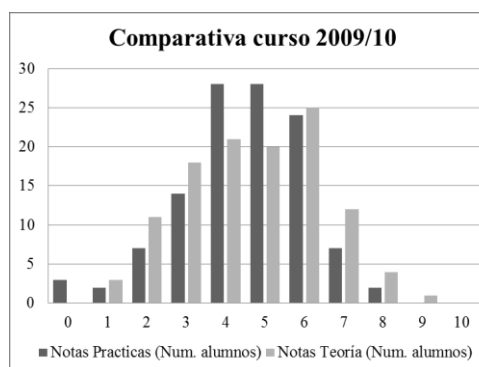


Figura 3. Comparativa calificaciones del curso 2009/10 de la asignatura ARC2

Por tanto, la cuestión que queda por analizar es si los resultados obtenidos en ARC2 son más congruentes que los obtenidos en ARC1, y si uno

u otro sistema tienen incidencia en el rendimiento de los alumnos en las calificaciones de teoría.

Aún no disponemos de datos suficientes que permitan llegar a algún tipo de conclusión (respecto a qué sistema es mejor: el de ARC1 o el de ARC2) más allá de la sensación que puedan tener los profesores de la asignatura. No obstante, como se indicó anteriormente, la idea de disponer de una evaluación global, que permita evaluar de forma conjunta todas las competencias prácticas del alumno (ARC2), parece una buena opción.

Tampoco disponemos de datos suficientes como para poder afirmar que los alumnos obtienen mejores resultados y, por tanto, aprenden más, en un sistema frente a otro. Será necesario disponer de datos de los cursos venideros para poder hacer un análisis formal.

9. Conclusiones

En este trabajo se han presentado una serie de iniciativas respecto a la calificación de las prácticas de las asignaturas ARC1 y ARC2. El objetivo principal de estas iniciativas es lograr que la calificación de las prácticas de la asignatura, se haga en base a las competencias individuales del alumno sin que ello menoscabe las posibilidades que ofrece el aprendizaje colaborativo. Además, se intenta que las calificaciones entre las notas de teoría y las de prácticas guarden cierta coherencia.

Se ha demostrado que un sistema basado en la corrección manual no es útil a la hora de conseguir los objetivos perseguidos, incluso si se sistematiza la corrección.

Por otra parte, un sistema basado en calificar las prácticas mediante cuestionarios de evaluación con WebCT, consigue alcanzar el objetivo de tener una calificación individualizada.

Además, a la vista de los datos estadísticos obtenidos, al calificar a los alumnos de forma individualizada hay una tendencia a que los resultados obtenidos por los alumnos en sus calificaciones de teoría y prácticas sean más homogéneos por lo que, desde nuestro punto de vista, parecen confirmarse las hipótesis señaladas en 3.3 y nos lleva a concluir que debe existir cierta homogeneidad (aunque no igualdad) entre los datos de las calificaciones de teoría y prácticas.

Finalmente, se ha establecido una comparativa entre los resultados obtenidos en ARC1 frente a

los de ARC2, observándose que en el segundo caso (donde se hace una única prueba de evaluación global) las diferencias entre las notas de teoría y de prácticas son casi insignificantes.

No obstante, no disponemos de datos suficientes que permitan asegurar que el sistema aplicado en ARC1 sea mejor o peor que el de ARC2.

10. Trabajo futuro

Como trabajo futuro, proponemos realizar un análisis estadístico de los datos de los próximos cursos, así como incorporar al análisis los resultados de los exámenes de segunda convocatoria.

Por otra parte, también estamos planteando la posibilidad de añadir cuestionarios de autoevaluación para cada práctica en la asignatura ARC2, a fin de incorporar medidas de evaluación continua a este sistema.

El objetivo principal será el de comprobar qué esquema permite que los alumnos afronten con mayor éxito el examen de teoría.

Referencias

- [1] BOE 4-ago-2009. *Recomendaciones para los títulos de Ingeniería Informática*.
- [2] Medina, A.V., Benjumea, J., Estrada, A., Barbancho, A. Aplicación de medidas de seguimiento y evaluación continua en las prácticas de Arquitectura Redes de Computadores I (ARCI) y Comunicaciones I (C1). Colección "Innovación y Desarrollo de la Calidad de la Enseñanza Universitaria" nº17, Experiencia de Innovación Universitaria (I) Curso 2006-2007,453-451. ISBN:978-84-86849-70-2.
- [3] Benjumea J., Medina, A.V., Estrada, A., Barbancho, A. *Evaluación de las Prácticas de Redes de Computadores Mediante Cuestionarios on-Line a Través de Webct*. Jornadas de Enseñanza Universitaria de la Informática (Jenui2008).
- [4] Medina, A.V., Benjumea, J., Barbancho, J., Estrada, A., Rivera, O. *Evaluación mediante WebCT de las competencias prácticas e instrumentales de la asignatura ARCII*. No publicado

Sesión 6B:
Métodos pedagógicos innovadores III

Tutorizando el aprendizaje proactivo de nuevas tecnologías: Taller de programación Android*

Maria J. Blesa Amalia Duch Joaquim Gabarró Hugo Hernández Maria Serna

Departament de Llenguatges i Sistemes Informàtics

Facultat d'Informàtica de Barcelona

Universitat Politècnica de Catalunya

Campus Nord, 08034 BARCELONA

{mjblesa, duch, gabarro, hhernandez, mjserna}@lsi.upc.edu

Resumen

Mediante el Programa Google EMEA's AndroidEDU, la UPC ha impartido un novedoso taller de programación para Android.

Durante diez semanas se tratan diferentes temas sobre este sistema y una veintena de alumnos trabajan ejercicios en grupos reducidos. Los grupos colaboran para investigar y solucionar las dificultades técnicas que surgen. Posteriormente, cada grupo realiza y defiende un proyecto propio.

El interés en Android ha permitido fomentar el aprendizaje proactivo, con muy buenos resultados. Como profesores esta experiencia nos ha permitido afrontar nuevos retos docentes, obligándonos a replantear nuestro rol, a crear material docente adecuado a los nuevos medios de difusión social (foro FIB, YouTube, etc.), y a suplir la falta de experiencia en el tema con una interesante colaboración profesor-alumno.

Summary

Within the AndroidEDU Google EMEA Program, the UPC has organized a innovative programming workshop for Android.

For ten weeks, twenty students (who are organized in small groups) deal with different topics about Android. Those groups work together to investigate and resolve technical difficulties that may arise. By the end of the course, each group works on and fends for its own project.

*Trabajo parcialmente financiado por la mención de la Generalitat de Catalunya a ALBCOM como grupo de investigación consolidado (ref. 2009 SGR 1137) y por los proyectos del Ministerio de Ciencia y Tecnología: TIN2007-66523 (FORMALISM) para M. Blesa, J. Gabarró y M. Serna, y TIN2010-17254 (FRADA) para A. Duch, H. Hernández está financiado por el Comissionat per a Universitats i Recerca del DIUE de la Generalitat de Catalunya y por el European Social Fund.

The growing interest in Android has allowed us to apply proactive learning techniques with very good results. As teachers, this experience has allowed us to meet new challenges, since it has forced us to rethink our role, to create educational material accordant with the new communication media (forum FIB, YouTube, etc.), and to supply the lack of expertise with an interesting collaboration between teachers and students.

Palabras clave

Google AndroidEDU, Android, taller de programación, nuevas tecnologías.

1. Motivación

A través del programa *AndroidEDU EMEA*[2] la empresa *Google* lanzó una convocatoria dirigida a las universidades con el propósito de fomentar el desarrollo de experiencias docentes utilizando el sistema *Android* y específicas para teléfonos móviles.

Para participar en el programa era necesario presentar una propuesta de desarrollo de un sistema novedoso en ingeniería móvil que requiriera un número limitado de teléfonos pero que al mismo tiempo pudiera alcanzar a una amplia audiencia. El sistema propuesto debía ser escalable y la propuesta debía facilitar el hacer la ciencia de la computación tangible para los estudiantes.

La propuesta sometida a dicho programa por los autores de este trabajo resultó ser una de las seleccionadas y para facilitar su puesta en marcha *Google* le donó 20 teléfonos al departamento de Llenguatges i Sistemes Informàtics (LSI) de la Universitat Politècnica de Catalunya (UPC).

Este es el origen del *Taller de Programación de Aplicaciones Android para Google Phones* con acrónimo TPAAGP objeto de este trabajo. Se trata de

una asignatura de libre elección (ALE) que ofrece la UPC a alumnos de diferentes titulaciones. El cupo de estudiantes en el taller está restringido al número de teléfonos del que disponemos. TPAAGP es un curso muy práctico de 4-ECTS que requiere que los alumnos que lo cursan tengan un buen conocimiento del lenguaje de programación Java y una buena cultura general de algoritmos y estructuras de datos.

Tras dos cuatrimestres impartiendo este taller, consideramos que se trata de una experiencia a compartir ya que los resultados han sobrepasado nuestras expectativas. Primero, los estudiantes se muestran mucho más proactivos y motivados que en otros cursos más convencionales y los proyectos resultantes, tanto en sus contenidos como en su presentación, son de gran calidad. Segundo, hay un cambio de rol en el papel del profesor, ya que actúa más de guía (*coach*) que como profesor convencional. Tercero, al utilizar tecnología punta este taller aporta aires nuevos a contenidos clásicos como son la programación y las estructuras de datos. A la vista de los resultados obtenidos, afirmamos que las nuevas tecnologías son un elemento motivador muy útil en la enseñanza universitaria. Creemos que este es un factor todavía por explotar en algunas áreas.

A continuación describimos el curso (en la Sección 2) dando los contenidos, el método de evaluación y el material de consulta y soporte utilizado para su impartición. Seguidamente, la Sección 3 describe la valoración del curso realizada en la segunda edición del curso por parte de los estudiantes. El trabajo concluye con la Sección 4, donde reflexionamos sobre diversos aspectos que consideramos novedosos dentro del ámbito universitario y que creemos que pueden llegar a ser útiles dado que, al menos para este curso, ya lo han sido.

2. El curso

El curso TPAAGP dura diez semanas y su orientación, como ya se ha dicho, es muy práctica desde el principio. Como se supone que los estudiantes tienen conocimientos previos de programación orientada a objetos utilizando Java, el curso se puede enfocar desde el principio en los detalles de programación del sistema Android, motivo por el cual, el curso es realmente un taller de programación.

TPAAGP está pensado para ser altamente interactivo tanto entre profesores y estudiantes como entre

estudiantes. El taller está organizado de tal manera que cada semana se explica un tema nuevo a través de distintos ejemplos, con énfasis en la implementación. La complejidad de los temas tratados semanalmente se incrementa gradualmente a lo largo del curso. Al final de cada sesión se deja un ejercicio de programación en el que los estudiantes deben trabajar durante la semana para, en la siguiente sesión, proponer una solución que será discutida y comentada entre los demás estudiantes y los profesores. Estos trabajos conforman una evaluación continuada de los estudiantes que se completa con el desarrollo de un proyecto de dificultad media en el que se combinan y aplican todos los conocimientos adquiridos durante el desarrollo del taller.

2.1. Contenidos

A continuación damos los detalles sobre las sesiones que conforman el curso.

1a sesión: Introducción. En la primera sesión del taller se muestra cómo instalar el entorno de programación y todas las herramientas de desarrollo. Se introduce también la filosofía de los proyectos en Android a través de un primer programa muy sencillo (la versión Android del clásico programa *Hola mundo*) y se muestra cómo utilizar unas cuantas herramientas (widgets) básicas de las aplicaciones de Android como son, por ejemplo, los botones, los menús, etc. Como trabajo para esta sesión se pide a los estudiantes un programa para jugar a la piedra, tijeras y papel. Se trata de un programa sencillo que permite utilizar algunas de las herramientas mencionadas y que a los estudiantes les resulta muy motivador. En particular se muestran muy creativos en la presentación gráfica del juego.

2a sesión: Widgets y ficheros. Las aplicaciones y los widgets conforman la capa más alta en la arquitectura de los proyectos del sistema Android. En la segunda sesión se profundiza en la estructura de dichas aplicaciones y se presentan elementos avanzados tales como las listas tanto estáticas como dinámicas (incluidas por defecto en el sistema) que facilitan su desarrollo.

Por otro lado, la mayor parte de las aplicaciones reales necesitan almacenar información y Android

permite hacerlo en el dispositivo móvil mediante diferentes técnicas. La más sencilla, el manejo de ficheros, se introduce en esta sesión. El programa que se pide al finalizar la sesión es desplegar y fusionar de manera dinámica listas de contactos leídas de diversos ficheros.

3a sesión: Localización. El tema de la tercera semana es la localización, incluyendo la utilización del GPS del teléfono y de aplicaciones web híbridas que combinan mapas con otra información. En particular, se muestra cómo obtener y actualizar la localización del teléfono, cómo utilizar el emulador para simular cambios de posición en el entorno de programación y cómo, con pocas líneas de código, se puede incluir un `MapView` (clase específica de Android para la visualización de mapas) en una aplicación. Cabe destacar la conexión de la clase `MapView` con todo el potente sistema de Google Maps. En efecto, esta clase requiere acceso a Internet para poder llamar a los servidores de Google y así obtener los mapas (que no dejan de ser propiedad de Google). El programa que se pide al finalizar la sesión consiste en desplegar mapas en la pantalla del teléfono y mostrar puntos y trayectorias específicas en él.

4a sesión: Bases de datos. En la segunda sesión se explicó como almacenar datos utilizando ficheros. Este sistema funciona bien cuando la cantidad de datos es pequeña o la información es del mismo tipo. Sin embargo para almacenar grandes cantidades de datos estructurados es mejor utilizar una base de datos relacional. Android incluye un pequeño sistema de base de datos: SQLite [17] al que se le dedica esta semana del taller. Se trata de un pequeño y potente motor de base de datos SQL muy difundido que también se utiliza en PHP, iPhone, Skype, Mozilla Firefox, etc. Como trabajo de esta sesión se pide integrar en bases de datos parte de los conocimientos adquiridos en ejercicios previos, como por ejemplo gestionar localizaciones generadas en la Sesión 3 almacenándolas en la base de datos del sistema.

5a sesión: Threads. El tema de la quinta semana de taller son los *threads*, una característica que Android hereda del lenguaje de programación Java y que permite la ejecución en paralelo de varias tareas. Esta característica es especialmente útil en la

programación de teléfonos móviles ya que los usuarios de estos sistemas se vuelven cada vez más impacientes y quieren aplicaciones que respondan de manera instantánea, lo que es difícil de conseguir pues los recursos de un teléfono son limitados. Mediante los *threads* es posible, por ejemplo, abrir una ventana de diálogo con el usuario para avisarle que la aplicación está realizando la operación requerida, lo que ayuda a disminuir su impaciencia. El trabajo de la sesión consiste en transformar una de las aplicaciones previamente programadas para que utilice *threads*, por ejemplo, programar una aplicación que cada determinado tiempo pida actualizar la localización del teléfono sin interrumpir la ejecución de otras aplicaciones.

6a sesión: Gráficos para el diseño de juegos. Durante la sexta sesión se introducen los aspectos fundamentales de las librerías gráficas de Android. El ejercicio de esta sesión se basa en el desarrollo de un juego de tenis. Durante la sesión, cada grupo de estudiantes desarrolla una parte de los gráficos del juego de manera que al final todos disponen de la versión completa. Como trabajo semanal se proponen otros juegos.

7a sesión: Comunicaciones entre procesos. Un tema de particular importancia en Android es el de las comunicaciones internas del sistema y la conforman los *intents* (mecanismos de comunicación entre distintas aplicaciones del sistema) y las llamadas a procesos remotos (con acrónimo en inglés RPCs).

Tanto los *intents* como las RPCs fomentan de manera muy eficaz un alto grado de modularidad y de independencia en las aplicaciones. En Android prácticamente todo funciona por medio de estos mecanismos, de modo que es muy fácil reutilizar o sustituir componentes. Por ejemplo, si una aplicación requiere enviar un SMS y hay un *intent* para ello, éste puede ser utilizado por la aplicación. Para consolidar la sesión se pide crear una aplicación que pueda imprimir en la pantalla del dispositivo móvil datos de tipo texto provenientes de *intents* creados por otras aplicaciones.

8a sesión: Conexiones y navegación. Además de su utilidad básica de realizar llamadas, el teléfono se utiliza cada vez más como dispositivo móvil para ac-



Figura 1: Teléfonos provistos por Google

ceder a Internet. Android está perfectamente equipado para estas funcionalidades ya que proporciona un navegador web muy completo basado en el proyecto de código abierto WebKit [18]. Se trata del mismo motor que puede encontrarse por ejemplo en Google Chrome y Safari. Además, se explican funcionalidades que Android proporciona a sus programas de acceso a servicios estándar de red como son los sockets TCP/IP, las conexiones HTTP mediante el lenguaje JavaScript o las conexiones Bluetooth entre teléfonos. Como trabajo se pide programar aplicaciones que se conecten mediante JavaScript a diversas páginas web y que se instale y se pruebe en los teléfonos un chat de Bluetooth proporcionado en [13].

9a sesión: Gestión de SMS y llamadas. La gestión de SMS y llamadas en Android es ya muy completa y central para las aplicaciones de telefonía. Los trabajos prácticos de esta sesión son, por ejemplo, sistemas de alarma que se activan y desactivan con llamadas o SMS, sistemas de votaciones y/o encuestas, o un juego de batallas navales.

10a sesión: Distribución de proyectos. En esta sesión se entregan y discuten los proyectos finales y se atiende a una charla invitada de alguna empresa del sector.

2.2. Material

Para la impartición de este curso se ha recurrido a una cantidad significativa de material docente y de soporte. Dicho material se describe a continuación.

Teléfonos Google. El Programa AndroidEDU de Google dotó nuestra propuesta con 10 teléfonos An-

droid Dev 1 Phone y 10 teléfonos Nexus One (véase Figura 1).

El Android Dev Phone 1 es una variante de HTC T-Mobile G1 diseñado exclusivamente como herramienta para desarrolladores de aplicaciones para el sistema operativo Android de Google. Ambos teléfonos funcionan con Android (1.0 en Dev Phone 1 y 2.2 en Nexus One), tienen Wi-Fi (802.11 b/g), Bluetooth (Bluetooth 2.0), radio GSM cuatribanda y USB (USB 2.0). La capacidad de almacenamiento es sensiblemente mayor en el Nexus One (512MB RAM, microSD card de 4GB y hasta 32GB), que en el Dev Phone 1 (192MB, microSD card de 1 GB y hasta 16 GB), así como su potencia de procesamiento (CPU de 1 GHz vs. CPU de 528 MHz). Para más detalles sobre las especificaciones y detalles de estos teléfonos, véase [4] para el Android Dev Phone 1, y [14] para el Nexus One.

Aunque el kit de desarrollo de software (*Software Development Kit (SDK)* en inglés) para Android incluye un emulador aceptablemente bueno, para los estudiantes siempre resulta muy atractivo e interesante disponer de teléfonos reales sobre los que probar sus aplicaciones. Si además son teléfonos de última generación (y, por tanto, de precio bastante elevado en el mercado), que todavía no están a disposición del público en general (como ha sido nuestro caso), la expectación y el interés que despiertan son remarcables. Poder disponer de tal hardware ha sido un plus importante para este curso.

Bibliografía. Cuando impartimos la primera edición del taller había muy pocos libros de Android. El curso ha sido preparado basándonos principalmente en los siguientes dos libros:

- M. Murphy. *Beginning Android*. Apress Ed, 2009.
- Z. Mednieks R. Rogers, J. Lombardo and B. Meike. *Android Application Development*. O'Reilly Media Ed., 2009.

Ambos son muy prácticos, suponen un buen conocimiento del lenguaje de programación Java, están muy enfocados a la parte de implementación, incluyen muchas discusiones acerca del código de diversas aplicaciones y sobre cómo, progresivamente ir mejorando la eficiencia de los programas y adecuándolos a las restricciones específicas que imponen las

limitaciones de memoria y batería de los dispositivos móviles.

Una característica importante de estos dos libros es que las editoriales ofrecen por Internet el código fuente de todos los programas que en ellos se muestran, de manera que es muy fácil acceder a ellos para probarlos, modificarlos y reutilizarlos.

En apenas un año, se ha incrementado notablemente la bibliografía y el material electrónico de soporte al aprendizaje de Android. Actualmente estamos consultando y contrastando las referencias [8, 7, 3, 11, 12] para considerar usarlas en futuras ediciones del curso.

En soporte electrónico también existe mucho material acerca de Android (guías, vídeos, etc.), siendo los foros de desarrolladores posiblemente los más abundantes. Un recurso electrónico imprescindible en este curso es la guía en línea para desarrolladores de Android que ofrece Google: *Android's Developer's Guide* [5]. Esta guía es muy completa en cuanto a que contiene todo lo que forma parte del sistema pero, al mismo tiempo, es poco explicativa y a menudo resulta difícil encontrar una información específica debido al gran volumen de datos que contiene.

Sitio web y Foro en línea. Todos los contenidos del curso, incluyendo referencias bibliográficas, enlaces a recursos electrónicos y proyectos realizados en ediciones anteriores del curso pueden encontrarse en <http://albcom.lsi.upc.edu/tpaagp/>.

Desde esta página web, los estudiantes y profesores también pueden acceder a un foro en línea en el cual compartir sus dudas y soluciones, hacer comentarios, etc. Dicho foro es uno de los que la Facultat d'Informàtica de Barcelona gestiona y pone a disposición de sus estudiantes y profesores dentro de la herramienta web *Racó* [16].

2.3. Evaluación

La evaluación en TPAAGP se divide en dos partes. La primera es una evaluación continuada de los trabajos semanales. La segunda es la evaluación del proyecto final.

Para la evaluación de los trabajos semanales se les pide a los estudiantes (por equipos de dos o tres) que enseñen y expliquen en cada sesión el trabajo encargado en la sesión anterior. Además, cada equipo de-

be hacer un informe semanal explicando cómo han hecho el trabajo, con qué dificultades se encontraron, cómo las resolvieron y, si no lo hicieron, cómo creen que lo pudieron haber hecho. Con este informe semanal los profesores podemos evaluar el trabajo realizado durante las sesiones y adecuar contenidos.

Para evaluar el aprovechamiento, y como ya habíamos mencionado anteriormente, se les pide que realicen un proyecto de dificultad media. Este proyecto es propuesto por los profesores (aunque excepcionalmente hemos aceptado proyectos propuestos por los estudiantes) y cada equipo realiza un proyecto distinto.

Para realizar el proyecto los equipos tienen un periodo de dos meses en el que no hay clases de TPAAGP de tal forma que pueden dedicar el tiempo designado a ellas a la realización del proyecto.

Como producto final, los estudiantes deben entregar el código completo de la aplicación que hayan desarrollado así como una documentación técnica. También deben hacer una presentación oral y pública de su proyecto incluyendo una demostración en vivo del funcionamiento de su programa. Finalmente se les pide que entreguen un vídeo de presentación del proyecto que incluya una demostración de cómo funciona. Todo este material se piden en inglés, en parte para facilitar la familiarización de los estudiantes con esta lengua, y en parte para poder mostrar a Google los trabajos realizados al final del curso.

A continuación resumimos los proyectos realizados hasta ahora (véase el sitio web del curso para más detalles).

- *Guide and vote (guía y vota)*. Guía al usuario del teléfono a lo largo de una visita turística a distintos puntos de interés de una localización concreta.
- *Catch the thief (pilla al ladrón)*. Es un juego en el que hay varios policías y un ladrón. Todos están distribuidos y se mueven sobre un terreno de juego cuadrado compartido. El objetivo es que los policías rodeen al ladrón.
- *Phone that guides a group (teléfono que guía un grupo)*. El teléfono calcula el mejor punto de encuentro para un grupo de amigos que están cerca y que quieren encontrarse.

- *Tracking a Bluetooth device (rastreado un dispositivo Bluetooth)*. Varios teléfonos colaboran en el rastreo de la trayectoria que describe un dispositivo ajeno que dispone de Bluetooth (por ejemplo un viejo teléfono móvil o un ordenador).
- *Personal Trainer (entrenador personal)*. Implementa un entrenador personal que sugiere, gestiona y mantiene un plan personalizado de ejercicios deportivos en el cual se determinan distancias a correr, calorías a consumir, etc.
- *Distributed Tetris (Tetris distribuido)*. Permite jugar al popular Tetris de manera distribuida entre varios jugadores organizados por turnos.
- *Easy Parking (fácil aparcamiento)*. Permite recordar el lugar de aparcamiento de nuestro coche y mostrar su ubicación en Google Maps. También mediante Google Maps, se puede calcular en cualquier momento el mejor camino de vuelta a él. Permite gestionar también la compartición del vehículo por varios usuarios.
- *Zimbra Synchronization Provider (proveedor de sincronización con Zimbra)*. Permite sincronizar un teléfono Android con el proveedor de correo de Zimbra.

3. Valoración de los estudiantes

Para dar una valoración de los resultados desde el punto de vista de los estudiantes, diseñamos un cuestionario de 15 preguntas que fue contestado por 11 de los 20 estudiantes del taller del cuatrimestre de otoño del presente curso. A cada pregunta los estudiantes podían contestar con un número del 1 (muy en desacuerdo) al 5 (muy de acuerdo). Los resultados obtenidos están desplegados en el Cuadro 1.

Como puede verse en el cuadro, la valoración global que tienen los estudiantes de la asignatura es positiva (Pregunta 15). Todas las respuestas indican que los contenidos de la asignatura son novedosos y de interés actual (Pregunta 1) y que en general el taller les ha aportado conocimientos de utilidad para su futura experiencia profesional (Pregunta 14).

El hecho de tener profesores que actúen como

guías en una búsqueda personal de conocimientos (Preguntas 2 y 3) ha sido valorado muy positivamente así como el material de soporte de la asignatura (Pregunta 7).

Con respecto al método de evaluación, los resultados indican que parece adecuado (Pregunta 9), tanto por la existencia de ejercicios semanales (Pregunta 8) como por la realización y características del proyecto final (Preguntas 11 y 13).

Los contenidos desde el punto de vista de su interés y dificultad (Preguntas 4 y 5) están valorados positivamente, sin embargo, la forma de impartirlos (Pregunta 6) es un punto a mejorar. Es posible que obre en nuestra contra en este punto la poca experiencia que tenemos programando en Android.

Otros puntos a mejorar son los correspondientes a las Preguntas 10 y 12. Los estudiantes no parecen valorar positivamente el que los trabajos y algunas presentaciones se pidan en inglés. Sin embargo, lo consideramos necesario para poder dar a conocer los resultados del curso y el trabajo de los propios estudiantes. Por tanto, debemos reforzar el explicarles el motivo de esta petición para facilitar que lo vean como algo provechoso para ellos.

En general, los resultados indican que un alto interés por el tema de estudio suple limitaciones docentes y/o mecanismos de evaluación sofisticados.

4. Impresiones e intuiciones

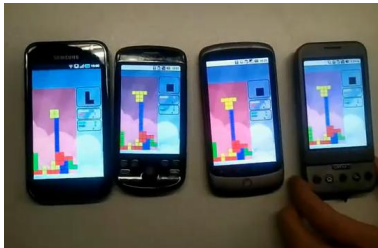
Como profesores hemos de decir que este curso nos ha sorprendido muy gratamente. Acostumbrados a impartir asignaturas teóricas en las que el común denominador es la indiferencia por parte de la mayoría de los estudiantes de Informática de nuestra Universidad, no esperábamos encontrarlos con estudiantes tan motivados y trabajadores en este curso. Cabe mencionar que esta motivación inicial no ha disminuido a lo largo del curso y que de hecho se consolida en el desarrollo del proyecto final, de cuyos resultados nos sentimos muy satisfechos. La sorpresa es aún mayor cuando reflexionamos sobre la carga lectiva del curso (4 créditos ECTS), ya que consideramos que el trabajo global que realizan los estudiantes es mayor los créditos reconocidos.

Adicionalmente, esta experiencia nos ha permitido adoptar nuevas posiciones y retos docentes, obligándonos a replantear nuestro rol habitual. En efecto, tradicionalmente estamos acostumbrados a cur-


Cuestión	Puntuación					Media
	1	2	3	4	5	
1. Los contenidos de esta asignatura son novedosos y de interés actual.	-	-	-	2	9	4.8
2. Es positivo tener profesores que guíen en la búsqueda de conocimientos.	-	1	2	2	6	4.2
3. Valoro positivamente la investigación que requiere esta asignatura.	-	-	2	8	1	3.9
4. Los temas tratados semanalmente me han resultado interesantes.	-	1	4	5	1	3.5
5. Los temas tratados semanalmente tienen un grado de dificultad adecuado.	-	1	4	4	2	3.6
6. Los temas tratados semanalmente se impartieron adecuadamente.	1	3	4	3	-	2.8
7. El material de soporte para la realización de la asignatura es adecuado.	-	-	2	6	3	4.1
8. Valoro de manera positiva la existencia de ejercicios semanales.	-	-	1	4	6	4.5
9. Me parece adecuado el método de evaluación de la asignatura.	-	-	4	6	1	3.7
10. Me parece adecuado que los trabajos se hagan en inglés.	1	1	1	5	3	3.7
11. Los proyectos finales son interesantes y de una dificultad adecuada.	-	-	1	8	2	3.9
12. Es adecuado hacer una presentación oral pública del proyecto.	2	1	3	4	1	3.1
13. Es adecuado realizar un vídeo que presente el proyecto.	-	1	2	4	4	4.0
14. Este curso aporta conocimientos útiles para mi futuro profesional.	-	-	2	2	7	4.5
15. Mi valoración global de la asignatura es positiva.	-	1	1	8	1	3.8

Cuadro 1: Resultados del cuestionario realizado en el cuatrimestre de otoño del curso 2010-2011 para valorar los resultados del taller TPAAGP. Para cada pregunta, se muestra el número de votos para las puntuaciones entre 1 (muy en desacuerdo) y 5 (muy en acuerdo).


(a) Proyecto *Distributed Tetris*



(b) Proyecto *Catch the thief*



(c) Proyecto *Guide and Vote*



(d) Vídeos en YouTube

Proyecto	Fecha de subida	Reproducciones
Guide and vote	23 jul. 2010	94
Catch the thief	23 jul. 2010	87
Phone that guides a group	23 jul. 2010	52
Tracking a Bluetooth	23 jul. 2010	76
Personal Trainer	02 feb. 2011	46
Distributed Tetris	02 feb. 2011	45
Easy Parking	02 feb. 2011	80
Zimbra Provider	08 feb. 2011	34

Cuadro 2: Imágenes capturadas de algunos de los vídeos de presentación de los proyectos (Subfig. (a), (b) y (c)) y número de reproducciones de todos los vídeos en YouTube a fecha de 8 de Mayo de 2011 (Subfig. (d)).

En los que el profesor se considera un *experto* en el tema y por tanto es la principal fuente de conocimiento para los estudiantes. Es el caso, por ejemplo, en la docencia de disciplinas bien establecidas como la programación o las estructuras de datos. Sin embargo, en el ámbito universitario esta situación es muy difícil de mantener (si no imposible) cuando se trata de un curso de una tecnología punta, como es el caso de Android. En nuestro caso hemos cambiado el rol de profesor *experto* a profesor *guía* o entrenador en el aprendizaje.

Desde el punto de vista de la Universidad, este curso ha permitido que se dieran de manera natural situaciones que son muy difíciles de encontrar en otros cursos universitarios. Nos referiremos específicamente al tipo de curso y a las relaciones Universidad-Empresa. A pesar de que las nuevas tendencias parecen indicar que es deseable una relación más estrecha entre la Universidad y el mundo empresarial, es muy difícil encontrar cursos que lo permitan. En TPAAGP establecer esta relación fue algo que surgió de manera espontánea, no sólo por la relación directa con Google, sino también porque al final del curso siempre intentamos ponernos en contacto con alguna empresa del sector.

Por otra parte, el taller TPAAGP puede considerarse un curso integrador: no sólo se trabaja Android como tema nuevo y de tecnología punta, sino que se deben aplicar muchos conocimientos consolidados provenientes de asignaturas de diversas índoles: programación, algoritmos y estructuras de datos, sistemas distribuidos, bases de datos, ingeniería de software, entre otros. Esto convierte a TPAAGP en un curso difícil pero al mismo tiempo idóneo para enfrentar a los estudiantes al tipo de tareas que muy probablemente requerirán en su vida profesional.

A fin de dar una mayor visibilidad al trabajo de los estudiantes y facilitarles posibles contactos, tanto en el mundo empresarial como a nivel de colaboraciones con otros desarrolladores de Android, se ha creado el canal `tpagpUPC` en YouTube accesible via <http://www.youtube.com/user/tpagpUPC>. El Cuadro 2 muestra algunas imágenes de los vídeos incluídos en dicho canal, así como información referente al número de reproducciones a fecha de 8 de Mayo de 2011. Junto con el sitio web de la asignatura, dicho canal pretende también dar visibilidad a la asignatura y captar nuevos estudiantes.

Los vídeos representan una nueva forma de difu-

sión del trabajo en temas tecnológicos. Para desarrollarla plenamente sería deseable que los futuros ingenieros tuvieran algún tipo de formación en aspectos relacionados con la comunicación multimedia.

Agradecimientos

Nuestro agradecimiento al *Programa AndroidEDU* del *Google's EMEA Engineering University Programs* por el hardware aportado para este proyecto.

Referencias

- [1] ADT Plugin for Eclipse (developer.android.com/sdk/eclipse-adt.html).
- [2] AndroidEDU EMEA (sites.google.com/site/androideduemea).
- [3] E. Burnette. *Hello, Android. Introducing Google's Mobile Development Platform*. The Pragmatic Bookshelf, 2009.
- [4] HTC (www.htc.com/www/product/g1).
- [5] Android Developers: The Developer's Guide (developer.android.com/guide).
- [6] Eclipse (www.eclipse.org).
- [7] C. Collins F. Ableson and R. Sen. *Unlocking Android: A Developer's Guide*. Manning Publications, 2009.
- [8] C. Haseman. *Android Essentials*. Apress Ed, 2008.
- [9] Android Developers: Hello, World. (developer.android.com/guide/tutorials/hello-world.html).
- [10] Java Platform (www.oracle.com/technetwork/java/javase).
- [11] J. Ledford. *Web Geek's Guide to the Android-Enabled Phone*. QUE Ed., 2009.
- [12] M. Miller. *Google-pedia: The ultimate Google resource*. QUE Ed., 2008.
- [13] M. Murphy. *Beginning Android*. Apress Ed, 2009.
- [14] Google Phone Gallery (www.google.com/phone/detail/nexus-one).
- [15] Z. Mednieks R. Rogers, J. Lombardo and B. Meike. *Android Application Development*. O'Reilly Media Ed., 2009.
- [16] Racó de la FIB (raco.fib.upc.edu).
- [17] SQLite (sqlite.org).
- [18] The WebKit Open Source Project (webkit.org).

Dos casos prácticos del uso de la Wiki en Ingeniería Informática: consideraciones para su uso en el Grado de Informática

Marta E. Zorrilla Pantaleón
Matemáticas, Estadística y Computación
Universidad de Cantabria
Avda. de los Castros s/n
39005 Santander
marta.zorrilla@unican.es

Inés González Rodríguez
Matemáticas, Estadística y Computación
Universidad de Cantabria
Avda. de los Castros s/n
39005 Santander
gonzalezri@unican.es

Resumen

En este trabajo se describen dos actividades colaborativas diseñadas en la Universidad de Cantabria (UC) utilizando la Wiki como herramienta instrumental. Se realiza un análisis detallado de cada experiencia y se presentan una serie de conclusiones que permitirán a los profesores involucrados en el nuevo Grado de Informática valorar en qué contexto utilizar esta herramienta, cómo organizar la actividad y cómo evaluarla. Así mismo, se destacan situaciones que han ocurrido durante el desarrollo de las actividades con objeto de que los profesores las puedan tener en cuenta y evitarlas o, al menos, minimizar su impacto en el diseño de su actividad Wiki.

Summary

In this paper we describe two collaborative learning activities designed at University of Cantabria (UC) based on Wikis. We provide a detailed analysis of each experience together with some conclusions. These should allow those instructors involved in the new Computer Science degree to evaluate in what context to use this tool and how to organise and assess the activity. We also highlight different issues which have arisen during the development of each activity, so that other instructors can take them into account and either avoid them or, at least, minimise their impact in their wiki activity.

Palabras clave

Wiki, Aprendizaje constructivista, Trabajo colaborativo, Web 2.0

1. Introducción

La incorporación de las TIC en la enseñanza ha supuesto una revolución en el proceso enseñanza-aprendizaje. Aunque aún no hay estudios concluyentes que permitan afirmar que la utilización de los medios informáticos en la educación ha servido para mejorar los resultados académicos, en [9] concluyen que la tecnología tiene un moderado y positivo impacto en el progreso cognitivo del estudiante.

Esta nueva forma de enseñar a aprender se sustenta en las teorías constructivistas del aprendizaje ([2][14]) que preconizan que los alumnos sean elementos activos y dinámicos en la construcción de su propio conocimiento y el profesor actúe como mediador para su correcta consecución. Esta misma idea es la que subyace en las nuevas directrices dictadas desde el EEES para la elaboración de los planes de estudio de las titulaciones de Grado, lo que nos lleva a los profesores universitarios a diseñar nuevas actividades que favorezcan la adquisición de conocimiento al tiempo que se trabajan las competencias transversales y específicas de la titulación.

Existen diferentes métodos y técnicas pedagógicas para diseñar estas actividades, tales como el método del caso, el aprendizaje basado en problemas, en proyectos o en cuestiones [8], pero hoy en día, parece adecuado desarrollarlas apoyándonos en las herramientas de la Web 2.0 si queremos garantizar un mayor éxito en su consecución. De acuerdo a [12] el 66% de los jóvenes se conectan diariamente a una red social.

La Web 2.0 ofrece diferentes herramientas que permiten desarrollar actividades educativas, como los blogs, la Wiki, los marcadores sociales, la elaboración de documentos de forma colaborativa (caso de GoogleDocs), entre otras.

Una fuente interesante de experiencias y recursos se encuentra en <http://www.unir.net/wikiweb20/>.

En el caso que nos ocupa, la herramienta utilizada es la Wiki. La decisión de elegir la Wiki se debió principalmente a su facilidad de uso, la posibilidad de organizar crítica y creativamente la información generando contenidos hipermedia no lineales (inconveniente que presentan los blogs) de forma colaborativa, y por su uso generalizado en el sector informático para las tareas de documentación y generación de ayudas en línea, por lo que resultará ser una herramienta imprescindible para nuestros futuros egresados de informática. La decisión no estuvo carente de dudas conocida la falta de herramientas automáticas para su evaluación [4] y la necesidad de que los trabajos realizados durante el curso no se perdieran (copias de seguridad incrementales).

En la literatura se encuentran experiencias educativas en el uso de la Wiki positivas [6][13] y negativas [3], estas últimas marcadas principalmente por la presión sobre los estudiantes para entregar trabajos o examinarse de otras asignaturas y la pereza de enfrentarse a un entorno de edición nuevo. La Wiki se ha utilizado con relativo éxito para la documentación de lenguajes de programación [10], el desarrollo de apuntes y ejercicios [5] o la generación de repositorio de contenidos [11] en master, entre otras tareas.

El presente trabajo se organiza de la siguiente manera. En las secciones 2 y 3 se describen y analizan las dos actividades desarrolladas en la UC. Y en la sección 4 se resumen las conclusiones que se han alcanzado y se ofrecen sugerencias para su utilización en el Grado de Informática.

2. Generación de contenidos en Bases de datos Avanzadas (BDA)

Esta actividad se desarrolló en el contexto de una asignatura optativa de 4º de I. Informática durante el curso 2009/10. El número de alumnos matriculados fue 15 y la actividad era obligatoria.

Los objetivos que se pretendía cubrir fueron:

- Que los alumnos profundizaran en una parte del temario y elaboraran material propio (análisis personal, opiniones, propuestas, etc.)
- Que los alumnos trabajasen de forma colaborativa desarrollando y discutiendo sobre un tema y realizando crítica constructiva

(sugerir modificaciones, aclaraciones, ejemplos, etc.) en temas escritos por otros.

- Mejorar la capacidad de síntesis y análisis del estudiante a partir de las consultas bibliográficas y de la lectura y la revisión del trabajo de sus compañeros.
- Realizar tareas de búsqueda bibliográfica y selección de las fuentes más adecuadas.
- Aprender a escribir de forma clara y concisa los detalles de un tema pensado para ser leído en la Wiki (uso de estilos, colores, etc.)

2.1. Descripción de la actividad

La actividad consistió en desarrollar, en grupos de 3 alumnos, un tema relativo a una tecnología avanzada del campo de bases de datos de entre los propuestos por la profesora o bien sugerido por ellos. Se les comentó que el trabajo debía incluir conceptos teóricos, ejemplos aclaratorios, casos prácticos y las referencias bibliográficas utilizadas. Además se les indicó que i) siempre que se transcribiera un texto se indicase la fuente; ii) que se incluyeran en la página Diccionario (creada en el menú principal de la Wiki por la profesora) las definiciones de los términos que aparecieran en el tema; y iii) que añadieran en la página Referencias la bibliografía utilizada según el estilo APA con objeto de que se habituasen a utilizar un formato. Asimismo se les indicó que utilizaran el entorno de Discusión para contribuir con comentarios, ejemplos u otras consideraciones que pudieran ayudar a mejorar la presentación y comprensión del tema propio o del resto.

2.2. Planificación temporal

Para su desarrollo se estableció un marco temporal que cubría todo el segundo cuatrimestre:

- Hito 1. Semana 1 (26/02/2010) – En sesión presencial, la profesora mostró la herramienta Wiki y explicó en detalle la actividad (tareas, entregables y criterios de evaluación). Cada grupo se identificó y eligió un tema, creando su página principal en la Wiki y enlazando ésta al menú propuesto por la profesora. La página principal de la Wiki <http://basesdedatosavanzadas.wikispaces.com> recogía los criterios de diseño a seguir.
- Hito 2. Semana 4 (19/03/2010) – Los grupos debían tener un esquema del tema a

desarrollar, seleccionada la bibliografía y definidos los términos más relevantes.

- Hito 3. Semana 8 (20/04/2010) – Los alumnos debían tener el tema desarrollado para que los compañeros los revisaran y escribieran sugerencias o modificaciones.
- Hito 4. Semana 12 (10/05/2010) – Cada grupo modificaba su tema atendiendo a las sugerencias recibidas.
- Hito 5. Semana 15 (31/05/2010) – Cada alumno evaluó el trabajo realizado por el resto de grupos en sesión presencial. Se realizó la encuesta de opinión.

2.3. Evaluación

Un punto importante en el diseño de este tipo de actividad es el modo de evaluación. En este caso, supuso el 20% de la calificación final de la asignatura, con objeto de que tuviera carácter motivador para el alumno y viera bien recompensado su esfuerzo. Asimismo, los alumnos participaban en la evaluación de los trabajos de sus pares de forma que se sintieran más implicados y se apoyaran mutuamente. Los contenidos desarrollados entraban en la evaluación ordinaria de la asignatura.

La tabla 1 recoge los criterios de evaluación tomados de [13]. Se evaluó sobre 100 puntos donde 70 se calcularon como media entre la nota del profesor y la media conseguida de la opinión de los alumnos y 30 repartidos en: 15% en función del número de ediciones realizadas por el alumno en su trabajo y el 15% por el número de discusiones abiertas o respuestas dadas.

Redacción (10%)	Nota media alumnos (a) Nota profesor (b)
Estructuración del trabajo y presentación visual (10%)	
Contenido (25%)	
Referencias (10%)	
Aportación propia del grupo (15%)	Nota prof. (c)
Funcionamiento del grupo (15%)	
Aportación individual (15%)	
Nota final	$0.7\left(\frac{a+b}{2}\right) + 0.3c$

Tabla 1. Criterios de evaluación en BD avanzadas

2.4. Herramienta seleccionada

La herramienta seleccionada fue Wikispaces (<http://www.wikispaces.com>). Las razones fueron:

- Su interfaz es muy intuitiva y absolutamente accesible para usuarios sin experiencia previa.
- El uso de la modalidad gratuita es ilimitado y contempla las prestaciones necesarias para facilitar la creación de una Wiki de calidad.
- La interfaz está disponible en español.
- El trabajo realizado tiene visibilidad global, con lo que se pretende que los alumnos se sientan motivados porque su trabajo pueda serle útil a otros usuarios de Internet.
- Dispone de un entorno de estadísticas que, aunque limitado, permite visualizar la actividad realizada en la Wiki en cada página (número de visitas, ediciones, etc.) y por cada alumno (número de ediciones y mensajes de comentario enviados).

Wikispaces también presenta limitaciones como son la imposibilidad de crear grupos de usuarios y de calificar en la plataforma (a diferencia de Moodle) y no permite la programación de copias de seguridad, deben realizarse manualmente.

2.5. Análisis de la experiencia

Como antes se ha mencionado se establecieron 5 hitos en la planificación de la actividad. En el primero, se constató que a pesar de que los alumnos nunca habían trabajado en una Wiki, están familiarizados con las tecnologías y rápidamente se hicieron con el entorno.

Al llegar el segundo hito, se observó que solo dos grupos habían estructurado la página de contenidos. La profesora notificó este hecho y los alumnos le indicaron que era por falta de tiempo ya que había una asignatura troncal que les consumía mucho tiempo. En ese mes solo hubo 5 editores de 15 alumnos posibles, lo que puso de manifiesto que no se trabajaba de forma colaborativa. Se comentó con los alumnos este hecho y respondieron que en general trabajan con otras herramientas ofimáticas y lo ponen en común de forma presencial, quedando uno de los alumnos responsable de editar la Wiki.

Llegado el tercer hito, al menos, 12 alumnos habían realizado alguna edición y se observó que esta actividad se realizó muy cerca de la fecha de

entrega. También se detectó que no utilizaban la herramienta de discusión lo que indicaba que utilizan otras vías de comunicación.

Al llegar el cuarto hito, los alumnos no habían escrito ningún comentario. Las reacciones se produjeron cuando se les indicó de nuevo que la actividad no se estaba realizando adecuadamente. A partir de ahí aparecen mensajes aunque no muchos, 20 comentarios y 14 respuestas, dado que había 5 temas y 15 alumnos.

Respecto a la calidad de los contenidos o mensajes escritos, decir que en la mayoría de los temas los alumnos se han apoyado en otras Wikis o documentos de la Red, copiando directamente la información y sin confirmar si su contenido y fuentes eran fidedignos. En solo un tema se refirieron a libros y en ningún caso a un artículo científico. La extensión de la Wiki osciló entre las 1800 y 4500 palabras. Para esta actividad en concreto una Wiki con menos de 3000 palabras se consideraba como poco elaborada. Respecto a los comentarios, en general, se referían a aclaraciones sobre el texto escrito (explicar un párrafo, aclarar un ejemplo, explicar la notación utilizada,...), que fueron contestadas directamente en el entorno de discusión pero salvo en un tema, no se mejoró la explicación en la Wiki. Apareció de nuevo el problema del tiempo, aunque algunos grupos a última hora reaccionaron cuando la profesora hizo comentarios sobre cada tema.

2.6. Análisis de la encuesta

Para valorar la actividad desde el punto de vista del alumno, se diseñó la encuesta de la tabla 2. Las preguntas P1 a P11 se contestaban en una escala de 1 a 5, siendo (1) Pésimo, (2) Malo, (3) Aceptable (3), (4) Bueno y (5) Excelente.

P1. Considera interesante la experiencia realizada.
P2. Considera que el uso de este tipo de plataformas ayuda a la realización del trabajo en grupo.
P3. Considera que este tipo de plataformas facilita el trabajo cooperativo y la coordinación de esfuerzos.
P4. Considera de utilidad el uso de la plataforma para el desarrollo de contenidos.
P5. Considera que este tipo de plataformas constituyen una herramienta útil en la compartición y acceso al conocimiento.
P6. Considera fácil el uso de la plataforma y la incorporación de contenidos.
P7. Considera que el hecho que ciertos temas de la asignatura estén recogidos en la plataforma ayuda a su

estudio y a aumentar su interés en ellos (en comparación con un pdf, libro o una presentación por parte del profesor).
P8. Considera que el uso de la plataforma ha repercutido positivamente en el aprendizaje de los elementos tratados por la asignatura.
P9. El proceso de generación y evaluación de los temas desarrollados en la Wiki ha resultado más sencillo y dinámico que utilizando una memoria (pdf) o una presentación
P10. El esfuerzo (horas de trabajo) dedicado a la actividad se ha visto incrementado por el uso de la herramienta
P11. Consideraría positivo incorporar este tipo de plataformas en más asignaturas con objeto de hacerlas más atractivas
P12. Para qué actividades considera que las plataformas Wiki son útiles en general.
P13. Existen otras herramientas/servicios web que considere más interesantes para conseguir los mismos objetivos (P.ej. Googledocs, Zoho, etc.). Cuales:
P14. Indique los aspectos negativos que haya encontrado :
P15. Nº de horas totales requeridas para realizar la actividad: 1. estudio y síntesis de contenidos de su tema: ____ 2. estudio y comentarios de otros temas: ____ 3. cuántas de estas horas son conectado en la Wiki:

Tabla 2. Encuesta a alumnos.

Se recogieron 9 encuestas de las 15 posibles. En la tabla 3 se recoge el valor medio y la dispersión en cada una de las preguntas.

Pregunta	Media	Desviación
P1	3,3	0,8
P2	3,7	0,6
P3	3,7	0,8
P4	3,4	0,6
P5	4,6	0,6
P6	4,3	0,9
P7	3,3	0,8
P8	2,8	0,6
P9	3,6	1,1
P10	2,9	0,5
P11	3,1	1,0

Tabla 3. Resultado de la encuesta BDA.

Las preguntas P1 a P4 reflejan que la actividad les resultó aceptable y que sí consideran útiles las Wiki para el desarrollo de contenidos de forma colaborativa. Las preguntas P5 y P6 refuerzan esta idea pero debe ser cuando la Wiki la construyen otros ya que P8 refleja que los

contenidos desarrollados por ellos no parecen haberles ayudado a estudiar el tema con mayor facilidad o claridad. Esto es que los contenidos no estaban suficientemente detallados y claros. P9 indica que la actividad Wiki les supuso menos tiempo que en una presentación o memoria, lo que se puede asociar con el formato habitual de la Wiki en la que se muestra información breve y concisa. Atendiendo a P10 parece que la herramienta no les supuso una carga de horas adicional al trabajo y aunque con bastante diferencia de opinión, no les importaría usar la Wiki en otras asignaturas. Según los alumnos, las Wikis les resultan útiles para:

- Crear documentación colaborativa pero no en grupo, idea de Wikipedia.
- Para consultar información y resolver dudas de forma rápida así como para adquirir conocimientos sobre temas variados.
- Para compartir información y crear comunidades sobre un tema.
- Para actividades enciclopédicas y redacción de manuales.
- Para buscar información técnica o informarse sobre nuevas tecnologías.

Y mencionan como aspectos negativos:

- El no haber podido dedicar el tiempo necesario para hacer un trabajo completo debido a la sobrecarga de trabajos de otras asignaturas.
- No saber qué interesa en la Wiki de cara a la evaluación (demuestra falta de atención en hito 1).
- La falta de una herramienta de comunicación interna al grupo que no sea pública.
- La posibilidad de perder trabajo si dos personas editaban a la vez.

Esto último enlaza con el uso de otras herramientas que en su opinión les parecían interesantes utilizar como GoogleDocs, Dropbox y GoogleWave. Esto parece indicar que una actividad de desarrollo de contenidos de investigación, según los alumnos, es mejor realizarla en formato memoria con su facilidad de seguimiento de versiones que en la Wiki.

En cuanto al tiempo dedicado a realizar la actividad (Tabla 4), en media es inferior al

estimado para la misma, 30 horas, observándose un fuerte desequilibrio entre los alumnos, donde el alumno 4 (A4) destaca sobremedida.

	A1	A2	A3	A4	A5	A6	A7	A8	A9	Media	Dispersión
1	10	4	20	41	10	12	10	20	5	14,7	10,1
2	2	2	10	24	4	5	3	2	3	6,1	6,4
3	3	3	15	30	6	5	5	2	3	8,0	8,2

Tabla 4. Dedicación a la actividad en horas totales.

3. Tema introductorio en Introducción a la Lógica (IL)

La segunda actividad, de carácter más modesto, se desarrolló en una asignatura obligatoria de primer curso de I. Informática en el curso 2009/10. Con 53 alumnos matriculados y una asistencia regular del 75%, la actividad se propuso como optativa y contó con la participación de 37 alumnos.

Dado el marcado carácter teórico de la asignatura, el objetivo principal de la actividad era motivar al alumno en el estudio de la asignatura, desarrollando el primer tema de introducción histórica en la línea de lo sugerido en [6]. Así mismo se incluían los objetivos mencionados en la sección anterior, aunque con un nivel de consecución menor (alumnos de primero y un desarrollo de contenidos menos ambicioso).

3.1. Descripción de la actividad

Basándose en la experiencia narrada en [6], se propuso a los alumnos elaborar una Wiki de Personajes de la Lógica. La profesora elaboró una página inicial, proporcionando un panorama general de la historia de la Lógica con menciones a sus personajes más influyentes. A partir de ahí, los alumnos, organizados en grupos de 4 miembros, debían elegir uno de los personajes y crear una breve página (enlazada a la inicial) con la información básica referente tanto a su biografía como a su contribución a la lógica. La actividad finalizaba con la realización de un test de elección múltiple, basado en la información plasmada por los propios estudiantes en la Wiki.

3.2. Planificación temporal

Para el desarrollo de nuestra actividad se estableció el siguiente marco temporal:

- Hito 1. Semana 1 (18/02/2010) – En una sesión presencial (2h), se explicó la actividad a realizar y se presentó la herramienta Wiki de Moodle. Cada grupo eligió su “personaje” y enlazó su página con una primera prueba de edición. En esta misma sesión se estableció un esquema y formato comunes para todos los grupos (decididos previamente por la profesora) y se permitió que los alumnos comenzaran el trabajo (reparto de tareas, documentación, etc.).
- Hito 2. Semana 5 (15/03/2010) – Los alumnos debían tener una primera versión de su página y leer las elaboradas por sus compañeros. Las ideas aportadas por cada grupo podrían ser utilizadas por el resto para mejorar su contribución. Dedicación estimada: 5 horas.
- Hito 3. Semana 15 (27/05/2010) – Cada grupo debía publicar la versión final de su página. Dedicación estimada: 1 hora.
- Hito 4. Semana 16 (03/06/2010) – En otra sesión presencial (2h), cada alumno realizó un test con preguntas relativas a la Wiki. A raíz de la necesidad de extraer información de los contenidos elaborados por sus compañeros, se estableció una reflexión informal sobre la claridad y calidad de los contenidos. Finalmente, se realizó la encuesta de opinión (anónima) sobre la actividad.

3.3. Evaluación

Los alumnos podían obtener dos calificaciones: una por la página elaborada en grupo y otra, con el test de elección múltiple realizado en la última sesión. Ambas notas se combinaban con otras de actividades voluntarias para poder mejorar la nota final de la asignatura hasta en 3 puntos.

Dado el nivel de madurez de los alumnos, para calificar la página elaborada por cada grupo no se recurrió a la evaluación sumativa por pares (como se hizo en BDA), sino que se limitó a la profesora. Se optó por utilizar escalas del 0 al 10 (de menor a mayor satisfacción) correspondientes a los siguientes criterios de alto nivel:

1. Formato: coincidencia con el formato acordado al comienzo de la actividad, incluida la presencia de referencias.
2. Redacción: buena organización de los contenidos y claridad en la redacción.
3. Contenido: selección y síntesis adecuados.

Se eligieron estos criterios porque, según [1], sirven para evaluar cualidades de alto nivel del aprendizaje (organización, fluidez de estilo, claridad de ideas, etc.) y suelen ser fiables, a la vez que permiten una retroinformación rápida a los alumnos. La nota final del grupo fue una media aritmética de las escalas obtenidas para cada uno de los tres criterios. En cuanto al test de elección múltiple, se puntuó de forma estándar.

Adicionalmente, el segundo hito debía servir para realizar una evaluación formativa (que no sumativa) por pares de las páginas de la Wiki. Otro instrumento de evaluación formativa por pares fue el test de elección múltiple final, con la necesidad de obtener información de la Wiki potenciando el espíritu crítico de los estudiantes. El cuestionario se realizó a través de la plataforma Moodle, permitiendo una corrección rápida y objetiva, además de proporcionar resultados inmediatos a los alumnos. Este tipo de evaluación, puede resultar más atractiva al alumnado actual y existen crecientes evidencias sobre sus ventajas desde el punto de vista pedagógico [7].

3.4. Herramienta seleccionada

Para desarrollar la actividad se optó por la Wiki proporcionada por Moodle. El principal motivo fue que este sistema ya se utilizaba en la asignatura para desarrollar diversas actividades de enseñanza-aprendizaje. Moodle ofrece un entorno de edición muy simple; esto, que puede suponer una limitación en un proyecto de más envergadura, facilita el uso de la herramienta por parte de los alumnos de primer curso. Como sólo los alumnos del curso pueden acceder a la Wiki, se evitan problemas de copyright, por ejemplo con imágenes incluidas por los alumnos en sus páginas. Finalmente, Moodle permite acceder al registro de cambios, proporcionando al profesor un elemento de evaluación adicional. Desafortunadamente, la Wiki de Moodle no dispone de un entorno de estadísticas, como sí ocurre con Wikispaces.

3.5. Análisis de la experiencia

De los cuatro hitos en que se planificó la actividad, el primero resultó muy positivo, pues eliminó posibles reticencias de los alumnos respecto al entorno Wiki y la actividad de edición.

Al llegar al segundo hito, todos los grupos respetaron la fecha de entrega. Sin embargo, se

observó gran disparidad en el número de ediciones realizadas así como en la variedad de miembros del grupo que participaron en las mismas. En un extremo, hubo un grupo con una única edición por parte de un único miembro y en el otro, un grupo realizó 59 ediciones por parte de todos los integrantes del grupo. Así, el uso de la Wiki como herramienta de trabajo colaborativo no fue general, aunque sí se dio en casos concretos.

El segundo hito también incluía una evaluación formativa por pares, con los alumnos leyendo y reflexionando sobre todas las contribuciones de la Wiki. Sin embargo, los estudiantes mostraron una ausencia total de espíritu crítico, mostrándose plenamente satisfechos tanto con su trabajo como con el del resto de grupos. De hecho, no hubo ninguna edición posterior a esta fecha, por lo que el tercer hito de la planificación perdió su razón de ser.

Por el contrario, el último hito sí consiguió su objetivo de evaluación formativa, dado que para realizar el test, los estudiantes necesitaron leer la Wiki de manera comprensiva y extraer información. Esto sirvió para que reflexionaran sobre la calidad de sus contribuciones: las bondades de una estructura y formato comunes pero también la existencia de frases ambiguas o mal construidas, errores ortográficos, poca claridad al estructurar algunos contenidos, etc.

También en relación a los contenidos, al igual que en BDA, la principal fuente de información de los alumnos fueron otros documentos de la Red (principalmente Wikipedia), con copias a menudo literales o, peor aún, parciales, sin mediar un análisis o comprensión real de lo que se escribía. En ningún caso hubo referencias a libros, aunque, debido a la carga de trabajo y el nivel en que se enmarcaba la actividad, esto resultaba esperable.

3.6. Análisis de la encuesta

La opinión de los alumnos sobre la actividad se recogió con la encuesta de la Tabla 2. Hubo un total de 37 respuestas, de las que la Tabla 5 recoge el valor medio y la dispersión.

Las conclusiones extraídas para BDA serían, en general, aplicables aquí. Cabe hacer una excepción con la P8 debido a que aquí el tema tratado en la Wiki es introductorio y no influye en el estudio del resto de contenidos de la asignatura.

Procede destacar las respuestas a la pregunta P11, puesto que supone que los alumnos

consideran la Wiki como una herramienta que ayuda a aumentar el interés por la asignatura.

Pregunta	Media	Desviación
P1	3,97	0,68
P2	4,14	0,81
P3	3,95	0,73
P4	4,00	0,70
P5	4,11	0,91
P6	4,14	0,91
P7	3,86	1,09
P8	3,27	0,83
P9	3,97	0,82
P10	2,81	1,01
P11	4,03	0,68

Tabla 5. Resultado de la encuesta II.

Respecto a para qué resultan útiles las Wikis, mencionar que 11 alumnos no dan ninguna respuesta y el resto sugiere usos de lo más variado y pintoresco, aunque quizá la respuesta más común es “trabajar en grupo”, en contraposición con la opinión de los alumnos de BDA que indicaban que colaborativo sí pero no en grupo. Esto podría deberse a una diferencia generacional en el uso de las herramientas Web2.0 en la etapa de secundaria y/o a otros hábitos en el desarrollo de trabajos en grupos en formato memoria

Por último, los alumnos no proporcionaron respuesta a las preguntas P14 y P15 probablemente por su juventud y falta de experiencia en el uso de este tipo de tecnologías.

El tiempo total dedicado a la elaboración de contenidos y a la lectura y comentario de temas ajenos, resultó ser en media 1.86 y 1 hora respectivamente, de las cuales 2 horas han sido conectadas a la Wiki.

4. Conclusiones

De ambas experiencias, podemos extraer las siguientes conclusiones:

1. El uso de la Wiki no supone ningún problema a los alumnos independientemente del curso.
2. Su utilización puede ayudar a acercar a los alumnos asignaturas que resultan menos atractivas a priori especialmente en los primeros cursos.
3. La actividad que se proponga en la Wiki debe estar orientada al desarrollo de contenidos breves y concisos, pero no para el desarrollo

de un tema. La facilidad de edición en línea parece contribuir a que los alumnos copien y peguen lo que se encuentran en la red sin meditar la organización de contenidos y sin realizar una lectura detenida a posteriori de lo escrito, ni una investigación bibliográfica exhaustiva y adecuada a su nivel

4. Es necesario motivar al alumno en el uso de la Wiki y del entorno de discusión, de lo contrario, se realizará la actividad fuera de la herramienta. Por ello es importante que en la evaluación se tenga en cuenta la actividad en la Wiki (nº de accesos, nº de ediciones, discusiones iniciadas, etc.) y la contribución de cada alumno que puede analizarse a través del histórico de cada página.
5. La evaluación por pares aunque interesante, no parece efectiva pues los alumnos no son críticos con sus compañeros.
6. Dado el esfuerzo que supone seguir todos los cambios realizados por los alumnos y evaluar los contenidos y las discusiones abiertas, la Wiki debe plantearse con grupos reducidos.
7. Otro aspecto importante es equilibrar y coordinar bien el número de trabajos que se piden por cuatrimestre a los alumnos, la falta de tiempo fue una de las razones de la baja calidad en los trabajos.

Como sugerencia para el grado, se propone realizar una actividad sencilla, breve y optativa en los primeros cursos en la línea de la propuesta en IL y otra posteriormente obligatoria en la que el trabajo de cada alumno sea necesario para otros estudiantes, evitando que se diluya o desaparezca su trabajo enmascarado por “el grupo”. Y mejor aún si el trabajo es transversal a más de una asignatura, por ejemplo la especificación y documentación sobre la implementación de un sistema de información (ing. de software, programación y bases de datos).

Referencias

- [1] Brown, G. *Assessment: A guide for lecturers*. LTSN Generic Centre, 2001
- [2] Bruner, J.S. *Toward a Theory of Instruction*. Cambridge Mass: Harvard University Press, 1966
- [3] Cole, M. *Using Wiki technology to support student engagement: Lessons from the trenches*. *Computers & Education*, vol. 52(1), pp. 141-146, 2009.
- [4] Córdoba Torrecilla, J., Cuesta Morales, P. *Adaptando un sistema de Wikis para su uso educativo*. XV JENUI. Barcelona, 2009
- [5] González Pareja, A., Calderón Montero, S., Galache Laza, T. y Torrico González, A. *Uso de wikis para la realización de trabajos colaborativos en el aula*. XIV Jornadas de ASEPUMA, Badajoz, España, 2006
- [6] Imberman, S. P. *Making nifty assignments niftier and not so nifty assignments nifty with online technologies*. AI Education Colloquium Workshop, vol. WS-08-02, pp. 46-59. AAAI Press, 2008.
- [7] Irons, A., Alexander, S. eds. *Effective learning & teaching in computing*. Effective Learning and Teaching in Higher Education Series. Routledge Falmer, 2004.
- [8] Jonassen, D. Y Rorher-Murphy, L. (1999): *Activity Theory as a framework for designing constructivist learning environments*. *Educational Technology: Research and Development*, 46 (1)
- [9] Michko, G.M. *Meta-analysis of effectiveness of technology use in undergraduate engineering education*. *Frontiers in Education Conference*, pp. S1A-1 - S1A-6, 2008
- [10] Palomo Duarte, M; Rodríguez Posada, E.; Medina Buló, I., Sales Montes, N. *Tecnologías Wiki en la docencia de Ingeniería Informática*. IX JENUI, 2010
- [11] Redondo López, JM, Ortín Soler, F., Zapico Rodríguez, D., Pérez Pérez, J.R. *Elaboración de Repositorios de Contenidos a Través de Wikis como Técnica de Evaluación*. XVIII JENUI, 2009
- [12] Sánchez, A., Fernández, M. P. *Informe generación 2.0 2010: Hábitos de los adolescentes en el uso de las Redes Sociales*. Universidad Camilo José Cela. <http://www.slideshare.net/ucjc/generacin-20-hbitos-de-los-ad>, 2010
- [13] Villanueva García, A. *Uso de wikis en Ingeniería Informática*. *Revista de Docencia Universitaria*, Monográfico V, 2009
- [14] Vygostsky, L.S. *Mind in Society*, Harvard University Press, Cambridge, MA., 1973

Investigación práctica en educación: *investigación-acción*

Mercedes Maqués Andrés
Departamento de Ingeniería y Ciencia de los Computadores
Universitat Jaume I
Campus Riu Sec
12071 Castellón
mmarques@icc.uji.es

Reina Ferrández Berruero
Departamento de Educación
Universitat Jaume I
Campus Riu Sec
12071 Castellón
ferrande@edu.uji.es

Resumen

En este artículo se expone la metodología de la investigación-acción, una metodología utilizada en investigación educativa en la que el profesor investiga sobre su propia práctica docente. En el trabajo también se realiza una justificación de la necesidad de aplicar una metodología como esta cuando nos proponemos llevar a cabo proyectos de mejora o de innovación en la docencia universitaria. Aunque esta metodología es bien conocida en el ámbito de las Ciencias Sociales, resulta desconocida para una parte importante del profesorado que trabajamos en otras áreas como las Ingenierías, por lo que consideramos muy interesante dar a conocer esta metodología en el ámbito de JENUI.

Summary

This paper describes the methodology of action research, an educational research methodology in which professors investigate on their own teaching practices. In this work we also justify the need of using such methodology when professors intend to innovate or improve their teaching. Although this methodology is well known in the field of Social Sciences, it is unknown for many professors working in other areas such as Engineering. This seems to be a good reason for us to present this methodology in the scope of JENUI.

Palabras clave

Investigación educativa, investigación-acción, innovación docente, investigación práctica.

1. Motivación

En los últimos siete años hemos revisado cerca de 40 trabajos de investigación educativa elaborados por

profesorado universitario de ingenierías en informática que han sido remitidos a JENUI o, más recientemente, a EDUCON¹. Aproximadamente, una cuarta parte de esos trabajos tenían un mismo error: una falta de coherencia en algún punto de su desarrollo. Esa falta de coherencia ha implicado su no aceptación en el congreso. Que esto suceda con tal frecuencia es, sin duda, un motivo para reflexionar, no sólo por el rechazo sufrido sino por cómo ha afectado dicha falta de coherencia a la investigación realizada.

El profesorado que firma estos trabajos se ha tomado la molestia de intentar aplicar alguna innovación en su metodología, la ha puesto en práctica, ha medido su impacto y, más tarde, lo ha recogido todo en un informe que envía a un congreso de educación universitaria. ¿Cómo es posible que un trabajo realizado con tanto empeño resulte rechazado por una falta de coherencia? ¿ha sido tan solo un problema al escribir el artículo o el problema se encuentra también en la investigación llevada a cabo?

Generalmente, cuando los profesores universitarios de informática investigamos en educación, no seguimos una metodología. Quizás porque, en muchas ocasiones, ni siquiera somos conscientes de que estamos investigando. Llevamos años investigando en nuestra área de conocimiento y, en cuanto a metodología investigadora se refiere, nos encontramos en el nivel de la *competencia inconsciente*: la habilidad investigadora ya es automática. Si nuestro objetivo es reducir el consumo de un *cluster* de computadores cuando lleva a cabo computaciones matriciales, realizamos distintos tipos de acciones que persiguen reducir dicho consumo (apagar nodos cuando no se usan, reducir la frecuencia del procesador en función de la carga, etc.) y medimos el consumo durante nuestras experimentaciones, por supuesto. Un análisis cuantitativo de los datos recogidos y su posterior interpretación nos ayudará a establecer modelos a seguir para conseguir nuestro objetivo: reducir

¹ Annual Global Engineering Education Conference Sponsored by the IEEE Education Society.

el consumo energético.

Sin embargo, un profesor que busca, por ejemplo, reducir el abandono de su asignatura por parte de los alumnos, elige realizar una serie de acciones que cree que pueden ayudar a conseguirlo, pero acaba midiendo el número de aprobados sobre el número de presentados, y analizando el motivo por el cual en el grupo de la tarde las notas son mejores que en el de la mañana. En algún momento, una distracción (la diferencia de rendimiento entre los alumnos de la mañana y de la tarde) le ha apartado de su objetivo: que abandonen menos estudiantes. El trabajo no solo fue rechazado: es posible que los profesores no llegaran a saber si sus acciones tuvieron el efecto deseado.

Se puede decir que muchos profesores universitarios de informática nos encontramos, en cuanto a investigación educativa se refiere, en el nivel de la *incompetencia inconsciente*: no sabemos, y no sabemos que no sabemos. Mediante este trabajo pretendemos que, quienes se encuentren en esta situación, puedan pasar al nivel de la *incompetencia consciente*: que sepan lo que no saben. Cuando uno se encuentra en este nivel del aprendizaje, es consciente de sus limitaciones y está en disposición de aprender para compensarlas. Pasar al siguiente nivel, el de la *competencia consciente*, es sólo cuestión de ponerse manos a la obra: llevar a cabo proyectos de investigación educativa de manera sistemática. Seguir la metodología de la *investigación-acción* es nuestra propuesta.

Para presentar esta metodología, introducimos en el siguiente apartado la figura del profesor-investigador como el profesor que investiga sobre su propia práctica. La metodología que usa habitualmente el profesor-investigador es la investigación-acción, que se introduce en el apartado 3, y se presenta con detalle en los apartados 4 y 5. El apartado 4 se centra en la identificación del problema o área de mejora, y el apartado 5 recorre las etapas que siguen a esta identificación y que conforman un ciclo de investigación-acción. Cerramos este trabajo en el apartado 6 con unas conclusiones que servirán para valorar la metodología expuesta.

2. El profesor-investigador

La investigación educativa se ha realizado casi siempre lejos de las aulas. Expertos en psicología y ex-

ertos en educación elaboran teorías sobre el aprendizaje y construyen conocimiento de carácter universal que es validado experimentalmente. El profesorado, en las aulas, se encuentra muchas veces problemas diferentes a los abordados por los investigadores, y el conocimiento educativo que utiliza es el que valida con su propia práctica.

Recientemente, ha surgido una nueva manera de entender la relación entre enseñanza e investigación, considerando ambas como una actividad integrada. Desde luego, tiene sentido esta visión integrada, ante la cual podemos plantearnos si no deberían entrar en las aulas los investigadores para entender la naturaleza de la enseñanza. Sin embargo, existen multitud de argumentos [7] en contra de que los profesores sean objeto de investigación por parte de personas externas: la práctica educativa es compleja, los datos recogidos en una primera observación serán distintos de los recogidos en la siguiente, cada profesor se rige por unas teorías implícitas y unas intenciones, del mismo modo que el alumnado tiene sus propias percepciones. Demasiadas variables para tenerlas en cuenta en unas observaciones puntuales.

Sabemos que la enseñanza no es una actividad técnica, que no se rige por leyes científicas ni consiste solamente en aplicar teorías. Son los profesores, gracias a su actividad cotidiana, quienes disponen de los datos necesarios para comprender lo que sucede en las aulas.

El movimiento del *profesorado como investigador* surge, pues, de la mano de Stenhouse [11], un pedagogo británico que defendía que son los mismos profesores quienes deben estudiar su propia labor y reflexionar sobre ella de manera crítica. Se trata de que el profesorado pruebe las teorías educativas en sus clases y que, de esta manera, la teoría se desarrolle a través de la práctica. Elliot [3] va más allá: de ver la práctica como espacio de aplicación de la teoría, pasa a considerar que la reflexión sobre la práctica puede llegar a revelar la teoría inherente a la misma, y que incluso permite teorizar sobre la práctica.

La figura del *profesor-investigador* (el profesor que investiga sobre su propia práctica) ha ido cogiendo fuerza en los últimos años en los niveles de enseñanza obligatoria [7,2]. Como indica Latorre [5]:

«La literatura especializada destaca el éxito de la investigación como modelo de for-

mación de los docentes de primaria y secundaria. A través de la investigación, el profesorado puede llegar a profesionalizarse, a interesarse por los aspectos pedagógicos de la enseñanza, y a motivarse por integrar investigación y docencia; todo ello conduce a una mayor satisfacción profesional, a mejorar los programas académicos y el aprendizaje del alumnado, y a avanzar en el conocimiento educativo.

Desde esta perspectiva, la capacitación del profesorado no depende tanto de la asimilación de teorías y técnicas científicas, ni del aprendizaje de competencias didácticas como del desarrollo de la comprensión reflexiva sobre su práctica, y de la reflexión e indagación de sí mismos. Lo anterior no significa que desestimemos las competencias didácticas del docente, sino que la docencia, considerada como actividad investigadora, requiere un nuevo perfil de docente. No se trata de sustituir unas cualidades por otras, sino de integrar ambas en el perfil de docente investigador, que supere la situación actual donde docencia e investigación continúan separadas.»

En cuanto al ámbito universitario, la figura del profesor-investigador también está emergiendo gracias, en parte, a la influencia de los programas de formación del profesorado universitario [9], y también a la moderna preocupación por evaluar la eficacia de las universidades [10].

3. La investigación-acción

Entre las distintas metodologías de investigación educativa, la que se adapta al perfil del profesor-investigador es la denominada *investigación-acción*. Elliot [3] define la investigación-acción como «un estudio de una situación social con el fin de mejorar la calidad de la acción dentro de la misma». Así, la investigación-acción conlleva la comprobación de ideas en la práctica como medio para mejorar las condiciones sociales e incrementar, a la vez, el conocimiento. En este proceso de investigación la importancia principal está en la acción: es la que conduce la investigación y es la fuerza que la motiva.

Latorre [5] hace una amplia revisión de las características de la investigación-acción desde la perspectiva de distintos autores. Recogemos aquí las que consideramos de mayor relevancia:

- *Es cíclica*. Cada ciclo se inicia considerando un problema práctico, que se analiza con el objetivo de mejorar la situación en la que tiene lugar. A continuación, se elabora un plan estratégico que se lleva a cabo a la vez que se observa su desarrollo y sus resultados. Por último, se analizan estos resultados, se reflexiona y se evalúa el impacto de la acción, siendo habitual que esto dé lugar a la identificación de otra área de mejora sobre la que iniciar un nuevo ciclo.
- *Es crítica*. En la investigación-acción se reflexiona de manera objetiva sobre el proceso y sobre sus resultados; implica registrar y analizar también nuestros propios juicios, reacciones e impresiones en torno a lo que ocurre.
- *Es participativa y colaborativa*. La investigación-acción implica al investigador como foco principal de la investigación, pero debe realizarse junto a otras personas, debe ser un trabajo colaborativo ya que esta manera de trabajar enriquece todo el proceso. Estas personas serán los propios alumnos involucrados en la investigación, también profesores de la misma asignatura o profesores que buscan el mismo tipo de mejora, o simplemente compañeros que están dispuestos a compartir una discusión de trabajo con una postura crítica.
- *Es una forma rigurosa y sistemática de investigación*. La investigación-acción lleva a generar teoría desde la práctica.

La investigación-acción propone, pues, mejorar la educación a través del cambio y aprender a través de las consecuencias de los cambios. Y dichos cambios no se producirán solamente a nivel práctico, también se darán a nivel mental. Esto significa que se debe estar dispuesto a cambiar la propia comprensión no sólo de la situación real que requiere un cambio, sino también de otros aspectos relacionados con el aprendizaje, la enseñanza y la evaluación. Los estudios realizados sobre las creencias de autoeficacia docente de los profesores [8] (capacidad percibida para ayudar a nuestros alumnos a aprender) demuestran que son estas creencias el motor principal de

nuestra actuación docente. Los profesores tendemos a emplear más o menos estrategias didácticas en función del grado de confianza que tenemos para utilizarlas de forma eficaz, en función de nuestras creencias en cuanto a nuestra autoeficacia. Poner en práctica cambios requiere, o conlleva, según se vea, aumentar el sentimiento de autoeficacia. Y para ello no siempre es suficiente aprender nuevas estrategias didácticas; quizás necesitemos cambiar creencias sobre el aprendizaje, sobre la enseñanza, sobre la evaluación, sobre nosotros mismos y, cómo no, sobre nuestros estudiantes.

Una vez introducida la metodología de la investigación-acción, los siguientes apartados detallan cada una de las etapas a seguir en un proyecto de este tipo.

4. El foco de un proyecto de investigación-acción

La etapa con la que se inicia todo proyecto de investigación-acción es la identificación del área que queremos mejorar, lo que denominamos el problema o, en general, el foco del proyecto. Tras esta etapa vendrán uno o varios ciclos de investigación-acción, cuyas fases se detallan en el apartado 5.

Antes de entrar en materia, detengámonos por un momento en el profesor que va a llevar a cabo la investigación. En el apartado 2 se ha dejado constancia de que el profesor-investigador resulta un modelo adecuado para la formación inicial del profesorado, tanto de primaria y secundaria [5], como de universidad [9]. Tanto creemos que esto es así, que a nuestros estudiantes del Máster en Profesor de Secundaria que impartimos en la Universitat Jaume I les pedimos hacer uso de esta metodología para llevar a cabo su Trabajo Final de Máster. Pero no sólo en formación inicial es adecuado este método, también lo es en cualquier etapa del desarrollo profesional del docente.

Valero-García [12] nos muestra las etapas que caracterizan el desarrollo profesional de los docentes según Kugel [4]. En cada una de estas etapas, el foco de atención es distinto y, en consecuencia, las preguntas que se plantea el profesor también lo son. Como se observa en la tabla de la Figura ??, al principio de la carrera docente el profesor centra la atención en sí mismo. Pasar de esta primera etapa a la segunda

es cuestión de tiempo, ya que lo que el profesor necesita es coger confianza y sentirse cómodo en clase. Sin embargo, pasar de la segunda etapa a la tercera puede ser un proceso largo, o puede ser traumático (como en el caso del profesor Valero-García), o puede ser que todavía no haya sucedido. Plantearnos una mejora educativa mediante la investigación-acción nos permitirá entrar de lleno en esta tercera etapa ya que el foco de atención de un proyecto de este tipo deben ser *los alumnos*.

Suponiendo, pues, que nos encontramos ante una situación de nuestra práctica docente que queremos mejorar, llevar a cabo un proyecto de investigación-acción requiere plantearnos una pregunta de este estilo: «¿Qué puedo hacer para mejorar mi práctica docente?». Es esencial la manera en que se plantea la pregunta: el foco son los alumnos, pero el sujeto es el propio profesor. Así, son ejemplos correctos de preguntas: «¿Qué puedo hacer para que mis alumnos aprendan de manera autónoma? ¿Qué puedo hacer para que mis alumnos estén motivados?». Mientras que los siguientes ejemplos corresponden a formulaciones incorrectas: «¿Por qué muchos de mis alumnos acaban abandonando la asignatura? ¿Por qué mis alumnos no piensan?».

Resulta interesante, en esta etapa inicial, preguntarse también por qué hemos identificado este área de mejora, cuál es la preocupación que nos mueve a cambiar. De este modo, descubriremos algunas de nuestras creencias (y valores), ya que, como se ha citado anteriormente, son el motor de nuestra actuación docente.

5. Fases de un ciclo de investigación-acción

Tras la identificación del problema iniciaremos un primer ciclo de investigación-acción que consta de las fases que se detallan a continuación.

5.1. Fase 1. Estudio del problema

Una vez tenemos clara el área de mejora sobre la que queremos actuar, debemos realizar un reconocimiento de la misma, un diagnóstico. Necesitamos evidencias que nos sirvan de punto de partida y con las que poder comparar cuando estemos llevando a cabo nuestras acciones de mejora. Nos fijaremos también en qué contexto tiene lugar el problema, en qué medida es un problema y qué es lo que sucede

Foco de atención	Preguntas que caracterizan esta etapa
Yo mismo	¿Seré aceptado por mis alumnos? ¿Sobreviviré a la siguiente clase? ¿Descubrirán que después de todo tampoco sé tanto de este tema?
Mi temario	¿Cómo puedo explicar este tema con la mayor claridad? ¿Cuáles son los mejores ejemplos? ¿Cómo puedo mejorar mi colección de transparencias?
Mis alumnos	¿Por qué no están aprendiendo? ¿Qué otras cosas puedo hacer para que lo aprendan? ¿Cómo puedo hacer que sean más autónomos?

Figura 1: Etapas que caracterizan el desarrollo profesional de los docentes, según Kugel [4] (tabla tomada de Valero-García [12]).

realmente.

Una vez realizado el diagnóstico podemos lanzar hipótesis sobre posibles acciones y cómo podrían ayudar a mejorar la situación problemática que queremos cambiar.

Es en esta fase cuando se debe realizar una revisión bibliográfica, tanto sobre el tema de nuestra investigación como de las acciones que hemos considerado en nuestras hipótesis. Las fuentes consultadas darán solidez al informe que escribiremos más adelante para dar a conocer nuestra investigación, un informe que podrá ser enviado a la próxima edición de JENUI.

5.2. Fase 2. Elaboración de plan de acción

En este momento, nos encontramos en disposición de diseñar nuestro plan de mejora: el plan de acción. Este plan guardará coherencia con el diagnóstico del problema realizado en la fase 1 y con la información obtenida en la revisión documental realizada en esa misma fase.

El plan de acción debe ser estratégico, debe contener los pasos a realizar, el cronograma que se va a seguir, las personas que van a intervenir, las metas a conseguir. Debemos trazar planes flexibles ya que podemos encontrar limitaciones (materiales, organizativas, sociales) que requieran introducir cambios.

Además, el plan de acción debe contener una serie de indicadores que serán los que utilizaremos para medir el impacto de la mejora. Hay que prestar una especial atención aquí ya que, durante la acción, se deberá observar cualquier situación que contenga dichos indicadores. Esta observación se realizará para

tomar datos que puedan servir como evidencias del cambio. Es importante tener en cuenta la relevancia de las evidencias: deben servir para demostrar que lo que ha ocurrido es debido a la intervención realizada, que realmente ha sido su influencia la que ha provocado el cambio.

5.3. Fase 3. Acción y observación

Durante la puesta en práctica del plan de acción es necesario supervisar que ésta se lleva a cabo del modo planificado, a la vez que documentar el proceso de investigación. Se trata, pues, de obtener evidencias de que el plan de acción se está ejecutando y de que lo que se está haciendo tiene un impacto. En esta fase es fundamental registrar con detalle cómo se desarrolla la acción (por ejemplo, mediante un diario), y también recoger todos los datos relacionados con los indicadores que hayamos fijado en la fase anterior. La recolección de estos datos debe ser sistemática y, si es posible, debe llevarse a cabo desde tres perspectivas distintas:

- Observando el efecto de la acción, tanto por parte del investigador como por parte de otros colegas. El investigador puede registrar en un diario sus observaciones y a la vez reflexionar sobre lo sucedido. Cuando es otro quien nos observa, es importante que lo haga utilizando una guía de observación [1].
- Preguntando a los estudiantes implicados en la investigación mediante cuestionarios, entrevistas, grupos de discusión, etc.
- Analizando todo tipo de material relacionado con los indicadores establecidos: grabaciones

(audio o vídeo), trabajos de alumnos, diarios de clase, exámenes, etc.

El objetivo al recoger la información es reducir la situación real a una representación que nos resulte más fácil de analizar.

5.4. Fase 4. Reflexión

Con esta fase finaliza un ciclo de investigación-acción. Es el momento de analizar e interpretar los datos recogidos para obtener evidencias que prueben los efectos de la acción. Esta fase resultará compleja ya que los datos son irrepetibles y pueden ser interpretados de distintas formas; es, al fin y al cabo, una investigación cualitativa, habitual en las Ciencias Sociales y nada frecuente en las Ingenierías. Es quizás en esta fase donde más expuestos estamos a caer en incoherencias, por lo que requiere de cierto rigor. Es preciso cuestionarse cómo se llega a cada conclusión y si hay otras formas de interpretar la información; ser consciente de lo que se está dando por sentado, qué suposiciones son las que llevan a las conclusiones a las que estamos llegando; qué punto de vista se está adoptando y si hay algún otro punto de vista que se deba considerar. Desde luego que en el ámbito de la investigación social existen estrategias que permiten validar una investigación de este tipo, pero consideramos que hacer un repaso de las mismas queda fuera de la intención de este trabajo.

En ningún momento se debe perder de vista la situación que se quería mejorar para poder argumentar correctamente cómo se ha hecho evidente esa mejora.

La reflexión que se realiza tras el análisis y la interpretación de los datos debe permitirnos teorizar, construir modelos, y también adoptar una postura crítica. Una investigación-acción no tiene un punto final porque siempre plantea nuevos interrogantes. No siempre funcionan las estrategias utilizadas para dar una solución al problema; o sí funcionan, pero surgen nuevas situaciones que requieren nuevos cambios, así que es habitual que en este punto se inicie un nuevo ciclo de investigación-acción.

El último paso de esta fase consiste en elaborar un informe de la investigación que se ha llevado a cabo: el artículo que enviaremos a JENUI, u otros congresos o revistas [6]. Quienes asistimos a estas jornadas cada año sabemos que la mejor manera de validar una investigación docente es haciéndola pública en

un entorno como JENUI.

6. Conclusiones

En este trabajo hemos presentado la metodología de investigación-acción como una propuesta para realizar nuestras investigaciones educativas en el ámbito de la enseñanza universitaria de la informática. Esta metodología, además de ser una guía sistemática que nos permitirá llevar a cabo investigaciones coherentes, también nos ayudará a evolucionar en nuestro desarrollo profesional cambiando no sólo nuestras prácticas sino también nuestras ideas, nuestras concepciones sobre la enseñanza, el aprendizaje y la evaluación.

Centrar el foco de la investigación en un cambio de la práctica docente para mejorar el aprendizaje es sólo el primer paso. Realizar un completo diagnóstico de la situación concreta a mejorar, hacer una revisión bibliográfica, diseñar un plan de acción, fijar los indicadores y establecer los mecanismos para llevar a cabo la observación del proceso son pasos que constituyen, de por sí, una garantía para la mejora. Queda poner en práctica el plan de acción, observar y, a continuación, evaluar la acción y reflexionar de manera crítica. Muy probablemente, encontraremos que la situación seguirá siendo susceptible de mejora, por lo que volveremos a planificar, actuar, observar y reflexionar, en un nuevo ciclo de investigación-acción. Y, en algún momento, al cerrar alguno de estos ciclos, elaboraremos un informe para dar a conocer nuestros resultados de investigación.

Referencias

- [1] Barceló, J.M., Cortes, T. Fernández, A., García, J., Morancho, E., Valero-García, M., “Observar las clases de nuestros compañeros puede ayudarnos a mejorar (y a ellos también). Un experimento y una propuesta”, Report UPC-DAC-1999-66, Universitat Politècnica de Catalunya, 1999. Disponible en web.
- [2] Bosco, A., “La construcción de la innovación educativa en el marco de un proyecto de investigación: el caso del School+ y la investigación-acción”, *Educar*, núm. 34 (2004), pp. 131-157.
- [3] Elliot, J., *El cambio educativo desde la investigación acción*, Morata, 1993.

- [4] Kugel, P., "How professors develop as teachers", *Studies in Higher Education*, vol. 18, núm. 3, 1993.
- [5] Latorre, A., *La investigación-acción. Conocer y cambiar la práctica educativa*, Graó, 2003.
- [6] López, D., "Investigar en educación: guía práctica", JENUI 2009.
- [7] Martín Muñoz, A. Muñoz Arrabal, T., Rodríguez Cardaña, A., "Experiencias de investigación-acción sobre el aprendizaje por descubrimiento de los circuitos eléctricos", *Enseñanza de las Ciencias*, núm. 7(2) (1989), pp. 168-172.
- [8] Prieto Navarro, L., *Autoeficacia del profesor universitario: eficacia percibida y práctica docente*, Narcea, 2007.
- [9] Rodríguez Espinar, S., "Nuevos retos y enfoques en la formación del profesorado universitario", *Revista de Educación*, núm. 331 (2003), pp. 67-99.
- [10] Rodríguez Rojo, M., "La investigación cualitativa acerca sus pasos a la universidad", *Educar*, núm. 34 (2004), pp. 97-111.
- [11] Stenhouse, L., *La investigación como base para la enseñanza*, Morata, 1987.
- [12] Valero-García, M., "El desarrollo profesional del docente: una visión personal", 2010. Disponible en web.

Sesión 7A:
Telemática / Multimedia e informática gráfica

Laboratorios virtuales de redes: SÍ, inténtelo en casa

Jesús Martínez, Juan José Ortega, José Alberto Fernández

Dept. Lenguajes y Ciencias de la Computación

Universidad de Málaga

E.T.S.I. Informática

29071 Málaga

{jmcruz, juanjose, jprat}@lcc.uma.es

Resumen

Las asignaturas de redes y servicios distribuidos han adquirido un papel destacado en los nuevos títulos de grado en Informática. Habitualmente, las actividades prácticas de diseño y configuración de redes requieren de costosas infraestructuras de laboratorio, con recursos compartidos que limitan el número de alumnos que pueden simultanear su manejo. No obstante, las nuevas posibilidades que ofrece la virtualización permiten explorar alternativas realistas con routers y switches virtuales que se pueden manejar en entornos de laboratorio con poca infraestructura. Este artículo repasa las posibilidades existentes para la docencia de laboratorios de redes y argumenta las soluciones adoptadas en la Universidad de Málaga con énfasis en actividades de carácter no presencial.

Summary

Computer Networks and Distributed Services have a relevant role in the Degree on Informatics curricula. Usually, the practical tasks regarding the design and configuration of network topologies require expensive infrastructures for laboratories with shared resources (routers, switches) which limit the number of students that can manage them simultaneously. However, new trends and possibilities in virtualization allow professors to explore new ways to develop and practice with complex network topologies in a cheaper way. This paper reviews the existing approaches in virtualization to teach subjects focused on networking, and argues what are the solutions planned at the University of Málaga, with focus on on-line and distance learning activities.

Palabras clave

virtualización, laboratorios, enseñanza no presencial, redes, telemática

1. Introducción

Las materias de redes y servicios distribuidos siempre han contado con un papel destacado en los títulos de ingenierías en Informática (y, obviamente, en Telecomunicación). Como no podría ser de otra forma, este papel se ha reforzado en los nuevos grados adaptados al Espacio Europeo de Educación Superior (EEES), en el que el trabajo continuado del alumno y las actividades prácticas son fundamentales para la correcta adquisición de las competencias profesionales.

Dentro de la materia de redes de ordenadores, se articulan habitualmente asignaturas de laboratorio en las que se configuran, administran y mantienen topologías de red que permiten la ejecución de distintos tipos de servicios distribuidos. Tanto ahora como en los antiguos planes de estudio, los docentes se enfrentan a serios problemas para realizar prácticas de diseño de redes, puesto que los equipos de interconexión requeridos (routers, switches) suponen un desembolso importante para un laboratorio, además de la limitación de alumnos que pueden manejarlos de forma simultánea. Es también necesario reseñar que el mantenimiento de dichos equipos durante el curso supone un esfuerzo mayor que el de los ordenadores de un laboratorio estándar. Por último, debe también existir un plan de renovación de equipamientos, puesto que la evolución de las tecnologías para redes es continua: evolución de protocolos inalámbricos, incremento del ancho de banda, migración IPv6, VoIP (voz sobre IP) y calidad de servicio, etc.

Sin embargo, las nuevas posibilidades que ofrece la virtualización permiten repensar la forma en que pueden impartirse asignaturas de esta índole, donde las complejas infraestructuras físicas se pueden sustituir por PCs y equipos de interconexión virtuales con todo el software (y la configuración necesaria) para prácticas disponibles en el momento en que se vayan a utilizar. Puesto que dichos entornos virtuales

van a ser modificados en el contexto de una sesión de laboratorio, se minimiza el riesgo de dañar equipos de forma permanente y afectar al normal funcionamiento de sesiones posteriores.

Este artículo repasa las posibilidades existentes para la docencia de laboratorios de redes y argumenta las soluciones adoptadas en la Universidad de Málaga para los grados en Informática y máster en Telemática con énfasis en actividades de carácter no presencial. Nuestro objetivo es proporcionar a los estudiantes un entorno gráfico completo para diseñar y gestionar una red de ordenadores a través del campus virtual. El requisito principal es que las prácticas deben ser realistas y, por lo tanto, siempre que se pueda se deben manejar equipos y tecnologías existentes en el mercado de los equipos para redes.

El artículo está organizado de la siguiente forma. La sección 2 introduce la situación actual de los laboratorios en el contexto de la Universidad de Málaga y la sección 3 presenta las propuestas de virtualización que han sido adoptadas de forma piloto para éste y los cursos venideros. La sección 4 presenta algunos ejemplos que están disponibles en el entorno virtual y la sección 5 discute propuestas relacionadas que han sido la inspiración para el modelo de virtualización presentado. Por último, se plantean las conclusiones más interesantes y nuestras líneas de trabajo futuro.

2. Prácticas de redes en el contexto de la Universidad de Málaga

Los actuales títulos en extinción tienen una previsión de créditos específicos para la enseñanza práctica de conceptos relacionados con las redes de ordenadores, que están adscritos al área de Ingeniería Telemática. Así, en Ingeniería Informática existe una asignatura optativa de seis créditos denominada *Diseño de Redes Telemáticas*. También en Ingeniería Técnica en Informática de Sistemas y de Gestión se imparten *Diseño de Redes Telemáticas* y *Laboratorio de Redes*, ambas de seis créditos. En estas asignaturas no se venían realizando prácticas que manipulasen directamente equipos de interconexión en laboratorio, sino que se realizaban actividades sobre la infraestructura de red existente. Una de las razones que han motivado este hecho es el abundante número de estudiantes que, tradicionalmente, cursan estas

asignaturas.

Los nuevos grados en Informática pretenden corregir esta situación al asignar grupos de prácticas más pequeños aunque, en el caso de los laboratorios de redes, sigue resultando complicado contar con equipos e infraestructura suficientes para que los estudiantes puedan trabajar simultáneamente en el aula. En la Universidad de Málaga se pueden cursar desde este año tres nuevos grados, los de Ingeniería Informática, Ingeniería del Software e Ingeniería de Computadores. El primero de ellos cuenta con una asignatura troncal que se impartirá en cuarto curso denominada *Administración de Redes y Sistemas* (6 créditos ECTS) y en los tres grados existe una optativa denominada *Redes Inalámbricas* (6 créditos ECTS).

Además, el área de Telemática también se encarga de docencia en la ETSI Telecomunicación relacionada con los laboratorios de redes dentro del Máster de Telemática y Redes de Telecomunicación. Aquí encontramos *Diseño y Configuración de redes Telemáticas* (obligatoria) y *Software de Comunicaciones Empotrado* (optativa), aunque hay que reseñar que en el Máster si hay equipos de interconexión para prácticas de la marca Avaya/Nortel Networks (dos routers core, dos de acceso, dos balanceadores de carga, dos cortafuegos y dos centralitas VoIP). Por último, también en los nuevos grados de telecomunicación se impartirán materias relacionadas, como la de *Administración de Equipos y Sistemas en Red* (obligatoria en el grado de Ingeniería Telemática) y otras.

La nueva realidad del Espacio Europeo de Educación Superior demanda nuevas metodologías didácticas, y fomenta el uso de herramientas TIC, como el Campus Virtual. La siguiente sección presenta las decisiones que se han tomado para garantizar la adquisición de competencias profesionales en el sector de las redes, donde se requiere del manejo de equipos de interconexión como routers y switches. La experiencia piloto basada en virtualización se va a desarrollar a partir del segundo cuatrimestre del curso 2010-11, entre las asignaturas de *Laboratorio de Redes* (I.T. Informática de Sistemas) y *Software de Comunicaciones Empotrado* (Máster).

3. Propuestas de virtualización y para enseñanza no presencial

Nuestro principal objetivo a la hora de preparar contenidos para laboratorios de redes mediante virtualización es buscar un entorno de trabajo con las siguientes características:

- Facilidad de configuración: buscamos un entorno gráfico que permita a nuestros estudiantes el diseño y gestión de topologías de red compuestas por ordenadores finales (PCs) y equipos de interconexión (routers y switches)
- Entornos de trabajo realistas: para desarrollar las competencias demandadas por el sector empresarial en materia de diseño y administración de redes los estudiantes deben manejar equipos reales, preferiblemente a través de emuladores y simuladores de marcas de fuerte implantación en el mercado. Si es posible, se priorizarán herramientas de código abierto
- Posibilidad de integración en el Campus Virtual, para que los estudiantes puedan flexibilizar su aprendizaje (haciendo hincapié en sesiones no presenciales) y para que los profesores puedan corregir las actividades fuera de línea, en otro momento.

Con estas premisas, se estudiaron y por fin eligieron tres herramientas básicas: GNS3 [1] como diseñador gráfico de redes, XORP [2] como router de código abierto con una interfaz de configuración y funcionalidades similares a routers de reconocido prestigio y, VirtualBox [3] y VMWare ESX [5] para la virtualización a distintos niveles.

3.1. GNS3

GNS3 son las siglas de Graphical Network Simulator. Esta aplicación está enfocada a la simulación de redes complejas para su estudio. Este entorno visual usa como motor de ejecución la plataforma Dynamips/Dynagen [6], creada originalmente para ejecutar y emular el firmware de los routers y dispositivos de Cisco Systems (IOS) y Juniper (JunOS), además de establecer topologías (escenarios) para conectarlos. Dichas topologías pueden, a su vez, interconectarse con otras ejecutándose en diferentes instancias de GNS3, lo que se denomina red virtual, bien en el mismo o en diferentes PCs (hacien-

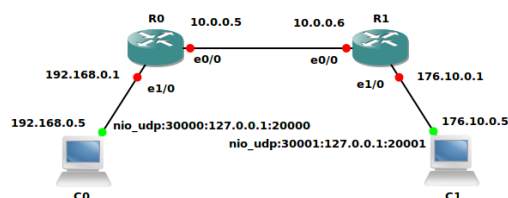


Figura 1: Topología de ejemplo

do uso de los elementos de red disponibles en las máquinas). Este último caso es muy interesante para diseñar actividades donde diferentes grupos de estudiantes cooperan para conseguir la interconectividad completa de cada topología (sistema autónomo, en terminología de redes) de la que son responsables/administradores.

La construcción de escenarios en GNS3 es muy intuitiva gracias a su interfaz visual. Mediante un sistema de arrastrar y soltar elementos se distribuyen los routers y concentradores definidos sobre el área de trabajo y, tras definir en sus características los interfaces de los que dispondrán, se unen mediante “cables” seleccionando con el ratón las interfaces origen y destino. Un ejemplo lo constituye la figura 1.

Una vez definida la topología de la red, los estudiantes pueden proceder a configurar el comportamiento de cada uno de los equipos de interconexión implicados, estableciendo las propiedades de los protocolos de encaminamiento que vayan a ser utilizados. Esta tarea se realiza a través de la consola de comandos del dispositivo en cuestión, accedida mediante emulación por Dynamips/Dynagen.

Las configuraciones de escenarios en GNS3, pueden ser guardadas en formato de archivo de texto. Además de las topologías, la configuración de cada router también se almacena, y puede ser importada para otro dispositivo. Esta última característica es tremendamente útil para labores de evaluación de prácticas, puesto que los profesores pueden pedir como entregables los archivos de configuración de los equipos y replicar el resultado de la actividad del estudiante con absoluta fidelidad.

Para inyectar tráfico a los escenarios de red creados con GNS3 y comprobar si su comportamiento es el esperado, los estudiantes tienen la posibilidad de conectar elementos externos simulándolos como

nubes a las que se asignan diversas interfaces de red externas (reales) del PC anfitrión. Las interfaces de una *nube* se denominan NIO (network input/output) dentro de GNS3, y existen distintos tipos. Las NIO UDP permiten la conexión a un puerto de un host (el anfitrión en las actividades desarrolladas para los estudiantes), y han sido utilizadas para conectar aplicaciones ligeras llamadas VPCS que simulan PCs sencillos con herramientas de diagnóstico de red (para realizar pings o trazar rutas de paquetes). Otros tipos de interfaces NIO en GNS3, como las NIO Ethernet (para conectarse a una tarjeta de red Ethernet), las NIO UNIX (para tomar las conexiones desde archivos de dispositivo en UNIX) o las NIO TAP (interfaces de kernel virtuales tipo puente de nivel dos), se han empleado en nuestra propuesta para interactuar con otros tipos de dispositivos de interconexión sin hacer uso de Dynamips/Dynagen.

Otra posibilidad útil de diagnóstico lo constituye la capacidad de GNS3 para registrar los paquetes que atraviesan un interfaz seleccionado. Estos registros se almacenan en un archivo de traza compatible con el analizador de protocolos gratuito Wireshark [4].

Sin embargo, hay dos cuestiones que hay que abordar con cautela a la hora de trabajar con GNS3. En primer lugar, está el tema de las licencias. Si bien GNS3 y Dynamips/Dynagen son programas gratuitos, no lo son las licencias de los sistemas operativos de los dispositivos de Cisco o Juniper. Por lo tanto, las prácticas de laboratorio podrán utilizar IOS o JunOS siempre que la Universidad disponga efectiva y legalmente de dichas imágenes del firmware del sistema operativo y, obviamente, estas imágenes no pueden ser entregadas a los estudiantes para actividades fuera del aula. Otro detalle importante es el consumo de recursos de los emuladores. Hemos comprobado que topologías con más de cinco o seis routers pueden ser bastante pesadas para ejecutarse sobre un PC estándar. Por todo esto, nos pareció importante buscar alternativas a los routers de Cisco y Juniper, y las encontramos.

3.2. Un router de código abierto: XORP

XORP son las siglas de eXtensible Open Router Platform, un proyecto GNU para la creación de una plataforma de código abierto que permite la creación de dispositivos de interconexión software multipla-

taforma. XORP tiene una arquitectura modular y flexible, que permite implementar distintos protocolos de encaminamiento, los cuales se cargarán y ejecutarán cuando sea requerido por el administrador del dispositivo.

La versión con la que se ha trabajado en nuestra propuesta para laboratorios (1.6) incluye los siguientes módulos de protocolos y tecnologías de red: RIP (Routing Information Protocol), OSPF (Open Short-Path First), BGP (Border Gateway Protocol), multicast, IGMP (Internet Group Management Protocol), MLD (Multicast Listener Discovery), SNMP (Simple Network Management Protocol) y cortafuegos. XORP se distribuye de manera gratuita desde su página web a través de live CD o como fuentes para Windows y Unix/Linux. De nuevo, y siguiendo nuestra propuesta, XORP podría ser virtualizado sobre el ordenador anfitrión que esté ejecutando GNS3. Para ello, se puede recurrir a cualquier plataforma de virtualización del mercado. En nuestro caso, hemos hecho pruebas con VirtualBox [3], de licencia libre y que ha resultado consumir menos recursos al ejecutar XORP que un emulador Dynamips/Dynagen ejecutando, por ejemplo, una IOS de Cisco.

Hay que destacar que las soluciones de virtualización no sólo consisten en emular sistemas operativos o máquinas dentro de un anfitrión, si no que también es posible crear redes para interconectar dichas máquinas virtuales. Para crear estas redes virtuales se ha hecho uso de interfaces de kernel tipo TAP, que son reconocidas por VirtualBox. En nuestro caso, se han configurado puentes de nivel dos para establecer la conexión entre las interfaces de las máquinas virtuales que ejecutan el router XORP.

El ejemplo presentado en la figura 2 imita al mostrado en la figura 1 del apartado anterior con la salvedad de que se hace uso de máquinas virtuales con XORP y de las interfaces y puentes necesarios para establecer las conexiones. En líneas de color rojo se muestran los enlaces que están siendo simulados por el resto de componentes. Para estudiar el tráfico de paquetes que circulan por esta red virtualizada, un analizador de protocolos tipo Wireshark puede escuchar directamente a través de la interfaz TAP deseada, puesto que aparecen disponibles en el sistema operativo.

Con respecto a las diferencias entre la configuración de un router tipo Cisco y uno XORP, los estu-

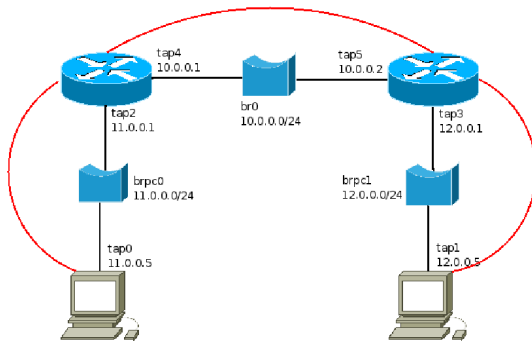


Figura 2: Uso de interfaces TAP y bridges para crear redes virtuales con XORP

diantes se encuentran con una consola de configuración similar (CLI o command-line interpreter). La única diferencia reseñable, y que no afecta a la funcionalidad de las actividades de prácticas es que en XORP los protocolos se configuran para cada interfaz, al contrario que en los routers de Cisco donde en el protocolo se indica para qué interfaces estará activo.

Como se ha comentado anteriormente, las pruebas realizadas con XORP han sido muy satisfactorias, y el consumo de recursos en el PC anfitrión se reduce considerablemente al usar routers XORP, por lo que las topologías pueden contener más elementos y realizar tareas de encaminamiento más complejas. XORP no tiene todas las capacidades de un router comercial, pero los protocolos de que dispone son más que suficientes para un buen curso de administración y configuración de redes a nivel de grado en Informática.

Obviamente, sería deseable combinar las fortalezas de GNS3 para creación y gestión de topologías a nivel visual y XORP con su bajo consumo y carencia de licencias propietarias. Para poder usar XORP como router dentro de GNS3 hemos hecho uso de las capacidades de este último para conectarse al exterior mediante interfaces NIO TAP. De esta forma, utilizamos el elemento *nube* de GNS3 y se selecciona el TAP adecuado, que aparece disponible en su menú de configuración. El resultado sigue siendo intuitivo y sencillo para los estudiantes, puesto que su PC de laboratorio contendrá: la herramienta GNS3, las imágenes de routers de emulación con Dynmips/Dynagen y/o las imágenes de routers XORP

como máquinas virtuales de VirtualBox juntos a los correspondientes interfaces TAP preconfigurados.

3.3. PCvirtual en el Campus Virtual

El proyecto de PCvirtual de la Universidad de Málaga se desarrolló en el marco del proyecto de la Junta de Andalucía denominado *Universidad Digital*. En dicho proyecto se estableció que se iban a compartir asignaturas de libre configuración entre las diez universidades andaluzas. Al compartir la docencia totalmente no presencial se vio la necesidad de que programas informáticos y recursos necesarios en dichas asignaturas fueran utilizados por los alumnos de otras universidades. Esto hizo que se desarrollara un sistema para que los alumnos de las otras universidades accedan remotamente a programas, evitando que tengan que ser instalados en todas las universidades. El sistema que se ideó es lo que llamamos PCvirtual. Este consiste en una infraestructura de máquinas virtuales que se crean dinámicamente y mediante el acceso con el protocolo RDP (Remote Desktop Protocol). La infraestructura la componen un broker que gestiona las máquinas virtuales creadas en VMware ESX. Esta misma instalación es la elegida también para las prácticas de las asignaturas presenciales. Con esto se consigue que los estudiantes tengan acceso remoto a la máquina virtual con la instalación necesaria para ejecutar GNS3, XORP y TAP, evitando así los problemas de instalación de versiones incorrectas y replicación del entorno de laboratorio en casa.

4. Actividades y ejemplos de uso

Para que los estudiantes puedan habituarse con rapidez al entorno de trabajo propuesto, se ha creado un manual de uso en el que se presentan las posibilidades de la herramienta y se presentan topologías de ejemplo y la configuración de equipos para distintos protocolos de encaminamiento. Estos ejemplos incrementan gradualmente su dificultad y van, desde los más básicos de interconexión (tablas de encaminamiento estáticas) hasta la configuración de routers mediante RIP, BGP, OSPF. También se incluyen otras configuraciones avanzadas relacionadas con tecnologías Frame Relay y ATM (Asynchronous Transfer Mode).

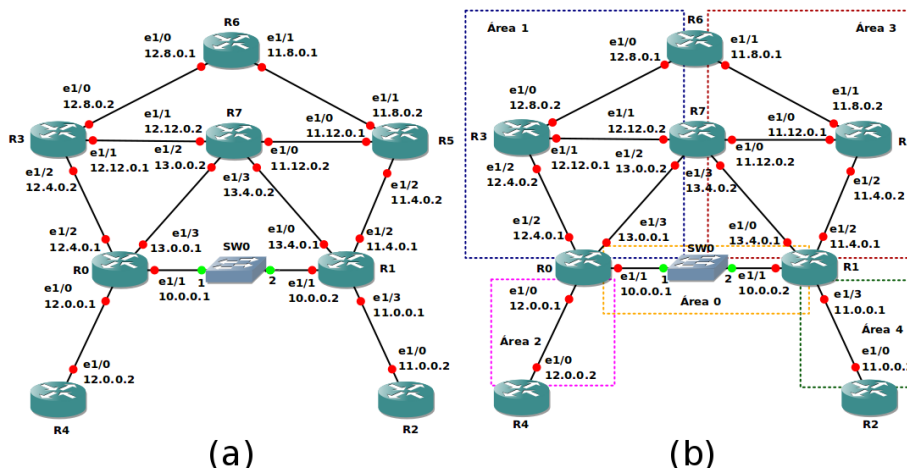


Figura 3: Ejemplo de uso de OSPF

Por ejemplo, los laboratorios presentados en la figura 3 hacen uso del protocolo OSPF con el fin de estudiar cómo emplea este protocolo el algoritmo de Dijkstra para buscar el mejor camino para los paquetes, teniendo en cuenta factores como la carga de los enlaces o el destino del datagrama. La figura presenta dos casos para conocer la importancia de la división en áreas: en un primer laboratorio se ha probado la red sin ninguna división (a), y en el segundo se ha descompuesto ésta en cinco regiones (b).

En otros laboratorios se proponen también topologías con tecnologías mixtas, es decir, con routers de distintos fabricantes, de forma que los estudiantes pueden comprobar su interoperabilidad gracias a los protocolos de encaminamiento estándar.

En otros ejemplos más avanzados, como en el laboratorio presentado en la figura 4, se puede combinar la creación de redes privadas virtuales (CE_1/PE_1 y CE_2/PE_2, en la figura) sobre redes gestionadas por MPLS (Multi-Protocol Label Switching) con salida al exterior a través de una pasarela (GW).

Con todo, se ha planificado una pequeña experiencia piloto para obtener los primeros resultados, tal y como se muestra en el cuadro 1. Los contenidos abarcan tanto sesiones de laboratorio presenciales como no presenciales. Las primeras se realizan al principio y presentan el entorno de laboratorio y

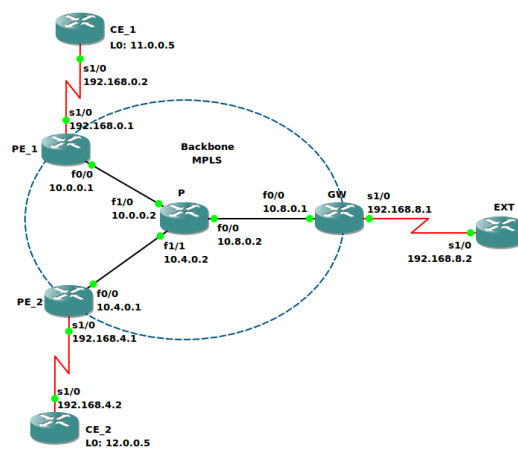


Figura 4: Ejemplo de uso de VPN

las herramientas para que el estudiante aborde las siguientes tareas. La configuración de los laboratorios propuestos se puede hacer de forma no presencial, teniendo en cuenta que el contexto de partida asume conocimientos previos acerca de los protocolos y algoritmos de encaminamiento a utilizar para la resolución de las actividades. Como se comentó antes, los entregables para la evaluación deben incluir tanto los proyectos de GNS3 como cada archivo de configuración de los routers que componen la red pro-

Actividad	Descripción	Horas	Carácter
Introducción	Conceptos de Virtualización y GNS3. Infraestructura de laboratorio	2	presencial
Interfaces	Manejo de consola (IOS) de los routers	2	presencial
Administración	Analizadores de protocolos y comandos de depuración (debug, ping, ...)	1	presencial
Encaminamiento (I)	Activación de interfaces y rutas estáticas. Activación de RIP y depuración	2	no presencial
Encaminamiento (II)	Activación de OSPF y depuración	3	no presencial

Cuadro 1: Planificación y contenidos de la experiencia piloto

puesta en cada ejercicio.

5. Trabajos y propuestas relacionadas

Las soluciones basadas en virtualización están siendo cada vez más demandadas en la industria por su importante reducción de costes en infraestructura y gestión de servidores. En la Universidad también se están haciendo esfuerzos importantes, que no se reducen a un contexto local de laboratorios o asignaturas concretas. Basta con entender el impulso que ha representado, por ejemplo, el proyecto de *Andalucía Digital* para entender que la virtualización juega y jugará un papel muy destacado en el Espacio Europeo de Educación Superior.

En el contexto de los laboratorios de redes, el proyecto más completo y de mayor relevancia a nivel internacional lo constituye VNUML (Virtual Network User Mode Linux) [7, 8], que ha servido de punto de partida e inspiración para la solución adoptada en la Universidad de Málaga. VNUML es una herramienta de virtualización que permite la definición y pruebas de escenarios de simulación de redes. Orientada a Linux, se apoya en el software de virtualización UML (User Mode Linux), que permite ejecutar máquinas virtuales Linux desde otro Linux anfitrión. La herramienta VNUML está formada por dos componentes: el lenguaje VNUML para describir simulaciones (topologías y guiones de ejecución) en XML, y el intérprete en línea de comandos, que ejecuta el escenario de forma automática, sin intervención por parte del usuario. Nuestra apuesta por GNS3 está fundamentada por su simplicidad, al ser una herramienta completamente visual y multiplataforma, pero VNUML permite opciones para configuraciones complejas en Linux muy interesantes.

Respecto a las tecnologías para virtualización, existen algunas opciones diferentes a VirtualBox y VMWare que son de alto rendimiento en Linux y de código abierto, como KVM [9] y OpenVZ [10]. Alguna de estas tecnologías puede convertirse en una opción futura para virtualizar imágenes de routers minimizando los recursos de la máquina anfitriona.

Por último, y muy recientemente, algunos fabricantes de equipos de interconexión han liberado sus sistemas operativos a licencias de código abierto. Tal es el caso de los routers de MikroTik [11] y Vyatta [12]. El caso de Vyatta es destacable, puesto que permite descargar un router de amplia funcionalidad (aunque, por ahora, no soporta protocolos de enrutamiento multicast como sí hace XORP) basado en la suite Quagga [13] para FreeBSD, Linux, Solaris y NetBSD.

6. Conclusiones

La virtualización de laboratorios constituye una baza importante para cumplir los requisitos de aprendizaje activo del EEES. Una buena infraestructura de virtualización proporcionada por la Universidad permite, primero a los docentes, preparar material práctico que depende de instalaciones software y hardware complejas, en forma de máquinas virtuales. Estas máquinas serán utilizadas por los estudiantes de una forma cómoda tanto en el aula como de forma no presencial, a través de los portales de Campus Virtual, con un importante ahorro en costes de infraestructura (crítico en el caso de los equipos de interconexión para redes) y tiempo (de instalación de software en casa y acceso a laboratorios por turnos, por ejemplo). En el artículo se han presentado las soluciones que van a ser adoptadas en la Univer-

sidad de Málaga para los grados en Informática y el Máster de Telemática y Redes de Telecomunicación, buscando la obtención de un entorno amigable apto para realizar laboratorios y prácticas realistas de redes tanto en modo presencial como no presencial. Mientras implantamos y evaluamos la experiencia por primera vez en este curso, seguimos trabajando en incorporar equipos de interconexión virtuales con más funcionalidad y, actualmente, se están evaluando las posibilidades del router de código abierto de Vyatta. También se están realizando pruebas para determinar qué configuraciones son más adecuadas para minimizar el consumo de recursos en el PC virtual.

Agradecimientos

Los autores agradecen a la ETSI Informática y a Enseñanza Virtual y Laboratorios Tecnológicos de la Universidad de Málaga el apoyo y estímulo para el estudio e implantación de estas soluciones para virtualización de laboratorios de redes.

Referencias

- [1] Graphical Network Simulator, *GNS3*, disponible en <http://www.gns3.net>, 2011.
- [2] Handley, M., Kohler, E., Ghosh, A., Hodson, O., and Radoslavov, P., *Designing extensible IP router software*, Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation - Volume 2, pp. 189–202, 2005.
- [3] Oracle, *VirtualBox*, disponible en <http://www.virtualbox.org>, 2011.
- [4] Riverbed Technology, *WireShark protocol Analyzer*, 2011
- [5] VMware, *VMware virtualization solutions*, disponible en <http://www.vmware.com>, 2011.
- [6] Fillot, C., *Dynamips: Cisco 7299 Simulator*, disponible en <http://www.ipflow.utc.fr/blog/>, 2011.
- [7] Galán, F., Fernández, D., *Virtual Network User Mode Linux (VNUML)*, disponible en <http://neweb.dit.upm.es/vnumlwiki>, 2011.
- [8] Ruíz, F., Fernández, D., Galán, F., Bellido, L., *Modelo de Laboratorio Docente de Telemática basado en Virtualización Distribuida*, VII Jornadas de Ingeniería Telemática (JITEL 2008), Alcalá de Henares (Madrid), 2008
- [9] RedHat Emerging Technology Project, *Kernel Based Virtual Machine*, disponible en <http://www.linux-kvm.org>, 2011
- [10] OpenVZ, *OpenVZ: container based virtualization for Linux*, disponible en <http://wiki.openvz.org>, 2011
- [11] MikroTik *MikroTik routers & wireless*, disponible en <http://www.mikrotik.com>, 2011
- [12] Vyatta *Vyatta Open Networking*, disponible en <http://www.vyatta.com>, 2011
- [13] The Quagga Project *Quagga Software Routing Suite*, disponible en <http://www.quagga.net>, 2011

EVALUACIÓN CONTINUA DE ALUMNOS MEDIANTE EL USO DE DISPOSITIVOS MÓVILES BASADOS EN iOS.

Carmen Martínez-Cruz Antonio J. Rueda

Departamento de Informática
Universidad de Jaén
Campus de las Lagunillas S/N.
23071 Jaén
{cmcruz, ajrueda}@ujaen.es

Resumen

Con la aparición de los dispositivos táctiles basados en iOS de Apple© se ha mejorado la usabilidad de las aplicaciones móviles debido a la potencia de estos sistemas, especialmente, su interfaz de usuario. Por este motivo y por su gran expansión en el mercado actual, se ha desarrollado para una aplicación para estos dispositivos, donde se mejora el proceso de evaluación continua del alumnado. Dicha aplicación permite satisfacer las necesidades del profesorado en el aula, donde se requiere evaluar al alumno in situ de forma remota, rápida, sencilla, y segura. En este artículo se presenta un primer prototipo de esta herramienta.

Summary

With the advent of devices based in Apple© iOS, the usability of mobile applications has been improved because of its features of intuitiveness, ease of use and efficiency, among others. Consequently and due of these systems expansion in the current market, an application based on this technology has been developed. This application improves the process of continuous evaluation of students in the classroom practices. This application meets the needs of teachers to evaluate the students in a remote, fast, easy and secure way. In this paper a first prototype of this tool has been presented.

Palabras clave

Dispositivos móviles, evaluación continua, dispositivos táctiles, iOS, aplicación web.

1. Motivación

La elaboración de herramientas que apoyen al profesorado para la evaluación de alumnos en el aula no es una idea nueva. No obstante la utilización de las nuevas tecnologías para la realización de estas actividades, no sólo es recomendable dada el área de conocimiento en la que nos encontramos, sino porque abre nuevas posibilidades docentes e investigadoras gracias a la funcionalidad que estas herramientas incorporan.

En este trabajo se presenta una herramienta software de apoyo a la evaluación docente basada en el uso de las nuevas tecnologías móviles que son de máximo interés en la actualidad. Se trata de las herramientas móviles proporcionadas por Apple©: iPhone©, iPad© o iPod touch©. Todas ellas son compatibles dado que utilizan un sistema operativo común, el iOS©. Además, estos dispositivos presentan una característica diferenciadora frente a generaciones anteriores de dispositivos móviles: un entorno amigable, intuitivo y sencillo que permite realizar cualquier actividad de manera rápida y cómoda. Este es claramente, uno de los principales motivos que han conducido a la aparición de innumerables aplicaciones para esta plataforma en los últimos años.

La herramienta que aquí se plantea se enmarca dentro de una asignatura de programación del grado de Informática, aunque podría ser utilizada en cualquier asignatura donde se requiera la evaluación por parte del profesor de una serie de prácticas en el aula u otro entorno donde el docente no tenga un ordenador a su alcance para tomar nota o visualizar el perfil del alumno, o donde el acceso a este ordenador implique

pérdidas de tiempo o interrupciones en el proceso de evaluación.

Dicha herramienta consiste en una aplicación móvil, con conectividad a la red (Wifi o 3G) que permite acceder y gestionar los datos que se encuentran ya accesibles a través de una aplicación web de gestión de alumnos. La aplicación móvil aquí propuesta ha aparecido frente a la necesidad del profesorado de calificar las prácticas del alumno in situ, y como alternativa al acceso a la web a través de navegadores de dispositivos móviles, cuya usabilidad plantea ciertos inconvenientes.

A continuación, en la sección 2, se describe brevemente el motivo por el cual se ha desarrollado un sistema de evaluación de alumnos alternativo al ofrecido por la Universidad de Jaén. En la sección 3, se hace un recorrido por los dispositivos móviles y táctiles que existen en relación con la docencia universitaria y especialmente por aquellas características que proporciona el sistema en el que está basada esta propuesta. La herramienta desarrollada se encuentra descrita en detalle en la sección 4. Por último, las conclusiones y los trabajos futuros se incluyen en la sección 5 de este artículo.

2. Sistemas Web de Control de Alumnos

2.1. Aplicaciones de evaluación en la Universidad de Jaén

Todas las universidades españolas disponen de sistemas de docencia virtual o plataformas de docencia que les permiten distribuir el contenido de las asignaturas, colaborar y comunicarse con el alumnado, evaluarlo, etc. [1]. Existe un gran número de aplicaciones, entre las que se destacan ILIAS, Moodle, Sakai, Blackboard, etc. Todas estas herramientas comparten muchas funcionalidades y depende de la organización, la utilización de unas u otras (véase [7]).

Concretamente ILIAS, la herramienta proporcionada por la Universidad de Jaén, a pesar de sus múltiples funcionalidades [2] presenta algunos inconvenientes, entre el que se destaca la imposibilidad de evaluar de manera continua al alumno. Además, este sistema únicamente permite realizar pruebas cerradas como, ejercicios tipo test en múltiples formatos o actividades para subir

ejercicios. Sin embargo no permite generar una estructura de la asignatura donde se puedan definir las diferentes prácticas o actividades a realizar por parte del alumnado y dichas actividades sean ponderadas dependiendo de su importancia, por parte del profesor. A su vez, tampoco es posible obtener de forma automática una calificación final del alumno en base a estas actividades.

De esta forma se han promovido numerosas iniciativas de innovación docente para impulsar al profesorado a proponer nuevas herramientas que satisfagan sus necesidades docentes, entre ellas, se destacan algunas propuestas de las últimas jornadas de innovación docente de dicha universidad [3], donde se proponen plataformas alternativas para la evaluación del alumnado en tutorías y otras actividades desarrolladas en el aula, incluyendo ponderaciones, comentarios, etc.

2.2. Aplicaciones para la evaluación continua del alumnado

Motivados por las carencias encontradas en la plataforma de docencia virtual ILIAS, solicitamos un proyecto de innovación docente de la Universidad de Jaén para el desarrollo de una herramienta que permita llevar un control del alumnado, tanto de sus datos personales como de las calificaciones obtenidas en su evaluación continua durante el curso académico. Dicha aplicación permite realizar las siguientes tareas, a través de diferentes perfiles de usuario, de administrador y de alumno.

En el perfil administrador encontramos:

- *Gestión de asignaturas*: Permite la creación de una asignatura y el diseño de su sistema de evaluación. Así pues esta aplicación permite definir el número de actividades/prácticas de la asignatura, la importancia de cada una, y los trabajos alternativos junto con su puntuación máxima. Por otro lado, también permite definir los grupos de prácticas que componen la asignatura (véase figura 1).
- *Inspección de grupos de teoría y grupos de prácticas*. A través de este enlace el profesor tiene acceso a los listados de alumnos, agrupados por grupo de teoría o grupo de prácticas.

profe/formulario_asignaturas.php?editar=16

Curso 2010 - 2011 Activa

Nombre Estructura de Datos y de la Información II (2º ITIG)

Grupos de teoría 2

Contribuciones a la nota final de la asignatura

Teoría (0-100%)	70
Prácticas (0-100%)	30
Prácticas vol. (0-10 puntos)	1
Trabajos vol. (0-10 puntos)	1

Contribuciones a la nota final de prácticas

Práctica 1 (0-100%)	10
Práctica 2 (0-100%)	40
Práctica 3 (0-100%)	30
Práctica 4 (0-100%)	20
Práctica 5 (0-100%)	0

Horario y aula

lunes 8:30 a 10:30 Prof. Carlos	<input type="checkbox"/>
lunes 10:30 a 12:30 Prof. Lidia	<input type="checkbox"/>
lunes 12:30 a 14:30 Pro. Lidia	<input type="checkbox"/>
lunes 15:30 a 17:30 Prof. Antonio	<input type="checkbox"/>
lunes 17:30 a 19:30 Prof. Antonio	<input type="checkbox"/>

Borrar

Figura 1. Creación de una nueva asignatura en la aplicación Web.

- *Ficha de los alumnos.* En esta ficha el profesor puede examinar los datos del alumno y sus calificaciones, con la posibilidad de modificar libremente cualquiera de los campos. Existen también campos de comentarios públicos (visibles por el alumno) y privados para realizar anotaciones de interés (figura 2).
- *Listados de calificaciones finales.* En este apartado el profesor podrá generar los listados de calificaciones finales de los alumnos, calculadas de manera automática a partir de los datos de cada ficha personal.

Desde el perfil alumno:

- *Ficha del alumno.* El alumno tiene acceso a su ficha, donde podrá indicar sus datos personales y seguir las calificaciones obtenidas en la parte teórica y práctica de la asignatura, trabajos voluntarios y la nota global final de la asignatura. Otra funcionalidad interesante es la selección o cambio de grupo de prácticas, que será posible

siempre que existan plazas disponibles en el grupo seleccionado.

Foto No se ha... archivo
Formato: png o jpg. Tamaño máximo: 100Kb.

DNI 123456

Apellidos Martínez Cruz

Nombre Carmen

Localidad Granada Provincia Granada

Teléfono E-mail cmcruz@ujaen.es

Clave de acceso Repetir clave

Comentarios del alumno
Una alumna cuyo comportamiento en clase deja mucho que desear.

Comentarios del profesor (privado)

Grupo de teoría Grupo de prácticas No hay grupo asignado

Notas de prácticas

Notas de teoría	1	0	
Diciembre	2	0	Adicionales
Febrero	3	0	Prac. Vol.
Junio	4		Trab. Vol.
Septiembre	5		
Media	0		

Nota Final

Figura 2. Ficha del Alumno en la aplicación web. Vista del Profesor.

Así pues, el profesorado utiliza esta aplicación tanto para obtener una nota final del alumnado atendiendo a los criterios de evaluación de la asignatura, como para realizar su evaluación de forma continua. Por otro lado, el alumno tiene acceso en todo momento a su ficha y comunicación con el profesor a través de los comentarios que éste realiza en la misma. Desde el punto de vista técnico es interesante mencionar que la aplicación web ha sido desarrollada en PHP usando SQLite como gestor de base de datos

3. Clientes móviles táctiles

3.1. Aplicaciones para dispositivos móviles

Con la venida de los dispositivos móviles (teléfonos o PDAs) y el acceso asequible a red, bien sea a través de 3G o a través de WIFI, se ha facilitado el acceso a cualquier página web desde cualquier ubicación. Sin embargo, la navegación y uso de aplicaciones web a través de dispositivos móviles deben sufrir algunas modificaciones para mejorar su funcionamiento:

- Aminorar la carga de imágenes y elementos pesados de las páginas web.
- Mejorar la distribución de los enlaces, para facilitar la navegación por teclado o táctil.
- Ajustar a la pantalla de dimensiones reducidas los elementos de la página web (tablas, márgenes, líneas, imágenes, etc.)
- Compatibilizar con los navegadores específicos de estos dispositivos, que normalmente tienen capacidades limitadas.

Además, estos dispositivos plantean los siguientes inconvenientes a la hora de su utilización:

- Requieren una conexión permanente a Internet.
- La interfaz web no siempre es adecuada y plantea problemas de usabilidad y eficiencia en muchos casos.

Como resultado y a pesar del esfuerzo que se ha realizado para el desarrollo de estas aplicaciones, sus limitaciones de uso han provocado que su utilización en el aula no sea tan atractiva como sería deseable.

3.2. Dispositivos móviles basados en iOS

El Objective C [5] es el lenguaje utilizado para el desarrollo de programas en dispositivos basados en el sistema operativo iOS de Apple ©.

Las ventajas que ofrecen estos dispositivos son:

1. Facilidad de manejo.
2. Aplicaciones intuitivas y optimizadas para la interfaz táctil.
3. Robustez de su sistema
4. Alta conectividad, disponibilidad de cámara, GPS y acelerómetros.

5. Precio razonable

Estas interesantes características han contribuido enormemente al éxito de estos dispositivos [4].

En el entorno universitario también se quiere unir al uso de estas tecnologías, tanto por la demanda de la sociedad actual que incrementa cada día su utilización con numerosas aplicaciones y usuarios, como por las ventajas que proporciona su utilización.

3.2.1. Proceso de desarrollo de aplicaciones

El desarrollo de una aplicación para iOS requiere de un entorno de programación (XCode) y un simulador que permite realizar pruebas sin necesidad de su instalación en un dispositivo compatible iOS. Posteriormente para la instalación de la aplicación en uno de estos dispositivos será necesario crear una cuenta como desarrollador Apple © y darse de alta en el programa de desarrollo iOS, siguiendo los pasos descritos en detalle en [7]. El coste del alta en este programa es de \$99 por año, aunque también existe un programa gratuito de desarrollo para el entorno universitario. Desafortunadamente este programa no permite la distribución de aplicaciones a través de la App Store.

Las aplicaciones iOS pueden implementarse en varios lenguajes de programación aunque el lenguaje oficial es Objective-C, un lenguaje compatible 100% con C pero con una extensión que permite programar con objetos. El API Cocoa Touch proporciona las clases necesarias para la definición de la interfaz de usuario táctil y el acceso a los recursos del dispositivo (conexión por red, cámara, GPS, etc.).

4. Propuesta de herramienta de evaluación continua con dispositivos móviles y táctiles

Durante las sesiones prácticas de las asignaturas de programación, los profesores realizan evaluaciones in situ, es decir, junto al puesto de trabajo del alumno examinando el trabajo de éstos y planteando cuestiones sobre las soluciones obtenidas. Una vez realizada la evaluación, el

profesor debe anotar la calificación obtenida y eventualmente comentarios que justifiquen dicha calificación. Esta anotación puede hacerse de distintas formas.

- En un papel, a la manera tradicional,
- Acercándose al equipo del profesor situado en otro puesto diferente al del alumno evaluado y anotar la calificación a través de la aplicación web (figura 2),
- Utilizando un dispositivo móvil que permita acceder a la aplicación web desde el puesto del alumno, pero con los inconvenientes que se han descrito en la sección 3.1.

Para dar una mejor solución a este proceso de evaluación in situ, hemos desarrollado una aplicación iOS denominada *StudentForms* que permite acceder a la base de datos de la aplicación de gestión de alumnos sin utilizar directamente la interfaz web (ver Figura 3). Cabe decir que el navegador Safari permite el acceso directo a la aplicación web, pero una aplicación a medida permite organizar, acceder y editar la información de una manera mucho más adecuada. Esta aplicación solo incluye parte de la funcionalidad de la aplicación web, concretamente la necesaria para la evaluación de los alumnos durante las sesiones prácticas. A continuación describimos detalladamente esta funcionalidad:

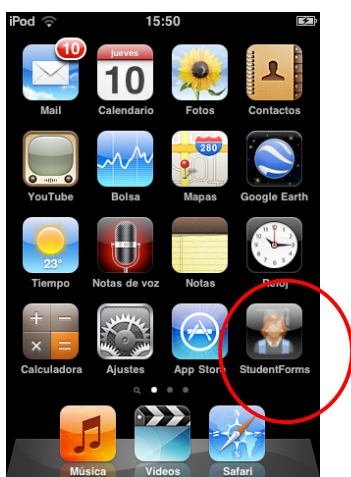


Figura 3. Aplicación de gestión de alumnos.

- El acceso está diseñado para que se realice únicamente desde el perfil del profesor, por tanto se requiere una única clave como puede verse en la figura 4.



Figura 4. Página de acceso a la aplicación.

- Una vez en la aplicación, se deberá seleccionar la asignatura cuyos datos quieren modificarse/consultarse (véase figura 5).



Figura 5. Listado de asignaturas

- Dentro de la asignatura aparece un listado con los alumnos ordenados alfabéticamente, como se muestra en la figura 6.

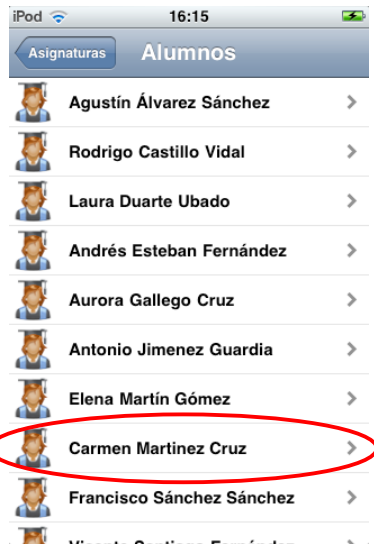


Figura 6. Listado de alumnos de la asignatura.

- Dentro de la ficha del alumno se tiene acceso a un formulario donde se puede consultar sus datos personales y modificar sus calificaciones prácticas, como puede verse en la figura 7. El resto de calificaciones, como la nota teórica o notas de trabajos alternativos, no son mostradas dado que el objetivo fundamental de la aplicación es evaluar al alumno en el aula de prácticas.

La calificación de un alumno puede realizarse simplemente tocando en la nota que se desea modificar y automáticamente aparece un teclado para introducir la nueva calificación, como queda ilustrado en la figura 8. Finalmente si el usuario desea salir de la ficha de un usuario deberá previamente confirmar o abandonar los cambios realizados

Todos los cambios realizados en las fichas son actualizados en tiempo real en la aplicación web, con lo que los alumnos pueden ver inmediatamente la valoración del trabajo realizado en prácticas.



Figura 7. Ficha de un alumno.

4.1. Arquitectura del sistema

En la figura 9 se muestra la arquitectura del sistema, donde puede verse la base de datos, los objetos de negocio que proveen de servicios a todo el sistema, los scripts PHP que implementan la aplicación web y los nuevos scripts que se han añadido para proveer de funcionalidad móvil al sistema. El acceso se realiza a través de un navegador web a un perfil concreto, o bien, usando del dispositivo móvil. Se utiliza XML como formato de intercambio de información entre el cliente móvil y los servicios web.

Como puede observarse en la figura 9, la aplicación está actualmente desarrollada para satisfacer las necesidades móviles desde el perfil del profesor. El acceso a la información por parte del alumnado únicamente será realizado a través de la aplicación web.



Figura 8. Modificación de una calificación.

5. Conclusiones y trabajos futuros

En este artículo se presenta una aplicación para la evaluación continua de alumnos en el aula de prácticas por parte del profesor, utilizando los dispositivos móviles táctiles más usados en la actualidad.

En resumen, esta aplicación evita muchos de los inconvenientes del acceso directo a aplicaciones web desde dispositivos móviles. La alternativa

planteada, pone a nuestra disposición una herramienta táctil sencilla, intuitiva y eficiente. Así, tras el uso de esta aplicación a lo largo de este curso se ha comprobado cómo se ha agilizado la tarea del profesorado de evaluación de las actividades del alumnado en el aula.

Además, aunque la herramienta descrita está siendo utilizada actualmente, también está en proceso de revisión y mejora para incluir nuevas funcionalidades, como la posibilidad de acceder y editar el resto de información de la ficha del alumno, aunque no sea esencial para la evaluación práctica, y la captura de la fotografía del alumno para la ficha.

Por último, sería deseable implementar una aplicación similar para los alumnos que dispongan de dispositivos compatibles. Finalmente sería también interesante disponer de versiones *Android* de las aplicaciones para el profesor y el alumno.

Agradecimientos

A la universidad de Jaén por el acceso a los medios técnicos para llevar esta implementación a cabo a través de su sistema de proyectos de innovación docente. A los proyectos de investigación de la Consejería de Innovación, Ciencia y Empresa de la Junta de Andalucía que han financiado parcialmente la realización de este trabajo:

- P07-TIC03175, “Representación y manipulación de objetos imperfectos en problemas de integración de datos: Una aplicación a los almacenes de objetos de aprendizaje”.
- P07-TIC-02773, “Gestión de Información Urbana Tridimensional”.

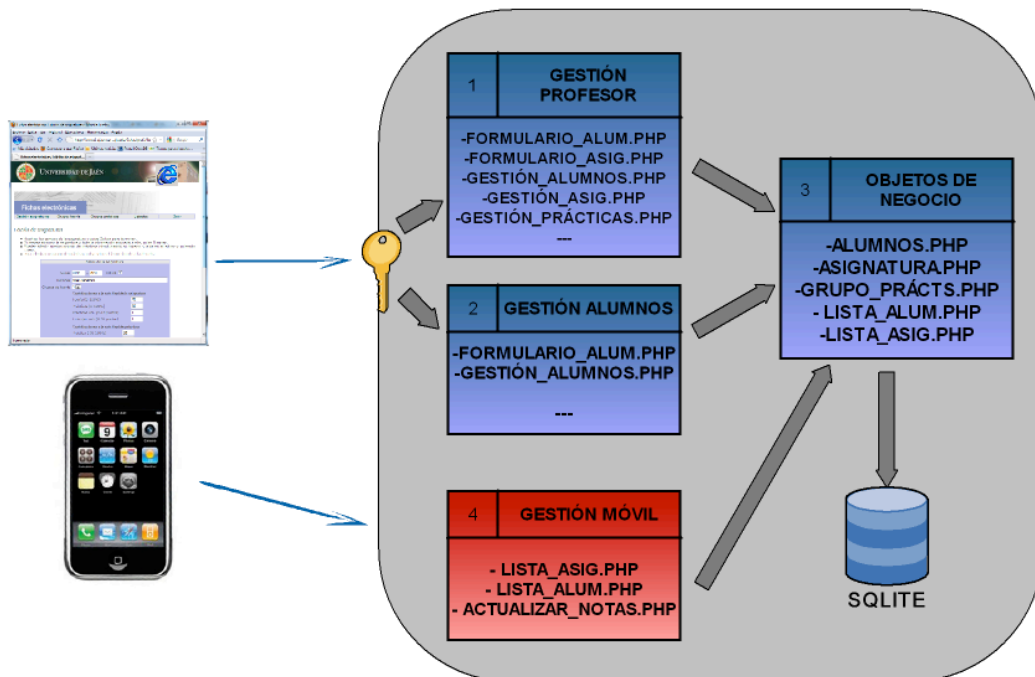


Figura 9. Arquitectura del sistema.

Referencias

- [1] José Sánchez Rodríguez . *Plataformas de enseñanza virtual para entornos educativos*. Pixel-Bit. Revista de medios y Educación, nº 34, pp. 217-233, 2009.
- [2] Blas Ogallar Fernández, Carlos Martínez Bazán. *Plataforma ILIAS como herramienta para docencia ETS. Utilización en la escuela politécnica superior de Jaén*. Jornadas de Trabajo sobre Experiencias Piloto de Implantación del Crédito Europeo en las Universidades Andaluzas Universidad de Cádiz. 2006. <http://www2.uca.es/orgobierno/rector/jornadas/documentos/127.pdf>
- [3] Jornadas de Innovación docente de la Universidad de Jaén. 2010. http://www.ujaen.es/serv/vicord/secretariado/seccino/comunicaciones_jornadas.htm
- [4] Datos económico de ventas de Apple©. <http://www.appleismo.com/apple-presento-los-resultados-financieros-del-tercer-trimestre-fiscal-del-2010/>
- [5] Objective C Programming Language. Apple©. <http://developer.apple.com/library/mac/#documentation/Cocoa/Conceptual/ObjectiveC/Introduction/introObjectiveC.html>
- [6] Guzmán Mancho Barés, María Dolores Porto Requejo y Carmen Valero Garcés. *Wikis e innovación docente*. Red U - Revista de Docencia Universitaria. Número monográfico IV. Número especial dedicado a Wiki y educación superior en España (en coedición con RED). http://www.um.es/ead/Red_U/m4/1-ManchoPortoValero.pdf
- [7] Deploying iPhone Apps to Real Devices. <http://mobiforge.com/developing/story/deploying-iphone-apps-real-devices>

Introducción a la Informática Gráfica 3D a través de un Taller de Realidad Virtual y Patrimonio Artístico

Laura Martínez
García

Departamento de
Ciencias Históricas y
Geografía
Universidad de
Burgos
C/ Villadiego s/n

09001 Burgos
lmg0002@alu.ubu.es

Andrés
Bustillo

Iglesias

Departamento de
Ingeniería Civil

Universidad de
Burgos
C/Francisco de
Vitoria s/n

09006 Burgos
abustillo@ubu.es

José Manuel
Sáiz Díez

Departamento de
Ingeniería Civil

Universidad de
Burgos
C/Francisco de
Vitoria s/n

09006 Burgos
jmsaiz@ubu.es

Resumen

Este trabajo presenta una experiencia educativa de introducción a alumnos de Educación Secundaria y primer curso del Grado de Comunicación Audiovisual en el proceso de generación de imágenes 3D por ordenador. La experiencia fue puesta en práctica durante la X Semana de la Ciencia de la Universidad de Burgos dentro del Taller “Realidad Virtual aplicada al Patrimonio Histórico”. Se pretendía que el alumno identificara las principales fases y conceptos propios del proceso de generación de imágenes 3D por ordenador. Para ello se utilizó un ejemplo práctico de herramientas de Modelado 3D a la reconstrucción de un edificio de interés Histórico-Artístico: la Cartuja de Miraflores de Burgos. Se permitía al alumno la modificación del modelo 3D, sus texturas e iluminación, así como su interacción con el modelo tridimensional en una Sala de Realidad Virtual. Las conclusiones de esta experiencia práctica se derivan del análisis de las encuestas realizadas a los alumnos participantes en la experiencia, donde se evalúan los puntos fuertes y débiles de esta metodología docente, además de la comprensión del proceso de recreación virtual por parte de los alumnos.

Summary

This paper presents an educational experience to introduce to secondary school students and first year of Audiovisual Communication Degree student in the process of 3D computer graphics generation. The experience was implemented during the tenth Science’s Week at the University of Burgos in a Workshop called “Virtual Reality applied to historical heritage”. It was intended that students identify the main stages and concepts of the process of 3D computer graphics generation. For this purpose, a practical example of 3D modeling tools for the reconstruction of a cultural heritage building: the Church of the Charterhouse of Miraflores in Burgos. The student was allowed to work on the 3D model, shaders and lighting, as well as to interactuate with the 3D model in a Virtual Reality Room. The results of this practical experience are extracted from the surveys filled in by the students. The survey evaluates the strengths and weaknesses of this teaching methodology, in addition to the level of understanding of the students about the process of virtual recreation.

Palabras clave

Realidad Virtual, Informática Gráfica, 3D, Educación, Patrimonio Histórico.

1. Introducción

La Informática Gráfica, y en especial su representación tridimensional, se ha convertido en una poderosa herramienta de ocio en nuestra sociedad. Desafortunadamente, el proceso de generación de imágenes 3D por ordenador resulta, en general, bastante desconocido para los estudiantes de enseñanzas secundarias y de muchas titulaciones universitarias, lo que limita el desarrollo de profesionales que puedan cubrir la oferta de empleo en este sector. A este desconocimiento se une la falta de desarrollo de metodologías específicas que permitan superarlo, existiendo muy pocos ejemplos de metodologías específicas para este reto tan actual [9,10]. Es importante señalar que estas experiencias se basan todas ellas en la integración del alumno en complejos entornos de trabajo 2D y 3D lo que sólo es posible si la experiencia educativa tiene una duración razonablemente extensa.

En determinadas orientaciones profesionales se está potenciando el desarrollo de metodologías docentes híbridas que incluyan herramientas tradicionales y nuevas tecnologías de modelado 3D y realidad virtual tales como la Arquitectura [7], la historia [2, 3] y la arqueología [8].

Dentro de este marco general se enmarca este trabajo cuyo objetivo principal es el de introducir a los estudiantes en el proceso de recreación virtual de un edificio histórico-artístico. Se pretende transmitir a los alumnos los conceptos fundamentales de las técnicas de modelado 3D por ordenador y de visualización 3D, además de permitirles interactuar tanto con un modelo real 3D, como con la recreación virtual final en tres dimensiones. Además, esta experiencia pretende demostrar que tal y como afirma García Ruíz [4], con esta tecnología los estudiantes "pueden aprender de manera más rápida y asimilar información de una manera más consistente que por medio del uso de herramientas de enseñanza tradicionales".

Esta experiencia educativa se ha articulado como un Taller específico dentro de la X Semana de la Ciencia, celebrada en Burgos entre el 12 y el 19 de noviembre de 2010. El objetivo de esta iniciativa es aumentar la cultura científica y el interés por la investigación desarrollada en la Universidad por parte de todos los ciudadanos, y en especial, por parte de los propios estudiantes

universitarios y alumnos de Bachillerato. El taller, denominado "Realidad Virtual aplicada al Patrimonio artístico", pretendía ser un acercamiento multidisciplinar a la aplicación de las nuevas tecnologías 3D a la difusión e interpretación del patrimonio histórico por lo que en su elaboración han participado ingenieros informáticos, comunicadores e historiadores del arte.

Para el desarrollo de este taller se disponía de una sala con estaciones de trabajo para la generación de modelos 3D por ordenador, una sala de visualización 3D semi-inmersiva y de un modelo tridimensional de la Cartuja de Miraflores de Burgos producido en trabajos anteriores [1]. El modelo 3D había sido desarrollado con el software de diseño 3D Blender [7]. El proceso de reconstrucción virtual 3D se ha desarrollado en cuatro fases diferenciadas: modelado 3D, texturizado, iluminación y postproducción. Esta metodología ya ha sido utilizada en otras recreaciones virtuales de edificios histórico-artísticos como la Catedral de Halle en Alemania [5] o la Torre de Hércules de La Coruña [6]. Una vez finalizado el modelo 3D, este se ha exportado a la sala de realidad virtual, donde se ha recreado un entorno semi-inmersivo.

En lo que resta de artículo se recoge, en primer lugar, los recursos hardware necesarios para la realización de esta experiencia práctica. Posteriormente se describe la experiencia realizada y se realiza una evaluación de la misma y de sus resultados a partir de los resultados de las encuestas cumplimentadas por los alumnos que participaron en esta experiencia. El artículo se cierra con la presentación de las conclusiones y las líneas futuras de trabajo.

2. Recursos disponibles

La Universidad de Burgos cuenta en la actualidad con un centro de innovación digital que tiene como objetivo fundamental la formación de profesionales en contenidos digitales. Este centro está compuesto por una sala de creación de contenidos multimedia, una sala de visualización 3D y un centro de cálculo.

En la sala de creación de contenidos multimedia se generan y publican los contenidos multimedia. Se trata de un aula de informática con

25 estaciones de trabajo equipadas con software de modelado y animación 3D.



Figura 1: Sala de creación de contenidos multimedia

El centro de cálculo está formado por una granja de renderizado y tres servidores de alto rendimiento: un servidor de almacenamiento, un servidor de streaming y un servidor web para la difusión de los contenidos en Internet.

En la sala de visualización se encuentra un sistema semi-inmersivo de realidad virtual 3D. Está formado por dos proyectores 3D, sonido de alta calidad, sistema de captura de movimiento y posicionamiento 3D y un equipo de gran capacidad de cálculo para la generación de los gráficos 3D.



Figura 2: Sala de Realidad Virtual

3. Descripción de la práctica

Esta experiencia práctica resulta compleja en cuanto a los objetivos a alcanzar, dada la multidisciplinariedad de los mismos. Este artículo recoge únicamente aquellos objetivos específicos de la enseñanza de Informática Gráfica, eludiendo entrar en los objetivos de aprendizaje histórico-artístico o de comunicación igualmente propuestos para este taller.

3.1. Objetivos específicos

Los alumnos deben de adquirir una serie de conocimientos específicos con la realización de esta experiencia práctica. Para alcanzarlos, esta experiencia plantea los siguientes objetivos específicos:

- Comprender las diferentes fases de la Reconstrucción Virtual de edificios históricos.
- Asimilar los conceptos de modelado 3D, texturizado, iluminación y renderizado.
- Comprender la importancia de la colaboración entre diferentes áreas y departamentos para la consecución de la Reconstrucción Virtual.
- Evaluar la aceptación del uso de la Realidad Virtual como herramienta pedagógica por parte de los alumnos.
- Fomentar el interés de los alumnos en otras disciplinas a través de la informática.

3.2. Destinatarios de la práctica

La práctica se ha dirigido a dos grupos de destinatarios:

1. 37 alumnos de primer curso del grado de Comunicación Audiovisual en la Universidad de Burgos. El grupo está formado por 16 hombres y 21 mujeres con edades comprendidas entre los 17 y los 28 años
2. 189 alumnos de de Bachillerato de Ciencias de diferentes institutos de Educación Secundaria Obligatoria de la provincia de Burgos. El grupo está formado por 85 hombres y 104 mujeres con edades comprendidas entre los 15 y los 21 años. De los 189 alumnos 148 se encuentran cursando 1º de Bachillerato y 46 2º de Bachillerato. Por otra parte, 148 alumnos provienen de centros de la capital de Burgos mientras que 41 proceden de localidades de la provincia de menor tamaño.

3.3. Fases de la práctica

Cada taller tiene una duración aproximada de 40 minutos y en él participan entre 15 y 20 alumnos. En este periodo de tiempo se desarrollan las cuatro fases en las que se divide esta experiencia educativa (Figura 3): presentación del proyecto de reconstrucción virtual del edificio a los alumnos, ejercicio práctico tutorizado, visita interactiva a la sala de Realidad Virtual y evaluación de la experiencia por parte de los estudiantes.

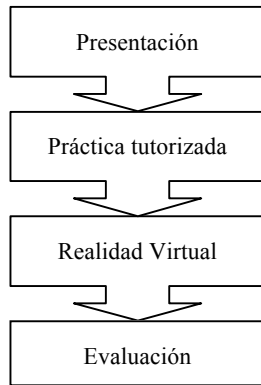


Figura 3: Fases del taller

En primer lugar, se realiza una introducción a la Reconstrucción Virtual del Patrimonio Artístico centrada en la reconstrucción virtual de la Iglesia de la Cartuja de Miraflores de Burgos, donde se explica la relación necesaria entre ingenieros informáticos, historiadores del arte y comunicadores audiovisuales para llevar a cabo esta reconstrucción.

En esta presentación se explican los conceptos básicos de modelado tridimensional, presentando como ejemplos algunos operadores de modelado 3D como extrusión, revolución, etc. Se explican las diferencias entre texturas procedurales y texturas de imagen y los diferentes tipos de luces que se pueden crear para ambientar una escena. También se define el concepto de renderizado, clave para la comprensión del proceso de reconstrucción virtual.

Una vez realizada la presentación se procede a la ejecución de una práctica tutorizada donde los alumnos pueden modificar el modelo original de la Cartuja de Miraflores aplicando las técnicas descritas en la presentación. En primer lugar, se presenta el interfaz de Blender, el movimiento por el espacio y la selección de vistas ortogonales y de perspectiva. Los alumnos deben modificar la textura de la vidriera de uno de los ventanales, aprendiendo de forma práctica como se aplica una textura de imagen a un modelo 3D, Figura 4.



Figura 4: Imagen del ventanal con la textura

Por otra parte, los alumnos deben modificar la iluminación de un elemento del modelo 3D (facistol) variando la intensidad y el color de las luces para ambientarlo de un modo más natural, Figura 5. En ambos casos los alumnos renderizan las imágenes antes y después de modificar los parámetros para comprobar los resultados de sus cambios.

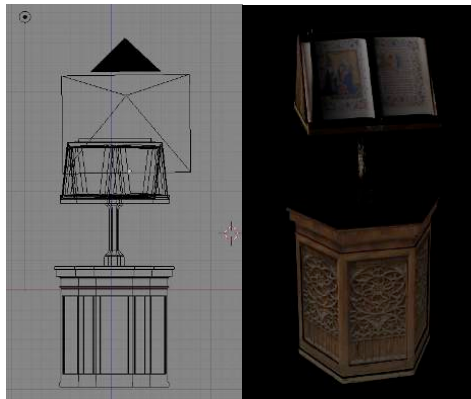


Figura 5: Imagen del facistol para la iluminación

La práctica tutorizada requiere la presencia de 2 docentes. Mientras uno explica utilizando una proyección los pasos a seguir, el otro puede resolver las dudas que les surjan a los alumnos, considerando el reducido tiempo para la realización de toda la experiencia práctica. Dado que el ritmo de adaptación al entorno 3D de los alumnos es muy variado, a los más avanzados se les da la posibilidad de moverse por las distintas capas que conforman el modelo 3D completo de la Cartuja de Miraflores.

Después de la práctica tutorizada los alumnos pasan a la sala de realidad virtual donde realizan una visita virtual inmersiva a la Cartuja de Miraflores. Primero se les explican los principales componentes que la conforman, centrándonos en el motor de movimiento y en los sistemas de captación de movimiento y visualización. A continuación se les explica el edificio desde el punto de vista artístico permitiéndoles la interacción con distintos elementos interactivos (videos, imágenes, textos, etc.) y se les presentan diferentes aplicaciones prácticas de la sala de realidad virtual como la generación de entornos virtuales industriales de simulación o juegos 3D.

La última fase de la experiencia consiste en la cumplimentación por parte de los alumnos de unas encuestas donde se valorará tanto la asimilación de los conceptos explicados durante el taller como el interés suscitado en los alumnos en la Reconstrucción Virtual.

4. Evaluación de la experiencia

4.1. Descripción del método de evaluación

La encuesta pretende valorar las diferentes etapas de la experiencia y los conocimientos adquiridos en cada una de las fases. Para ello la estructura es la misma que la del taller, de esta manera se analizarán también los puntos débiles y fuertes de cada una de las fases.

Está compuesta por 13 preguntas: 8 de ellas presentan 5 posibles respuestas de las cuales el alumno debe elegir 2, 4 preguntas son de tipo sí/no y una pregunta posibilita una valoración entre 1 y 10.

En primer lugar, se pregunta sobre los conceptos básicos explicados durante la presentación. Las cuestiones incluidas dentro de este grupo nos van a permitir conocer si los estudiantes han entendido para qué se realiza una reconstrucción virtual, cuales son las fuentes de información necesarias, por qué es necesario aplicar texturas sobre el modelo o para qué sirve renderizar.

En segundo lugar, se pide que valoren la dificultad del proceso de generación de imágenes 3D por ordenador y se les pregunta sobre su predisposición a realizar un modelo 3D por sí mismos.

A continuación se les pregunta por las ventajas y limitaciones que, desde su punto de vista, tiene la difusión de edificios histórico-artísticos a través de internet y de la sala de realidad virtual.

Por último, se les pregunta si han visitado y si estarían dispuestos a visitar el edificio “in situ” y si están más interesados tras el taller en el modelado 3D.

4.2. Resultados

En total se han recogido 226 encuestas válidamente cumplimentadas. Las respuestas han sido normalizadas al número de encuestas válidas para su análisis y representación.

En las primeras preguntas de la encuesta los alumnos responden a los conceptos básicos explicados durante el transcurso del taller. En general, tanto los alumnos universitarios como los de bachillerato comprenden el proceso de recreación virtual asimilando los conceptos teóricos y valorando la importancia de cada una de las etapas que forman el proceso.

Cuando se les pregunta por la dificultad que creen que tiene el proceso de modelado 3D todos los alumnos consideran que es un proceso complejo. En una valoración de 1 a 10 los alumnos de Comunicación Audiovisual muestran un valor promedio de 8 frente a un valor promedio de 7 de los alumnos de Bachillerato.

Si lo analizamos desde el punto de vista del género las mujeres, estas ven el proceso de reconstrucción virtual como más complejo que los hombres con una diferencia de medio punto, Figura 6. En ambos casos la desviación estándar no resulta significativa (barras de error en la Figura 6).

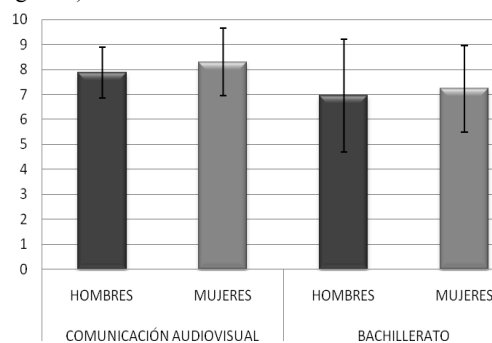


Figura 6: Dificultad del proceso de modelado

Cuando se les pregunta si estarían dispuestos a realizar un modelado 3D ellos mismos la mayoría responden que sí. Es interesante señalar que, aunque los alumnos de Comunicación Audiovisual consideran más complejo el proceso de creación de imágenes tridimensionales, están dispuestos a realizarlo en mayor medida que los alumnos de Bachillerato que lo consideraban más sencillo. Aunque en general, como se puede comprobar en la Figura 7, la mayoría de los alumnos estarían dispuestos a realizarlo.

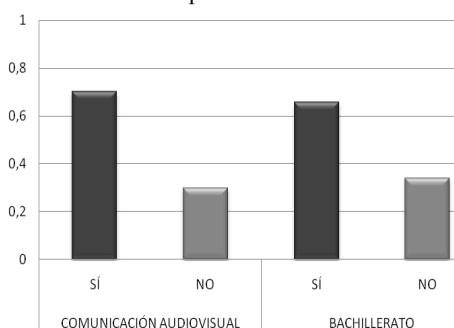


Figura 7: Predisposición de los alumnos a realizar un modelado 3D.

Cabe destacar, que dentro de la predisposición de los alumnos a realizar el modelado 3D encontramos una gran diferencia por sexos, Figura 8. Los hombres están más dispuestos que las mujeres a realizarlo cuando esta diferencia no se aprecia a la hora de valorar la dificultad del proceso de modelado por ordenador.

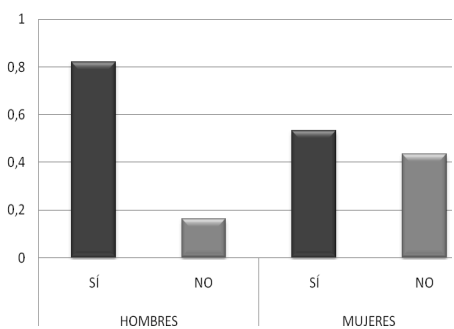


Figura 8: Predisposición de los alumnos a realizar un modelado 3D por sexo

Una vez comprobado que los alumnos han comprendido el proceso de generación de

imágenes 3D, se les pregunta por las ventajas y limitaciones que desde su punto de vista tiene la comunicación de estos elementos a través de Internet y de la Sala de Realidad Virtual. Tanto los alumnos de Bachillerato como los de Comunicación Audiovisual demuestran comprender claramente los puntos fuertes y débiles.

Todos los alumnos están más interesados después del taller en la generación de imágenes 3D por ordenador y en la Realidad Virtual, Figura 9. En cuanto al género los hombres se interesan más que las mujeres aunque la diferencia no es tan grande como la que podíamos observar en la predisposición a realizar por sí mismas el proceso de modelado 3D.

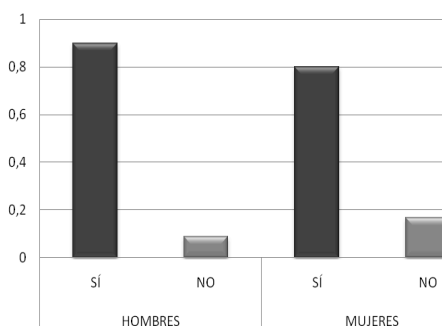


Figura 9: Interés por el modelado 3D.

En las encuestas se aprecia cómo también aumenta el interés de los alumnos en el propio edificio histórico y cómo, tras realizar el taller, se encuentran más dispuestos a visitarlo. Se puede concluir por tanto que la Informática Gráfica se puede convertir en una herramienta útil para fomentar el interés en otras disciplinas del conocimiento.

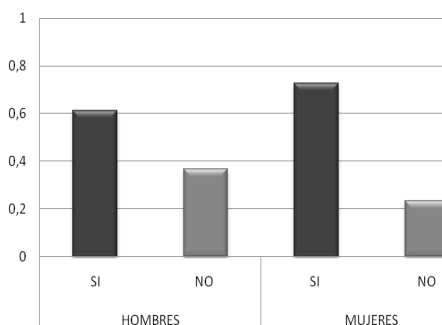


Figura 10: Interés por el monumento

Los resultados de las encuestas nos demuestran que la metodología empleada a la hora de realizar el taller ha sido efectiva. En 40 minutos los alumnos han asimilado conceptos básicos relacionados con la generación de imágenes 3D y comprenden las ventajas y limitaciones del uso de la Realidad Virtual, conceptos fundamentales tanto de la Informática Gráfica como de algunas de sus aplicaciones más prometedoras.

También han comprendido la importancia del trabajo entre los diferentes departamentos de la Universidad a la hora de realizar una investigación.

5. Conclusiones

Este trabajo presenta una experiencia educativa de introducción a conceptos fundamentales de la Informática Gráfica y de algunas de sus aplicaciones más prometedoras tanto de tipo industrial como en la docencia multidisciplinar. La experiencia es de una duración muy limitada, 40 minutos, y pretende mostrar como, aún en un tiempo tan limitado, la implicación práctica del alumno en la experiencia y la utilización de un ejemplo real cercano al alumno pueden permitir alcanzar objetivos complejos. Esta experiencia fue puesta en práctica durante la X Semana de la Ciencia de la Universidad de Burgos dentro del Taller "Realidad Virtual aplicada al Patrimonio Histórico". El Modelo 3D utilizado es una reconstrucción de un edificio de interés Histórico-Artístico: la Cartuja de Miraflores de Burgos.

Las conclusiones de esta experiencia práctica se derivan del análisis de las encuestas realizadas a los alumnos participantes en la experiencia, donde se evalúan los puntos fuertes y débiles de esta metodología docente, además de la comprensión del proceso de recreación virtual por parte de los alumnos. Se puede concluir que la metodología empleada a la hora de realizar el taller ha sido efectiva en la comprensión de conceptos y en el análisis de ventajas e inconvenientes de la utilización tanto del modelado 3D como de la Realidad Virtual. Además, la experiencia aumenta el interés de los alumnos en el mundo de la Informática Gráfica y de la Realidad Virtual.

Entre las líneas futuras que esta experiencia plantea se encuentra generar metodologías

similares para alumnos de otras titulaciones universitarias que pudieran encontrar interés en el Modelado y la Animación 3D, dos grandes nichos de empleo en los próximos años en España. Además, se pretende repetir la experiencia a través de un taller de mayor duración en el que el alumno pueda aplicar técnicas avanzadas de Modelado 3D y que introduzca también conceptos y técnicas básicas de Animación por computador.

Agradecimientos

Esta investigación ha sido parcialmente financiada por el Programa de Español "Impulso de la Industria de Contenidos Digitales desde las Universidades" del Ministerio de Industria, Turismo y Comercio y el "Aula para La Creación y Difusión de Contenidos Multimedia" de la Universidad de Burgos.

Referencias

- [1] Bustillo, A., Martínez, L., Alaguero, M., Iglesias, L. S., "The Church of the Charterhouse of Miraflores in Burgos: Virtual reconstruction of an artistic imaginary", *Abstracts of the XXXVIII Conference on Computer Applications and Quantitative Methods in Archaeology*, pp: 425-428, Granada 2010
- [2] De Paolis L.T., Aloisio G., Celentano M.G., Oliva L., Vecchio P., "A Game-Based 3D Simulation of Otranto in the Middle Ages", *Proceedings of the 3rd International Conference on Advances in Computer-Human Interactions*, pp:130-133, Saint Maarten 2010
- [3] Di Blas N., Poggi C., "Investigating entertainment and learning in a multi-user 3D virtual environment", *Human-Computer Interaction Symposium. International Federation for Information Processing*, vol: 272, pp: 175-188, Milano 2008
- [4] García Ruiz, M. A., "Panorama General de las Aplicaciones de la Realidad Virtual en la Educación". *Revista Educación 2001*, nº 43, pp:37-40, 1998
- [5] König S., "Realización de la Catedral", *Blender Art Magazine* N°6, pp: 25-27, 2006, www.blenderart.org

- [6] Noya, R., Otero, A., Goy, A., Flores, J., “The Tower of Hercules: A walk through time and space”, *Abstracts of the XXXVIII Conference on Computer Applications and Quantitative Methods in Archaeology*, pp: 409-412, Granada 2010
- [7] Roussou M., Drettakis G., Tsingos N., Reche A., Gallo E., “A user-centered approach on combining realism and interactivity in virtual environments”, *Proceedings of the Virtual Reality Annual International Symposium*, pp: 251-252, Chicago 2004
- [8] Styliadis, A. D., Konstantinidou, D. G., Tyxola, K. A., “eCAD system design - Applications in architecture”, *International Journal Of Computers Communications & Control* , vol: 3, pp: 204-214, 2008
- [9] Van Langeveld M., Kessler R., “Digital Visualization Tools Improve Teaching 3D Character Modeling”, *Proceedings of the 41st Acm Technical Symposium on Computer Science Education*, pp: 82-86, New York 2010
- [10] Wang J., Tian F., Seah H.S., “Sketch-up in the Virtual World”, *Proceedings of the 2008 International Conference On Cyberworlds*, pp: 109-116, Hanzhou 2008

Sesión 7B:
Tecnologías de la información
en la gestión empresarial
Compromiso social y medioambiental

Docencia de Arquitectura Orientada a Servicios

P. García-Sánchez, J. González, P. Castillo, M.G. Arenas
Departamento de Arquitectura y Tecnología de los Computadores
Universidad de Granada
ETS. Informática y Telecomunicación, C/Periodista Daniel Saucedo s/n
18071 Granada
{pgarcia,jesus,pedro,maribel}@atc.ugr.es

M. A. López
Fundación I+D del Software Libre
BIC Granada-CEEI
Avenida de la Innovación, 1
18100 Granada
malopez@fidesol.org

Resumen

Este trabajo presenta los contenidos del curso “Web 2.0: Arquitectura Orientada a Servicios en Java” de la Escuela de Posgrado de la Universidad de Granada. El objetivo del curso es familiarizar al alumno con la programación de Servicios Web. Dada la gran variedad de técnicas disponibles para utilizar Arquitectura Orientada a Servicios, se presentan los siguientes temas: utilización de protocolos bien definidos para comunicación y contrato (SOAP y WSDL), creación de Web Services con JAX-WS y orquestación de Servicios Web con BPEL. Al final del curso, el alumno será capaz de crear, utilizar y mantener Servicios Web para el desarrollo de aplicaciones interempresariales, utilizando servicios creados o ya disponibles en la web, así como la orquestación lógica de los mismos.

Summary

This work presents the contents of the course “Web 2.0: Service Oriented Architecture on Java” from the Graduate School of the University of Granada. The course objective is to familiarize students with Web Services programming. Due to the wide variety of available technologies, several subjects are presented: the usage of well-defined protocols to contract and communication (SOAP and WSDL), web services creation using JAX-WS, and service orchestration with BPEL. At the end of the course, students will be capable to create, use and manage Web Services for business applications, using new or available services in the web, and also their logical orchestration.

Palabras clave

Arquitectura Orientada a Servicios, Web Services, Java, BPEL, Curso

1. Introducción

Los problemas más comunes en el desarrollo del software suelen ser la incompatibilidad entre aplicaciones, modelos de datos, lenguajes de programación y sistemas de comunicación, lo que obliga a rediseñar todas las aplicaciones y reescribirlas para que operen entre sí. Por lo tanto una aplicación que desee crecer en un futuro debería obviar características restrictivas y partir de un buen diseño que permita la extensibilidad y la comunicación con el mayor nivel de abstracción posible. De esta idea surge la Arquitectura Orientada a Servicios (SOA) [5, 9], ya que se hace necesaria una forma de comunicación eficiente y escalable, independiente del lenguaje de programación y plataforma de cada una de las aplicaciones que deseen intercomunicarse. Los elementos básicos que conforman SOA son:

- Proveedores de servicios: Una aplicación expone operaciones que cualquier otra puede usar
- Consumidores de servicios: Utilizan las operaciones de los proveedores para obtener información
- Bus de servicios empresariales: para integrar los servicios de forma lógica y ampliable.

Estos servicios, llamados Servicios Web (Web Services), son un sistema software diseñado para soportar interacción Máquina a Máquina sobre una red, es decir, son interfaces (APIs) que pueden ser accedidas remotamente. Utilizan protocolos para comunicación bien definidos, como SOAP y sus interfaces se publican utilizando el formato WSDL (donde se indican las operaciones y tipos de dato que se pueden intercambiar), siendo su implementación realizada cualquier lenguaje. Esto permite por ejemplo que una aplicación escrita en Java reciba información generada por otra aplicación realizada con C++, PHP o cualquier otro lenguaje de programación.

Debido al auge de estas tecnologías en el mundo empresarial se proyectó la realización del curso

“Web 2.0: Arquitectura Orientada a Servicios en Java”, organizado por la Escuela de Posgrado de la Universidad de Granada, para familiarizar a los estudiantes de carreras técnicas (sobre todo los de Ingeniería en Informática o Telecomunicación)¹ en estas tecnologías, ya que no están presentes en el plan de estudios de la Universidad de Granada. Sin embargo, es tal la adaptación de la Arquitectura Orientada a Servicios que otras universidades, como la Universidad Autónoma de Barcelona, ya cuentan con un plan docente basado este paradigma, el Grado en Informática y Servicios [2].

El resto del artículo se estructura como sigue: inicialmente se introduce el concepto de Arquitectura Orientada a Servicios. A continuación, en la Sección 3, se presentan los contenidos del curso y las tecnologías utilizadas (Java, XML, PHP, JAX-WS y BPEL), para finalmente mostrar la recepción del curso por parte de los alumnos y las conclusiones a este trabajo.

2. Introducción a los Servicios Web

Actualmente las Arquitecturas Orientadas a Servicios (*Service Oriented Architecture*, SOA) están en auge, debido a los beneficios que proporcionan a la hora de desarrollar e integrar aplicaciones distribuidas o modulares. El principal concepto de SOA es el de *servicio*. Podemos ver un servicio como una llamada a una función, que se ejecutará local o remotamente, y que es independiente del lenguaje de programación y plataforma en la que se ejecuta. Este servicio consta de una interfaz bien definida y que depende de la tecnología que se desea utilizar para implementar SOA.

El resto de elementos básicos que conforman SOA, y cuyas relaciones pueden verse en la Figura 1, son los Proveedores, Consumidores y Publicadores de Servicios. El *Proveedor de Servicios* es un ente (nodo, clase, programa, etc.) que brinda un servicio en respuesta a una llamada o petición desde un *Consumidor de Servicios*. Éste utiliza el *Publicador de Servicios* para obtener información sobre los servicios que estén disponibles para su uso y sobre las interfaces (*Descripción del servicio*) para invocarlos.

¹ Cualquiera se puede apuntar, pero por los conocimientos previos que se recomiendan, está orientado especialmente a alumnos de esas dos titulaciones.

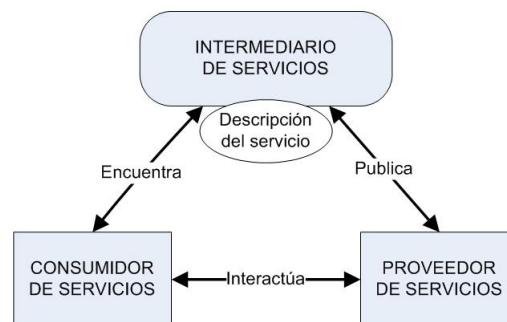


Figura 1: Esquema de interacción de servicios. El proveedor de servicios publica una descripción del servicio que es utilizada por el consumidor para encontrar y usar servicios.

A la hora de desarrollar sistemas software se hace necesario que sean compatible con las implementaciones SOA más extendidas, como por ejemplo los *Servicios Web (Web services)* [9]. Su arquitectura está diseñada para soportar interacción máquina a máquina sobre una red, utilizando sobre todo el protocolo SOAP (*Simple Object Access Protocol*) [13] para transmitir mensajes entre los diferentes computadores. Las interfaces de los servicios están descritas en WSDL (*Web Service Description Language*) [12], un lenguaje basado en XML que proporciona un modelo para describir Servicios Web y la manera de comunicarse utilizando éstos (equivale a *Descripción del servicio* en la Figura 1). Estos servicios pueden ser listados usando UDDI (*Universal Description, Discovery, and Integration*) [7], un registro basado en XML independiente de la plataforma (*Intermediario de servicios* en la Figura 1).

En un ambiente SOA, los nodos de la red suelen poner disponibles sus recursos a otros participantes en la red como servicios independientes, a los que tienen acceso de un modo estandarizado. La mayoría de las definiciones de SOA identifican la utilización de Servicios Web (empleando SOAP y WSDL) en su implementación, pero no obstante se puede implementar SOA utilizando cualquier tecnología basada en servicios, como por ejemplo, OSGi [8].

3. Temario

Esta sección presenta los contenidos teóricos y prácticos del curso. El curso dura 50 horas, divididas en cinco bloques: Introducción a los Servicios Web y Java (10 horas), XML (5 horas), Creación de servicios web en Java utilizando Jax-WS (15 horas), PHP (5 horas), Orquestación de Servicios Web con BPEL (10 horas) y finalmente un día para presentar otras arquitecturas SOA y examen (5 horas).

3.1. Introducción a los Web Services y a Java

Como introducción al curso se le presenta al alumno de forma sucinta el funcionamiento de los Servicios Web, para luego profundizar en una introducción al lenguaje Java, ya que será el que se utilizará durante el curso. En esta sección se explican las peculiaridades del lenguaje y se realizan distintos ejercicios simples para comprender el funcionamiento y uso. Asimismo se explica el uso de NetBeans ², entorno de desarrollo libre usado para los ejercicios en el curso y algunas de las funcionalidades básicas (compilación, depuración, creación de proyectos...).

3.2. XML

En la segunda parte del curso, se les muestra a los alumnos en qué consiste XML (Extensible Markup Language) junto con herramientas para su manejo y ejemplos prácticos variados de utilización en diversos ámbitos de la programación web.

El orden del temario que se incluye en esta parte es el siguiente:

- **Introducción:** Simplemente se presenta al docente, sus datos, datos de donde encontrar la documentación necesaria para esta parte y se sitúa esta parte del curso dentro de la totalidad del temario.
- **XML:** Donde se introduce en qué consiste XML, su utilidad, ventajas frente a otros lenguajes de marcas y cada uno de las partes que lo componen.
- **Protocolos XML:** Donde se mencionan varios de los protocolos de comunicación más utilizados en la actualidad y que están basados en XML, como son: XML-RPC, SOAP y RSS.

²<http://www.netbeans.org>

- **XML y Java:** Java es la herramienta que se utiliza en todas las partes del curso para programar, por lo que en esta parte se introduce lo que ofrece Java para tratar documentos XML. Se mencionan aspectos básicos para el tratamiento de la información como el análisis de la formalidad de los documentos, la aplicación de “plantillas” a los documentos para comprobar si están correctamente formados, etc. También es en esta parte donde se mencionan la gran cantidad de herramientas que Java proporciona para tratar XML. Y concretamente, se detalla el funcionamiento básico de dos de ellas en los siguientes apartados del curso.
- **SAX:** Es una de las APIs disponibles en Java para el tratamiento de documentos XML. Durante el curso se ve la estructura general de SAX, así como diversos ejemplos de utilización que comienzan siendo sencillos para ir complicándose a lo largo del temario.
- **DOM:** Es la segunda API que se muestra a los alumnos puesto que el funcionamiento es bastante diferente a la anterior, dando así dos enfoques totalmente distintos a los asistentes al curso de cómo tratar documentos XML con Java.

A lo largo de todo el temario se le propone a los alumnos la realización de numerosos ejemplos relacionados con la materia que se está introduciendo y son ellos los que los realizan aunque al final del curso se les proporciona la dirección web donde pueden encontrarlos ya resueltos.

Entre estos ejercicios se encuentra realizar un programa Java que se conecte a un servidor remoto donde se encuentran disponibles varios servicios web ya programados y en funcionamiento. El alumno puede aprender como realizar con Java una conexión remota de estas características, cómo acceder a los servicios web disponibles y cómo obtener la respuesta de estos servicios a la petición que ellos formulan.

Para esta propuesta se utiliza una herramienta adicional denominada SoapUI (<http://www.soapui.org/>). Se trata de una herramienta de software libre que nos permite testear el funcionamiento de servicios webs y que los alumnos comienzan a utilizar en esta parte del curso y continúan en las partes siguientes.

3.3. Desarrollo de Servicios Web con Jax-WS

Sabiendo que la teoría sobre Java, XML y los Servicios Web ya se han impartido en los primeros capítulos del curso, este apartado se centra estructurar conocimientos que permitan relacionar conceptos cuando estemos desarrollando servicios web.

Para realizar un desarrollo de servicios web haciendo uso de Netbeans es necesario explicar nuevos conceptos de la herramienta Netbeans (ya que se usarán nuevas funcionalidades). Además, se explica el uso del servidor de aplicaciones web donde nuestros servicios se desplegarán para su ejecución, que en este caso es el servidor Glassfish³. El alumno aprende conceptos básicos del servidor como son: arrancar, parar, cambiar a modo de depuración, desplegar nuevos proyectos y sincronizarlo con el Netbeans.

El temario comienza con la explicación del desarrollo de un servicio web sin utilizar la ayuda de un entorno de desarrollo moderno. En este tema se introduce al alumno en el desarrollo de un WSDL (xml de descripción de un servicio web) escribiendo cada uno de los elementos necesarios para que el sistema pueda comprender y levantar un servicio web. Es decir, a partir de un WSDL se crea automáticamente el código a rellenar con su comportamiento. También se explica el paso contrario: a partir de una clase Java se genera automáticamente el WSDL que la representa.

El siguiente tema presenta la creación de clientes para web services, introduciendo al alumno en las técnicas de programación necesarias para la invocación de servicios web desde cualquier programa Java (una aplicación de escritorio, una aplicación web u otro servicio web). Durante la explicación de este apartado el alumno puede crear clientes para todos los ejemplos realizados en el curso y comparar los resultados por los ofrecidos por un cliente genérico como es SoapUI.

También se profundiza en la creación de Servicios Web y clientes más completos, con estructuras de datos reales y más complejas.

Como conclusión de esta sección se introduce al alumno en el desarrollo de una aplicación final haciendo uso de servicios web desarrollados por ellos mismo que realizan operaciones simples. Como ejemplos de proyecto se propone el desarrollo

de interacciones de personajes de una conocida serie de televisión, utilizando objetos complejos, como Usuarios, Listas de Usuarios y otros objetos compuestos.

3.4. PHP

PHP es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas.

Es un lenguaje interpretado de propósito general ampliamente usado y diseñado especialmente para desarrollo web que puede ser incrustado dentro de código HTML. Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. Veamos un ejemplo sencillo:

```
<html>
<body>
  <?php
      echo ‘‘Esto es un script PHP’’;
  ?>
</body>
</html>
```

Cuando el cliente hace una petición al servidor para que le envíe una página web, el servidor ejecuta el intérprete de PHP. Éste procesa el script solicitado, que generará el contenido de manera dinámica (por ejemplo obteniendo información de una base de datos). El resultado es enviado por el intérprete al servidor, quien a su vez se lo envía al cliente. Así, al ejecutar el script anterior, el cliente recibirá sólo los resultados de la ejecución por lo que es imposible para el cliente acceder al código que generó la página.

A lo largo del curso se muestran diversos ejemplos, partiendo de algunos muy sencillos para estudiar la sintaxis básica, definición de tipos de datos y uso de estructuras de control. Más adelante se estudian conceptos y ejemplos avanzados, tales como orientación a objetos, tratamiento de formularios web, despliegue y uso de servicios web, y por último, acceso a bases de datos MySQL desde PHP. Con esta sección el alumno comprende que los Servicios Web son independientes del lenguaje, pudien-

³<http://www.glassfish.org>

do llamar a servicios hechos en Java desde programas PHP.

3.5. Orquestación de Servicios Web con BPEL

BPEL (*Business Process Execution Language*, Lenguaje de Ejecución de Procesos de Negocio) es un lenguaje basado en XML diseñado para el control centralizado de la invocación de diferentes servicios Web, con cierta lógica de negocio añadida que ayudan a la programación en gran escala (programming in the large).

Su objetivo es definir procesos de negocio que interactúen con entidades externas mediante operaciones de un servicio Web definidas usando WSDL y que se manifiestan a si mismas como un servicio Web. Para ello utiliza un lenguaje basado en XML. BPEL sirve para orquestar servicios web, ya que consume servicios existentes para crear nuevos servicios de grano grueso. Un ejemplo de un proceso BPEL puede verse en la Figura 2. Al estar definido en XML existen muchos programas que permiten desarrollar procesos BPEL de forma gráfica, como por ejemplo NetBeans.

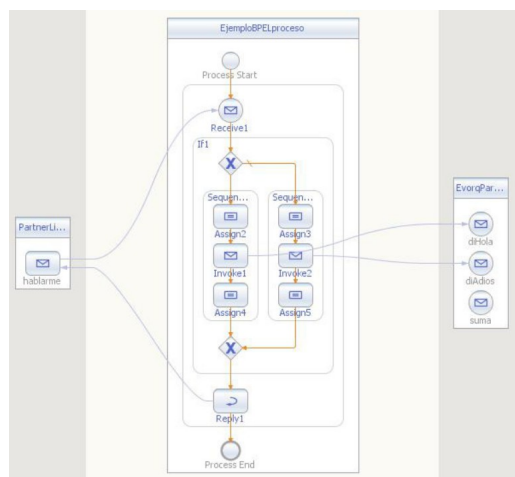


Figura 2: Ejemplo de proceso BPEL realizado en NetBeans. Este proceso llama a las operaciones diHola o diAdios dependiendo de un booleano, y devuelve un saludo

En la parte teórica de este curso se explica también el concepto de ESB (*Enterprise Service Bus*). Un ESB es un contenedor de aplicaciones en el que

cualquier aplicación que entienda XML puede ser emisora/receptora de mensajes dentro del bus. Permite una enorme flexibilidad a la hora de diseñar la arquitectura de nuestro sistema, ya que se puede utilizar desde una arquitectura cliente/servidor hasta una arquitectura orientada a eventos. Los componentes son “enchufables”, esto da la flexibilidad de añadir funcionalidad a un sistema en producción sin tener que pararlo y sin comprometer el funcionamiento de otros componentes del sistema. Por ejemplo, un componentes básico para ESB es un motor que ejecuta BPEL. Un ejemplo de ESB libre es OpenESB [11], incluido dentro del servidor Glassfish de Sun y utilizado en el curso.

En esta parte del curso se explican los distintos conceptos del lenguaje BPEL y se propone al alumno realizar varios ejercicios utilizando los servicios web creados en secciones anteriores del curso. Para ello el alumno utilizará el programa NetBeans para desarrollar procesos BPEL y desplegarlos dentro de OpenESB. Los primeros ejercicios son iterativos, los alumnos van añadiendo nuevos servicios simples ya desplegados en el servidor del curso a su proyecto BPEL para ir aprendiendo el uso de todas las actividades que se usan en este lenguaje (*for, if, invoke, reply...*). Una vez comprendidos estos conceptos, se les presenta un proyecto más completo en el que tienen que utilizar otros servicios desplegados más complejos para crear distintos procesos BPEL que los orquesten.

3.6. Otras arquitecturas y metodologías SOA

Como cierre al curso se presentan de forma teórica otras implementaciones de SOA y metodologías de desarrollo. Entre las arquitecturas orientadas a servicio más extendidas está OSGi [8], que es una SOA para máquinas virtuales en Java, o ebXML [10], orientada sobre todo al intercambio de datos empresariales. Además, se presentan algunas metodologías para desarrollar sistemas basados en SOA, como BCM [6] o SOMA [1].

Finalmente se presentan brevemente algunos proyectos basados en SOA en los que los docentes del curso han trabajado tanto en el ámbito académico como el empresarial, como por ejemplo eIntegra [5], GAD, AmIVital⁴ [3] u OSGiLiath [4].

⁴<http://www.amivital.es>

4. Respuesta de los alumnos y Conclusiones

El curso ha sido impartido en tres ediciones en la ETS. de Ingenierías en Informática y Telecomunicación de la Universidad de Granada. En todas ellas se ha cubierto el cupo de 27 alumnos. Al final de cada edición se pasó un cuestionario para evaluar el curso, siendo las notas 4,08, 4,5 y 4,8 (sobre 5) en las tres ediciones respectivamente. Además, se propuso a los alumnos qué mejoras habrían de añadirse en ediciones posteriores, siendo aplicadas en tales ediciones, como por ejemplo el aumento de la parte de prácticas respecto a teoría, y orientando la enseñanza a la realización de ejercicios iterativos.

Todo el software utilizado en el curso es libre, con lo que el coste para el alumno es nulo. Toda la información, software y ejemplos del curso se encuentran en la web <http://evorq.ugr.es/cursows>, a disposición de los alumnos y el resto de personas interesadas en la temática.

Agradecimientos

Trabajo financiado por proyecto CEI BioTIC GENIL (CEB09-0010), Programa CEI del MICINN (PYR-2010-13) y beca FPU AP2009-2942.

Referencias

- [1] Arsanjani, A. Ghosh, S. Allam, A., Abdollah, T., Ganapathy, S., y Holley, K. SOMA: A method for developing service-oriented solutions. *IBM Systems Journal*, 47(3):377–396, 2008.
- [2] Boixader, F., Guardiola, J., Albert, M., y Ribas, J. El Grado en Informática y Servicios. Una respuesta a la nueva demanda del contexto social. *XVI Jornadas de Enseñanza Universitaria de la Informática*, pages 129–135, 2010.
- [3] García-Sánchez, P., González, J., Castillo, P. y Prieto, A. Using UN/CEFACT'S Modelling Methodology (UMM) in e-Health Projects. *Bio-Inspired Systems: Computational and Ambient Intelligence*, pages 925–932, 2009.
- [4] García-Sánchez, P., González, J., Castillo, P., Merelo, J., Mora, A., Laredo, J. y Arenas, M. A Distributed Service Oriented Framework for Metaheuristics Using a Public Standard, Nature Inspired Cooperative Strategies for Optimization (NICSO 2010), pages=211–222, 2010.
- [5] García-Sánchez, P., Merelo, J.J., Castillo, P., Sevilla, J.P., Martín, M. y López, M. Plataforma de integración de servicios para la administración basada en BPEL y SOA. En *Actas de las III Jornadas en Servicios Web y SOA (JS-WEB 2007)*, pages 111–118, 2007.
- [6] OASIS BCM TC. Business-Centric Methodology for Enterprise Agility and Interoperability. Executive White Paper. 2003. <http://businesscentricmethodology.com/>.
- [7] Oasis Open. UDDI Specification. 2006. Disponible en: <http://www.uddi.org/specification.html>.
- [8] OSGi Alliance. OSGi service platform release 4.2. 2010. Disponible en: <http://www.osgi.org/Release4/Download>.
- [9] Papazoglou, M. P., y Van Den Heuvel. W. Service oriented architectures: Approaches, technologies and research issues. *VLDB Journal*, 16(3):389–415, 2007.
- [10] Patil, S. y Newcomer. E. ebXML and Web Services. *IEEE Internet Computing*, 7(3):74–82, 2003.
- [11] Salter, D. y Jennings, F. Building SOA-Based Composite Applications Using NetBeans IDE 6. *Birmingham-Mumbai: Packt Publishing*, 2008.
- [12] World Wide Web Consortium. Web Services Description Language (WSDL) 1.1. 2001. Disponible en: <http://www.w3.org/TR/wsdl>.
- [13] World Wide Web Consortium. SOAP Version 1.2 Part 1: Messaging Framework (Second Edition), 2007.

Informática en la edad de oro

Rosa María Gil Iranzo

Departament d'Informàtica i Enginyeria Industrial
Universitat de Lleida
C./ Jaume II, 69
25001 Lleida
rgil@diei.udl.cat

Resumen

Los estudios superiores para nuestros mayores se están consolidando en nuestra geografía. El acceso a la universidad a este colectivo ha supuesto un reto pedagógico difícil de afrontar a los docentes universitarios. Esto es debido a dos factores principalmente: la heterogeneidad en el perfil del alumnado respecto a sus conocimientos y por otra parte la excesiva auto-exigencia del mismo. En este artículo veremos las conclusiones después de cuatro años de la impartición de dos asignaturas de informática centradas en informática gráfica y multimedia además de ofimática que pretenden ser una guía para aquellos que emprendan el reto de incorporarse a estos nuevos grados en el ámbito informático.

Summary

Higher education for old people is being consolidated in our country. The university entrance for this group has been a difficult educational challenge. University teachers have to face two challenges: knowledge heterogeneity in the profile of students and an extremely demand themselves. In this article we look at the findings after four years of teaching two computer courses focused on multimedia and office automation. The results are intended to be a guide for those undertaking the challenge to move into these new grades.

Palabras clave

Informática, sénior, social, grado, guía

1. Introducción

En las universidades españolas han ido apareciendo unos nuevos estudios que toman diferentes nombres dependiendo del ámbito

geográfico, es decir, encontramos titulaciones como *Títol Sènior en Cultura, Ciència i Tecnologia* (Título Sénior en Cultura, Ciencia y Tecnología) de la *Universitat de Lleida*, UNED-Sénior de la UNED, Universidad de Mayores de la Universidad Complutense de Madrid así como una larga lista en las distintas comunidades. Se celebran unos encuentros nacionales para la puesta en común de los diferentes programas. El último tuvo lugar en Lleida y estuvo centrado en la evaluación¹

El objetivo de los mismos es dotar de conocimientos avanzados en diferentes disciplinas tales como la biología, informática o la historia a personas que dependiendo del programa acceden desde los 50 a los 55 aproximadamente^{2,3}.

Son personas que después de haber acabado la mayoría su vida laboral esperan aprender de forma pausada y disfrutar de una nueva experiencia⁴. Aquí como es de suponer encontramos también que ambos factores tienen un peso diferente dependiendo de la persona.

El perfil de los egresados es muy variopinto en cuanto a su nivel de conocimientos. En este artículo nos centraremos concretamente en los conocimientos sobre informática en general. Curiosamente la aproximación a esta problemática se ha afrontado desde el punto de vista de la pedagogía [7] sin que existan demasiados debates entre la profesión de informáticos, pues estos últimos rechazan generalmente la impartición de estas asignaturas. Este artículo nace de la creencia que la actitud de los ingenieros informáticos debería cambiar en su mayoría respecto a la impartición de estas asignaturas.

¹ <http://www.senior.udl.cat/11encuentro/>

² http://redaberta.usc.es/uvi/public_html/images/pdf2005/MORALES%20CALVO,%20SONIA.pdf

³ http://www.fadaum.org/revistas/madurez_activa_15.pdf

⁴ http://issuu.com/asauma/docs/revista_contracorriente_n_1

Con este fin, se ha organizado el artículo de la siguiente manera, se verán los retos de aprendizaje que presentan los diferentes perfiles a continuación, para pasar en el tercer punto a ver las estrategias y actividades utilizados para la enseñanza de los diferentes cursos. La última parte presenta una reflexión sobre el futuro de la enseñanza a este colectivo respecto a asignaturas con contenidos informático.

2. Perfiles y retos de aprendizaje

2.1. Perfiles

En concreto, en el ámbito informático durante los últimos cuatro años se han podido distinguir los siguientes perfiles :

- Amas de casa sin experiencia alguna con el ordenador. En este perfil, había casos más preocupantes como el caso en que no habían tenido contacto alguno con una máquina de escribir, lo que dificultaba en exceso la redacción de cualquier documento o ejecución de órdenes. Es concretamente una cuestión de género, pues no se presentaba esta problemática en el género masculino. La proporción ha oscilado estos cuatro años entre el 5% y el 10%
- Personas con poca familiaridad con los ordenadores. Es el perfil que muestra mayor proporción, pues oscila en ese periodo de tiempo entre el 50% y el 60%
- Personas que han utilizado ordenadores de manera sistemática para tareas concretas en sus puestos de trabajo 20% y el 35%
- Personas que realizan tareas complejas con ayuda de un ordenador. También nos encontramos con una discriminación de género en este punto. Pues, en cuatro años sólo he encontrado una mujer que cumpliera este punto, habitualmente han sido hombres. El porcentaje de este perfil ha sido muy bajo, tratándose del 5% o incluso inferior.

Cada perfil presentaba una problemática diferente que afrontar, de manera que el perfil de las amas de casa tendía a adoptar una actitud muy negativa respecto al aprendizaje, existiendo un rechazo inicial al mismo muy fuerte, al tiempo que experimentaba una fuerte frustración y cierta ansiedad que juntamente con la alta auto-exigencia lo convertía en el colectivo más

problemático. Dicha problemática también aparecía pero en menor grado en las personas con poca familiaridad con los ordenadores. Los dos últimos perfiles son los que se adaptan más fácilmente a los cursos y decididamente marcan el rumbo de la clase, pues suelen ser estos alumnos los que se convierten en los líderes del grupo de clase y dependiendo de la vertiente colaborativa/competitiva son una gran ayuda/inconveniente en la labor del profesor.

2.2. Discapacidades

No es el objetivo de este trabajo exponer de una forma pedagógica o psicológica la problemática en el aprendizaje de las personas de edad avanzada, pues existen excelentes trabajos acerca de estos temas [1] [2] [3] [4] [5] [6], sino resaltar la problemática concreta respecto a la tarea de trabajar con el ordenador y desarrollar tareas de diferente complejidad.

Uno de los elementos clave a la hora de plantear contenidos en las asignaturas que se debe tener en cuenta son las discapacidades que pueden presentar los alumnos, pues modifican el formato del material que se les suministra, como por ejemplo: tamaño de la letra de los documentos, colores, contraste entre forma y fondo entre otros. Estos problemas también pueden conllevar problemas de ubicación dentro del aula, aquellas personas que presenten sordera severa o dificultad de visión deberán tener un lugar privilegiado en el aula. Este puede ser un problema grave dependiendo de las características de la misma.

Los alumnos suelen informar de cualquier dificultad y resolverla es tarea conjunta del profesor y de la clase, pues los compañeros deben facilitar la tarea del espacio o de movilidad a las personas que lo necesiten. Todos los años ha habido problemas de 2 o 3 personas con problemas de audición y la gran mayoría de pérdida de visión (90%)

2.3. Motivación

Es un apartado que debe tenerse en cuenta antes de general cualquier tipo de material. El profesor debe preguntarse qué lleva a matricularse a los alumnos a estos cursos, pues de ese objetivo va a

depender el éxito o fracaso de los contenidos que desee impartir.

Se parte de que son personas que ya han acabado su vida laboral en un 80%, y el resto no busca otro empleo. Así que la principal motivación que lleva a una persona de 18 años a la universidad no tiene nada que ver con el colectivo que nos ocupa.

Encontramos que principalmente vienen porque tienen ganas de seguir aprendiendo, sin embargo, no es este el único motivo, es más, para muchas de estas personas la parte social, es decir, tener un grupo de compañeros con los que relacionarse supone la razón de hallarse en este programa. En realidad, supone un lastre en el proceso de aprendizaje, pues al no ser el principal objetivo, se encuentra que existe siempre un grupo de personas que les cuesta sobremanera avanzar en conocimientos; lanzo aquí la reflexión de si en estos programas deben tener cabida más contenidos sociales en asignaturas técnicas. Para ilustrar este hecho me gustaría resaltar que los alumnos que acabaron cuarto el año pasado, la mitad expresaron un cierto malestar por no poder seguir haciendo cursos, por lo que se ha propuesto un Postgrado que todavía se está desarrollando en nuestra universidad.

A nivel individual, se han encontrado alumnos que expresaban su rechazo a salir de casa porque no se sentían con ánimos, ya que no existían clases a las que asistir cuando habían llegado al último curso.

2.4. Modelos mentales

Tampoco me extenderé en este punto, pero sí que se debe destacar el hecho de que al presentar diferentes modelos mentales, por tratarse de personas de generaciones de mayor edad, existen metáforas que no son aceptadas de inmediato.

Para ilustrar este punto me remito a los dispositivos de almacenamiento que se utilizan habitualmente conocidos como 'usb' o 'pen'. Mientras que la metáfora del escritorio como simbología para el trabajo con el ordenador como si de un escritorio se tratase, donde ponemos carpetas y tiramos elementos, es válida hasta que estos dispositivos entran en escena. Aproximadamente un 10% del alumnado presenta problemas en interiorizar que el dispositivo de almacenamiento pasa a ser una carpeta como el

resto de almacenamiento de un disco duro. Esperan encontrar otra cosa que tampoco saben describir y la parte más crítica es que no saben como encontrar estos dispositivos, curiosamente no miran en iconos como 'mi PC'.

3. Estrategias y actividades

En concreto, se ha trabajado durante cuatro años, un total de 2 asignaturas de media cada año, pues ha habido veces que se ha dado soporte a una tercera. Estas asignaturas reciben el nombre Taller I y Taller IV. Se cursan en el Programa de Mayores *Títol Sènior en Cultura, Ciència i Tecnologia* (Título Sénior en Cultura, Ciencia y Tecnología) de la *Universitat de Lleida*

Cada asignatura dura un cuatrimestre y las sesiones son de 1 hora y media aproximadamente. Siempre se suelen alargar unos 10 minutos aproximadamente resolviendo dudas.

Respecto al número de alumnos de nueva incorporación, se ha visto incrementado año a año, partiendo de unos 15 aproximadamente el primer año a unos 23 en el último año.

Ambas asignaturas son obligatorias.

3.1. Asignaturas

La asignatura de Taller I, se imparte el primer cuatrimestre del primer curso, lo que significa que la cursan alumnos recién incorporados a estos estudios. Se marcó como objetivo de esta asignatura el resolver ayudándose de la ofimática la entrega de trabajos. Es decir, en el marco de estos estudios la gran mayoría de profesores no evalúan a los alumnos mediante exámenes sino que deben realizar trabajos además de otras actividades. Es por este motivo que parecía lógico ayudar a los alumnos a saber presentar un trabajo por ordenador utilizando una herramienta ofimática. Habitualmente venimos trabajando con dos herramientas: Microsoft Word y OpenOffice. La primera fue escogida porque la mayoría de alumnos tienen el software de Microsoft y la segunda por ser de distribución gratuita.

La elección del software presenta una problemática añadida que es la versión. Es una dificultad añadida consensuar una versión con los alumnos, pues aunque en clase se muestre una en particular, cada alumno en su casa intenta reproducir los ejercicios o bien avanzar en ellos y

al interactuar con una versión diferente suelen bloquearse, pues la tarea de encontrar una instrucción cambia sensiblemente de versión en versión.

Aunque en las aulas se ha utilizado versiones relativamente antiguas del paquete Office, como la del 2003, algunos alumnos al comprar ordenadores nuevos y tener versiones superiores como la 2007 han experimentado verdaderos quebraderos de cabeza para encontrar las funcionalidades que habían aprendido en clase ya que el sistema de menús había cambiado considerablemente.

Esta asignatura muestra un planeamiento totalmente individual debido a dos motivos, la disparidad de conocimientos y la necesidad de autoafirmación que tienen frente a los nuevos retos que se les presentan. Aunque es bien cierto que por la dinámica de las clases se crean algunas sinergias entre grupos de 2 alumnos, pues es el número en el cual se sienten cómodos trabajando, siempre y cuando sean ellos mismos los que se agrupen por afinidad. De esta manera, a medida que la asignatura avanza se apoyan el uno en el otro para seguir avanzando en el aprendizaje.

La asignatura de Taller IV, se imparte en cuarto curso, así pues la realizan alumnos que tienen un bagaje en estos estudios y la problemática del primer curso apenas aparece.

En esta asignatura se profundiza en el conocimiento del trabajo colaborativo en la red, de manera que se trabaja con la aplicación *Google Docs*, repasando la ofimática en red y aprendiendo búsqueda avanzada en buscadores, es decir, viendo cómo mejorar las búsquedas refinando por tipo de fichero, tamaño, fechas, etc. Dado el carácter colaborativo de las herramientas utilizadas para tal efecto, se tomó la decisión de formar grupos aleatoriamente, pues ya eran alumnos que llevaban cuatro años conviviendo en las clases. No fue una decisión bien recibida en un principio aunque la actividad planteada fuese del agrado de todos. En otras asignaturas se habían visto obligados a trabajar en grupo y esa idea no les había gustado en general, los motivos y preocupaciones quedaban bastante claras:

- Inseguridad respecto a la evaluación, pensaban que no quedaría justamente reflejada la valoración del trabajo de cada uno en la nota.

- Carácter personal.
- Diferente nivel de conocimientos.
- Las tareas eran abordadas de diferente manera, lo cual creaban conflictos que no deseaban enfrentar.

Pese a los anteriores recelos trabajaron en grupo satisfactoriamente.

De forma genérica es necesario resaltar, que en este tipo de clases el profesor debe prestar una atención muy personalizada a cada alumno como veremos en el diseño de las actividades, lo que conduce a una deseada ratio muy baja entre profesor y alumno como se comenta en las conclusiones

3.2. Estrategias

Las estrategias son completamente diferentes dependiendo en gran manera del curso donde se realiza la asignatura.

Las clases magistrales son muy numerosas al principio de la asignatura de primer curso (Taller I), pues ayudan a contextualizar al alumno dentro del proyecto que deben realizar.

En todo momento desde el primer día conocen el objetivo de la asignatura y lo tienen presente todos los días, pese a eso siempre se encuentran alumnos que no saben que deben realizar.

La asignatura al estar planteada como transversal al resto y de ayuda para las otras, normalmente es bien recibida, no obstante en estos estados iniciales como se ha comentado en el anterior punto, los perfiles con menor conocimiento sobre la misma necesitan mayor atención, así que se debe compaginar en los primeros estadios estas clases magistrales con una atención personalizada sobre todo en estos colectivos.

Las universidades utilizan gestores de contenidos para comunicarse con sus alumnos y éste es uno de los primeros puntos que presenta una barrera para ellos, así que hay que asegurarse que son capaces de trabajar con ellos de una forma óptima, por lo que parte de Taller I en un principio está dedicada al gestor para poder realizar acciones básicas como:

- Acceder a correo.
- Acceso a los materiales electrónicos.
- Debates

- Chat

A medida que se avanza en los contenidos de la asignatura el número de clases magistrales disminuye encontrándonos que pasadas $\frac{3}{4}$ partes del cuatrimestre apenas existen. Consiguiendo que el trabajo de los alumnos sea autónomo o bien de forma colaborativa en grupos de 2 que se han formado de forma espontánea. Solamente en una ocasión en cuatro años un alumno que tenía amplios conocimientos informáticos se convirtió en un soporte para todos sin excepción.

A medida que las clases magistrales han ido disminuyendo se hace más necesaria la atención personalizada, ya que en principio cuando realizan acciones que les llevan a caminos inesperados la mayoría presentan pocos recursos para volver a seguir la tarea propuesta.

La dinámica de Taller IV es totalmente diferente, el trabajo es colaborativo, pues también como se ha comentado lo fomentaba la filosofía de las herramientas colaborativas utilizadas en software. En este caso las clases magistrales son más bien escasas, las justas para contextualizar el marco de trabajo y que rápidamente comiencen a trabajar de forma autónoma cada grupo.

La atención personalizada se reduce a los grupos, donde normalmente hay un miembro que se erige como portavoz, aunque no todos los grupos tengan la misma dinámica. Durante todo el cuatrimestre la atención personalizada es la principal interacción con los alumnos, resolviendo las principales dudas para que puedan avanzar.

3.3. Actividades

También en este aspecto, se trabajan las actividades de manera muy diferente, según se trate de alumnos de primer curso o del último curso.

Así pues, en el primer curso dados los perfiles que se han comentado en secciones anteriores, se generan diferentes actitudes frente a la realización de actividades:

- Cualquier actividad es muy complicada y no voy a ser capaz de afrontarla.
- Iré poco a poco y saldré adelante
- Yo ya sé cualquier cosa que me hayan explicado.

Todos los años se han dado las dos primeras actitudes, y mediante las actividades se pretendía sobre todo que todo el mundo tuviese la segunda actitud, generalmente había un 100% de éxito, sin embargo, este último año han aparecido personas que presentan la última actitud, principalmente la razón de ello se cree que es porque cada vez más el ordenador llega a las casas y todos los miembros participan de él.

A continuación se detallará cómo se diseñan las actividades para conseguir el paso de la primera actitud a la segunda.

Se enmarca el trabajo de todo el cuatrimestre en un único proyecto, el cual, como he comentado tiene que ser recordado cada sesión. En este caso, saber realizar un trabajo en un procesador de textos.

Se desglosa en pequeñas actividades, de manera que el proyecto tiene diferentes secciones, como son:

- Realizar una portada
- Confeccionar un índice de contenidos, un índice de tablas y un índice de ilustraciones de forma automática.
- Desarrollar un tema en forma de capítulos, lo que significa realizar secciones
- Crear una bibliografía

Así pues, a modo de ejemplo la primera sección permite enseñar acciones que se repetirán en el resto: realizar una tabla para maquetar, insertar imágenes, alinear texto, cambiar tamaño del texto, cambiar el color del texto, etc... De esta manera se diseñan actividades sobre cada una de esas acciones, y se repiten las actividades en contextos diferentes y con elementos diferentes. También se va incrementando la dificultad de la actividad a medida que se avanza en el cuatrimestre.

Se tomó la decisión de hacerlo así porque salvo las personas que poseen amplios conocimientos sobre informática, las cuales constituyen una minoría, el resto, incluso aquellos que realizan una tarea mecánica de forma excelente, tienen verdaderos problemas para recordar las acciones que he comentado. De esta manera, es decir, practicando reiteradamente una acción, la asimilan mucho mejor que enseñarla en una clase magistral. Es por eso, que las clases magistrales sirven para indicar lo que se va a

hacer y el por qué se va a hacer, pero la gran mayoría del tiempo debe dedicarse a qué se va a hacer, y necesitan practicarlo de forma autónoma para avanzar.

El proceso de aclimatación dura alrededor de dos meses, por lo cual en el ecuador de la asignatura habitualmente todos se encuentran en el segundo punto que he apuntado respecto a la actitud.

No se puede hacer una estimación respecto a los tiempos que se dedican a cada actividad así como la repetición o la variación de la misma, pues depende al final de cada alumno, pues, se va personalizando en cada sesión.

Aunque en un principio los puntos de partida respecto a conocimientos son dispares, a medida que va avanzando el curso, los conocimientos se igualan, pues curiosamente, aquellos que conocían en profundidad la herramienta muestran verdaderos problemas cuando tienen que aprender nuevas acciones o bien nuevas formas de realizar aquellas acciones que ya sabían. Suelen opinar que con lo que saben ya pueden resolver los problemas que puedan acontecer. Por el contrario, las personas que se mostraban escépticas respecto a lo que podían aprender, a menudo son las que al final de la asignatura muestran una disposición más optimista e incluso en muchos casos han avanzado más que los compañeros que poseían más conocimientos en un primer estadio del aprendizaje.

Respecto a la segunda asignatura, Taller IV, las acciones son de mayor complejidad y más abiertas:

- Gestionar documentos colaborativos.
- Confeccionar grupos que puedan compartir documentos
- Insertar material multimedia

Alrededor de estas acciones se articulan un numeroso grupo de actividades, que son más o menos complejas dependiendo del nivel de conocimientos de cada grupo.

Se organizaron los documentos colaborativos alrededor de un proyecto que cada grupo se personalizó: realizar unas vacaciones. De esta manera elegía cada grupo a qué lugar del mundo querían irse, debían generar tres tipos de documentos: uno donde escribiese el itinerario, un segundo donde se calculase el coste y un tercero

que iba a mostrar de forma gráfica los lugares a visitar, de esta manera, trabajaban con un procesador de textos, una hoja de cálculo y una presentación de forma colaborativa. Los miembros del mismo equipo trabajaban cada uno en su ordenador con el mismo documento.

Siempre en todas las asignaturas se deben realizar actividades transversales, es decir, aquellas actividades que los alumnos necesitan realizar para dar soporte a sus tareas, concretamente la más repetida era la de la búsqueda de información en Internet, que a su vez integra la recuperación de texto e imágenes. Es necesario recordar que el proceso de aprendizaje mejoraba extensiblemente cuando el alumno entendía el porqué de aquel conocimiento, es decir, al crear la necesidad de tener material para confeccionar un documento, eran los propios alumnos los que demandaban aprender más sobre estas actividades transversales

4. Conclusiones

Hemos visto los factores que influyen en el desarrollo de las clases así como la confección del material. También hemos visto las dificultades en el aprendizaje centrado en el contexto de la informática y concretamente centrándonos en aspectos como la ofimática y multimedia. De manera también se han expuesto y razonado las soluciones tomadas así como también la reacción a las mismas por parte de los alumnos.

Los requisitos óptimos que se deberían cumplir son los siguientes:

- Dependiendo del perfil del alumno se recomienda una relación de un profesor por cada seis alumnos en el caso de las amas de casa, hasta una relación de 1 a 10 en el caso del último nivel que corresponde a los alumnos avanzados en el uso de ordenadores.
- Numerosos asientos cercanos al profesor así como a la pantalla de proyección, para facilitar el aprendizaje a aquellos alumnos que presenten problemas de audición y/o visión.
- Documentación accesible donde se haya cuidado el contraste, color y el tamaño de los caracteres entre otros factores.

- Numerosos ejercicios con igual contenido y diferentes escenarios para que puedan practicar en casa.
- Control de versiones del software. Si la interfaz cambia, puede provocar que los alumnos aún sabiendo realizar una tarea en una versión se sientan incapaces de realizarla con otra versión del mismo software, como es el caso del Microsoft Word.
- Aproximación diferente a cada perfil, se deben diseñar actividades diferentes dependiendo de cada perfil.

Como temas abiertos y difíciles de resolver destacaría la evaluación [8], pues se pueden encontrar multitud de opiniones a favor y en contra respecto a la ponderación de los factores que se tienen que tener en cuenta, pero plantearía como pregunta abierta sobretodo la cuestión de progresión vs nivel de conocimientos pensando en si debe integrarse la labor social, ya que si la respuesta es afirmativa, deberíamos cuestionarnos si los ingenieros informáticos están preparados para ello y si desearían estarlo.

Referencias

- [1] Barberá Heredia, E. *Modelos explicativos en Psicología de la Motivación*, Revista electrónica de Motivación y Emoción, 5(10). 2002
- [2] Campo Adrián, E. *Dificultades de aprendizaje y tercera edad*: Prácticas. Cuadernos de la UNED. N° de la colección 35191. 978-84-362-4040-5. Madrid. 2003
- [3] Catena Mir, J. *Definición y valoración de la demencia avanzada*. *Revista Multidisciplinar de Gerontología* [Barcelona] : Nexus, 1998. Vol. 16 núm. 4 (octubre-diciembre 2006) 176-182 p. 1139-0921.
- [4] Cuervo, A.H. Imágenes de la vejez y nuevas tecnologías, *Rev. Tiempo N° 4*, marzo/ 2000.
- [5] Fundación Ingema (Fundación Instituto Gerontológico Matia). *Las dimensiones subjetivas del envejecimiento: premio Imsero Infanta Cristina 2008 / Madrid : Imsero, 2009. -- 165 p. -- (Estudios. Personas Mayores ; 6) p. 161-163. ISBN 978-84-8446-122-7.*
- [6] Gómez Bedoya, M. *El aprendizaje en la tercera edad. Una aproximación en la clase de ele: Los aprendientes mayores en el Instituto Cervantes de Tokio*. Biblioteca Virtual . Número 10. Primer Semestre 2009.
- [7] Morales Calvo, Sonia. *La asignatura de informática en la Universidad de mayores: Una experiencia en la Universidad de mayores José Saramago de la Universidad de Castilla- La Mancha*”. Tesis Dirigida por Dr. González Jiménez, F. E, 2009
- [8] *Revista madurez Activa*. (Especial verano 2010) Año 4 / N. 13 (p. 54/57)

Diseños para todos. Una experiencia en primero de grado

Juan José Escribano Otero¹,
M^a José Terrón López²

¹Dpto. Sistemas Informáticos

²Dpto. Electrónica y Comunicaciones

Universidad Europea de Madrid

28670 Villaviciosa de Odón

{juanjose.escribano, m_jose.terron}@uem.es

Resumen

Uno de los objetivos de la formación universitaria es el desarrollo de competencias en los estudiantes, siendo una de esas competencias a desarrollar en los futuros ingenieros la formación en valores. Por otro lado un ingeniero tiene que ser capaz de resolver problemas. Los ingenieros tienen que ser capaces de diseñar distintos artefactos (hardware, software, productos o servicios) “para todos”. No todos los usuarios van a ser “usuarios tipo” con buena vista y mejor pulso. Por tanto, parece conveniente que nuestros estudiantes sean capaces de diseñar “para todos”...

El artículo explica la experiencia realizada en una asignatura de primero de grado (en varios grupos), donde se pide a nuestros alumnos que diseñen soluciones para que una persona concreta con discapacidad intelectual supere las barreras con las que se encuentran en su entorno.

Summary

One of the purposes of university education is the skills development in their students. One of these skills is the ethical one among being able to find solutions to problems. Tomorrow's engineers will have to design different devices (hardware, software, products or services) for everyone, not only for “standard users”.

This paper explains the experience done in a first year degree subject where the student is asked to design a device to overcome a real barrier to a real person whom they will meet. This person has an intellectual disability.

1. Introducción

En este artículo se cuenta la experiencia desarrollada con alumnos de las titulaciones de grado de diversas ingenierías (Informática e Industrial) en el marco de la asignatura transversal “Habilidades Comunicativas en la Ingeniería”. Como sabemos los nuevos títulos de grado [1] implican un diseño de las asignaturas basado en competencias y objetivos de aprendizaje. Tengamos en cuenta que los egresados al optar a un empleo son evaluados no sólo por sus conocimientos técnicos si no en gran medida por las competencias personales que puedan aportar para el desempeño de sus tareas (“Accenture” y la plataforma “Universia” [2], CHEERS [3], Libro Blanco Grado Informática [4], Hoffmann [5], REFLEX [7], TUNING [7]).

Entre estas competencias a desarrollar en nuestros alumnos una de ellas es la de educar en la diversidad. Ahora bien, educar en la diversidad no se basa sólo en la adopción de medidas excepcionales para las personas con necesidades educativas específicas, sino en la adopción de un modelo de currículo que facilite el aprendizaje de todos los alumnos en su diversidad y el saberse enfrentar a la misma. Igualmente, se trata de un modelo integrador pensando no sólo en las deficiencias o en las necesidades educativas especiales, sino en la potenciación de las capacidades propias de cada individuo, es decir en la individualización. Por otra parte, una de las competencias que atañe también a los estudios universitarios es la formación en valores ya que la universidad es parte activa de la construcción de una ciudadanía global y responsable con los demás y con el entorno. Esto queda refrendado en la Ley

Orgánica 4/2007, de 12 de abril, por la que se modifica la Ley Orgánica 6/2001 de Universidades del 21 de diciembre [11], en la cual se establece explícitamente la responsabilidad de la universidad en la formación en valores. La legislación recoge así un aspecto clave en el informe Delors [12] en el que se establece que la educación en su pleno desarrollo exige basarse en cuatro pilares fundamentales: aprender a conocer, aprender a hacer, aprender a vivir juntos y aprender a ser. Cada vez más, aunque aún tímidamente, algunas universidades plantean experiencias para la incorporación de los valores en las aulas ([13], [14] y [15]).

Nuestra experiencia se basó en el hecho de que los problemas son cuestiones que pueden responderse mediante la aplicación de conocimientos previos interiorizados, pero que además pueden actuar como evaluadores del aprendizaje y como indicadores de la necesidad de conocimientos más complejos de los que en ese momento el estudiante tiene disponibles (es en cierta medida la base del aprendizaje basado en problemas). Para el estudiante será tan importante llegar a una solución exitosa como detectar la frontera para saber cómo encaminar sus futuros aprendizajes.

Paul Halmos en 1991 dijo “Lo que se puede enseñar es la actitud correcta ante los problemas, y enseñar a resolver problemas es el camino para resolverlos (...). El mejor método no es contarles cosas a los alumnos, sino preguntárselas y, mejor todavía, instarles a que se pregunten ellos mismos” [8].

El planteamiento inicial de la actividad se basa en que los alumnos son futuros ingenieros y que es necesario concienciarles de que, entre otras cosas, ser ingeniero significa estar al servicio de las personas, llevar a cabo proyectos para ellas. Para conseguir realizar su trabajo de ingenieros tendrán que comunicarse con personas de toda índole (como sus potenciales clientes) y ello incluye, en ocasiones, comunicarse con personas que tienen dificultad para transmitir sus necesidades y ellos tendrán que conseguir obtener esa información si quieren que el proyecto sea finalmente una realidad. En este marco, nos pusimos en contacto con la Confederación Española de Organizaciones en favor de las Personas con Discapacidad

Intelectual (FEAPS) [9] con la idea de que nos sugirieran distintas situaciones basadas en personas a las que nuestros alumnos tenían que buscar soluciones. FEAPS por su parte nos puso en contacto con la “Fundación APROCOR” [10] quien trabaja para ofrecer oportunidades que fomenten la realización personal plena, la independencia personal y la integración social y laboral de personas con discapacidad. Así pues invitamos a algunas de las personas de la “Fundación APROCOR” a visitarnos para que les contaran a nuestros alumnos con qué barreras se enfrentaban las personas con discapacidad intelectual. La idea fundamental era enfrentarles a pensar en qué “artefacto” podrían diseñar que les facilitara su independencia personal y la integración en su ámbito laboral y social.

2. Objetivos de la experiencia

Según el diccionario de la Real Academia de la Lengua Española (RAE) un ingeniero es un “hombre que discurre con ingenio las trazas y modos de conseguir o ejecutar algo”. En definitiva, ser ingeniero implica ser un profesional que tendrá que diseñar distintos artefactos (ya sea hardware, software, productos o servicios) para todo tipo de usuarios. Muchas veces, estos usuarios son personas demasiado similares al diseñador: universitarios de entre 25 y 35 años, con buena vista y mejor pulso. Pero no siempre va a ser así, por lo que parece conveniente diseñar “para todos”... Precisamente un “diseño para todos” es lo que se solicita a los profesionales cuando uno se mueve en entornos dedicados a la integración de personas con discapacidad.

Con esta perspectiva de fondo y con la idea de que formar a nuestros estudiantes implicaba no sólo “adiestrarles” si no también acercarles a la sociedad actual, nos planteamos diseñar una actividad con nuestros alumnos de primer curso en la que alcanzaran los siguientes objetivos:

- Tomar conciencia de que las personas con discapacidad intelectual son clientes potenciales.
- Conocer las barreras con las que algunas personas con discapacidad intelectual se encuentran en su día a día.
- Identificar como ámbito de trabajo la mejora de la accesibilidad de los entornos mediante el

diseño de adaptaciones para que éstos sean comprensibles, practicables y utilizables por todas las personas.

Para ello, se diseñó la actividad en distintas fases cuyo fin último era que los alumnos alcanzaran los objetivos propuestos con éxito.

La primera fase de la experiencia consistió en organizar una sesión de sensibilización y formación donde se centraría a los futuros ingenieros en los problemas con los que se podrían encontrar en su ámbito laboral o incluso de su vida diaria. Esta sesión constaba de:

- Presentación del proyecto por parte de los profesores responsables de la asignatura.
- Charla para contextualizar por parte de profesionales de la Fundación APROCOR. En dicha charla se les habló de algunos aspectos clave de la discapacidad intelectual, se les mostró la visión del preparador laboral de dicha fundación y se les mostraban distintas realidades a las que se enfrentan los miembros de dicha fundación.
- Explicación del desarrollo de las prácticas a desarrollar por parte de los alumnos desde el punto de vista del profesor y desde el punto de vista de las personas con discapacidad.

En segundo lugar, se les explicó a los estudiantes los proyectos que iban a poder desarrollar. Para ello profesionales de la Fundación APROCOR les presentaron las situaciones de siete personas con discapacidad intelectual en diferentes contextos laborales y de vivienda. Para realizar esta presentación vinieron a nuestra universidad algunas de estas personas con discapacidad de forma que los alumnos conocieran de primera mano los problemas con los se tenían que enfrentar.

En esta presentación se realizó una breve descripción no sólo de cómo es la persona sino también de cómo es el contexto donde se desarrolla la situación y las barreras con las que ha de enfrentarse cada una de esas personas en su día a día.

Tras esta descripción, los alumnos tenían que identificar qué barreras encuentra la persona en su entorno para poder alcanzar el objetivo deseado o bien para poder realizar la actividad que quiere. Así mismo, se les propuso como objetivo el proponer diferentes adaptaciones y

apoyos que permitieran a esas personas desarrollar su objetivo o actividad deseada con éxito.

3. Realización de los proyectos

Para la realización de los proyectos propuestos, se sugirió a los alumnos que trabajaran en grupos de 4 o 5 personas con la finalidad de que trabajaran de forma colaborativa enriqueciendo sus soluciones con las ideas aportadas por cada uno de ellos.

A cada grupo se le asignaba una situación vinculada a una persona con discapacidad. Para facilitarles el desarrollo del proyecto se les entregaba una ficha de esa persona y situación donde se detallaba:

- Una descripción general de la persona
- Las habilidades y competencias donde más destaca la persona en cuestión (sus puntos fuertes).
- Las áreas donde la persona necesita más apoyos (teniendo en cuenta que entendemos por apoyos aquellos recursos que pueden ayudar a una persona con discapacidad a conseguir una completa integración y normalización).
- Las características del entorno donde se desarrolla la situación.
- Las barreras, tanto físicas como de acceso a la información, con las que se encuentra la persona en su entorno laboral o personal.
- El objetivo u objetivos de las adaptaciones buscadas.

En base a esas fichas, los alumnos tuvieron que elaborar una propuesta de proyecto en el que se detallaran las adaptaciones propuestas para facilitar a la persona conseguir sus objetivos.

Los alumnos por tanto tenían que elaborar un documento técnico en el que explicaran su propuesta de proyecto a “sus clientes” pero, además, dicho documento debía ser redactado de forma que pudiera ser comprendido por la persona “cliente”. Es decir, la parte menos técnica del documento debía ser comprensible y accesible para una persona sin formación en ingeniería.

4. ¿Cómo se evaluaban los proyectos de los alumnos?

Para evaluar las propuestas que los alumnos hicieron para facilitar la vida a las personas que les propusimos se decidió hacerlo en una doble vertiente. Por un lado, los profesores evaluábamos la actividad como una actividad más de la asignatura, pero, por otro, los proyectos fueron remitidos a la “Fundación APROCOR” para ser analizados y valorados por los propios “clientes”. Para evaluar los proyectos los alumnos sabían que dicha evaluación se basaría en los siguientes criterios:

1. Accesibilidad: es decir, la medida en que la adaptación que proponen hace que el entorno sea más comprensible, utilizable y practicable por la persona en condiciones de seguridad y comodidad, y de la forma más autónoma y natural posible.
2. Funcionalidad: es decir, la medida en que la adaptación es práctica y útil para la persona, de forma que le permita mejorar en aspectos cotidianos de su vida, ser más autónoma y por lo tanto mejorar aspectos relacionados con su calidad de vida.
3. Viabilidad: es decir, la medida en que la adaptación se puede llevar a cabo satisfactoriamente teniendo en cuenta su coste, el acceso a la misma y las posibilidades de su aplicación en la vida de la persona y en el entorno.
4. Normalización: es decir, la medida en que las adaptaciones permiten a la persona desarrollar su vida en condiciones de normalidad, disfrutando de los espacios y recursos en igualdad de condiciones que el resto de las personas.
5. Innovación: es decir, la medida en la que la adaptación no supone más de lo mismo, sino que, de forma novedosa y sencilla, se consigue que la vida de las personas sea más fácil y exitosa.

Hay que decir, que a pesar de la dificultad que encontraron los estudiantes para la búsqueda de estas soluciones, casi todos ellos lograron obtener soluciones cumpliendo con los criterios exigidos.

5. Resultados de la experiencia

Según los comentarios vertidos por la “Fundación APROCOR”, parece ser que la experiencia ha sido altamente gratificante para los profesionales de dicha Fundación y la consideran muy estimulante para las personas con discapacidad intelectual que participaron. Dicha satisfacción se suma a la expresada tanto por los profesores como por los alumnos de la Universidad Europea de Madrid. Los alumnos en general expresan que, aparte de haberse puesto en contacto con “otras” realidades, uno de los aspectos que más les ha gustado de esta experiencia, es que han sentido la cercanía con la profesión y que lo que están estudiando efectivamente les va servir para algo. De hecho fue una actividad en la participaron activamente casi el 100% de nuestros alumnos.

En el marco de la asignatura, debían hacer un blog donde volcaban sus reflexiones. Incluimos aquí algunas de ellas por considerarlas paradigmáticas:

“Me parece que esta actividad con la fundación APROCOR me parece que ha sido la actividad más interesante del curso pues nos ha ayudado a solidarizarnos y a darnos cuenta de la situación que tienen algunas personas y de que entre todos podemos hacer cosas para ayudarlos”.

“me pareció muy interesante y me ayudó a darme cuenta que en un futuro me puede tocar trabajar al lado de personas con discapacidad y hay que saber tratar con ellas.”

“En mi opinión es una de la practicas más interesantes de todas ya que de cierta forma nos adentra en el mundo laboral haciéndonos caer en la cuenta de la infinidad de posibilidades que significa ser ingeniero, aparte que la actividad en sí.”

“Tengo que decir que no ha resultado sencillo, porque siempre tendemos a buscar soluciones costosas y por tanto difíciles de realizar.”

“Esta actividad nos ha ayudado a entender todos los problemas que pueden llegar a tener los discapacitados intelectuales en su vida cotidiana. Personalmente, me ha encantado este proyecto y aunque no he podido aportar tanto como mis compañeros, porque tuve que trabajar en navidades. Me parece que las universidades

tenían que favorecer más este tipo de ideas en las que aparte de conseguir una nota, lo que puedes lograr es hacerle la vida un poco más fácil a gente que verdaderamente lo necesita.”

“Esta actividad nos ha abierto la mente, a que las personas con discapacidad son unos futuros clientes potenciales, para nosotros que estamos estudiando ingeniería.”

“Primeramente me gustaría decir que creo que la experiencia fue un tremendo éxito, a todos nos viene muy bien que de vez en cuando nos recuerden que hay gente que no tiene tanto como nosotros y que aun así, se esfuerza muchísimo por hacer una vida normal. En segundo lugar quisiera expresar que los proyectos, aunque a nosotros nos puedan parecer nimiedades, son realmente difíciles de llevar a cabo, en nuestro caso, nos ha costado muchísimo encontrar una solución y aun así a mí personalmente no me llega a convencer del todo, pero soy incapaz de pensar algo mejor. Para terminar, decir que es espectacular lo que está haciendo la gente que trabaja en esta fundación así como las personas afiliadas a ellas.”

“Esta actividad nos ha ayudado a entender todos los problemas que pueden llegar a tener los discapacitados intelectuales en su vida cotidiana.[...] Este proyecto nos permite de alguna manera, ayudar a personas que tienen ciertas discapacidades, para que puedan llevar una vida independiente. A nosotros también nos ayuda para en un futuro, trabajar en equipo y desarrollar nuevos "inventos" capaces de facilitar la vida a todo el mundo.”

Hay que decir que estas reflexiones se solicitaron a los alumnos una vez que la asignatura estaba evaluada y calificada de forma que se evitaran sesgos.

Por otro lado, quedan pendientes unos cuestionarios que se les van a pasar a los alumnos ahora que ha finalizado la asignatura y cuyos datos estarán recogidos, casi con toda seguridad, en el momento de la celebración de JENUI 2011, por lo que en la exposición de la ponencia se podrán poner a disposición de los asistentes y de los organizadores.

Por último, parece adecuado destacar la participación en la experiencia de la Unidad de discapacidad perteneciente a la Oficina de Responsabilidad Social de la UEM [16]. Gracias

a su apoyo, se consiguió dotar a la experiencia de la cobertura presupuestaria y logística necesaria. Al fin y al cabo, la visita de los profesionales y usuarios de la Fundación Aprocor debía enmarcarse para ellos (sobre todo para los usuarios) no solo como una actividad donde debían contar públicamente detalles de su vida privada, sino como una actividad más amplia, una excursión a una universidad, donde, además de esa exposición pública, podían comer en un comedor universitario y ver ese ambiente vital tan peculiar que todos llamamos “mundo universitario”.

6. Conclusiones

Siendo conscientes de la importancia de la formación en competencias en los grados universitarios, el profesorado universitario actual debe replantear los objetivos de sus asignaturas centrándose no sólo en el desarrollo de determinadas competencias sino también en propiciar espacios para la formación en valores, el razonamiento crítico y el acercamiento a las profesiones y a la individualización.

Para esto se hace vital mostrar a los ingenieros del futuro situaciones reales con sus peculiaridades con el fin de desarrollar en ellos valores además de habilidades comunicativas y capacidad de aplicar conocimientos a la práctica.

Mostramos en este artículo la experiencia desarrollada con ese fin. Esta experiencia ha resultado muy motivadora y enriquecedora para los alumnos y para los profesores participantes.

Como resultado de esta experiencia, en estos momentos el equipo docente participante, la dirección de la Escuela politécnica y los profesionales de FEAPS Madrid, se encuentran diseñando el procedimiento que permita replicar la experiencia en todos los grupos de esta asignatura de los grados de ingeniería y valorando la posibilidad de replicarla en los estudiantes de máster de la Escuela Politécnica.

Agradecimientos

Queremos agradecer a FEAPS Madrid y a la Fundación APROCOR la colaboración prestada para el desarrollo de esta experiencia.

Queremos agradecer a todos los estudiantes que han participado en esta experiencia por su alto grado de implicación con la misma.

También queremos agradecer a la citada Unidad de Discapacidad de la UEM y a su responsable, Eva Moral, el apoyo ofrecido en todo momento y la ayuda ofrecida para conseguir que esta experiencia y otras similares se conviertan en actividades repetidas año tras año.

Referencias

- [1] Real Decreto 1393/2007 del 29 de octubre de 2007 por el que se establece la ordenación de las enseñanzas universitarias oficiales
- [2] ACCENTURE y UNIVERSIA. “Competencias profesionales en los titulados. Contraste y diálogo Universidad-Empresa”, 2007
- [3] CHEERS [Accesible en http://www.uni-kassel.de/wz1/TSEREGS/publi_e.htm]
- [4] Libro Blanco del Título de Grado en Ingeniería Informática: capítulo 9. COMPETENCIAS Y PERFILES PROFESIONALES DEL TÍTULO ACADÉMICO DE GRADO [Accesible en http://www.aneca.es/activin/docs/libroblanco_jun05_informatica.pdf]
- [5] Hoffman, T., Preparing generation Z, ComputerWorld, 25 de agosto, 2003.
- [6] REFLEX. Informe ejecutivo: El profesional flexible en la Sociedad del Conocimiento. ANECA (Unidad de Estudios). CEGES V. 2.0 - 28/06/2007. [Accesible en http://www.aneca.es/estudios/docs/Informe_ejecutivoANECA_jornadasREFLEXV20.pdf]
- [7] Tuning educational Structures in Europe (Proyecto Tuning) financiado por la Comisión Europea en el marco del programa Socrates, [Accesible en <http://www.relint.deusto.es/TUNINGProyecto>]
- [8] Halmos, P.R., Is Computer Teaching Harmful?, Notices of The AMS 38 (1991), 420-423
- [9] FEAPS: Portal de la Confederación Española de Organizaciones en favor de las Personas con Discapacidad Intelectual [accesible en <http://www.feaps.org/>]
- [10] Fundación APROCOR [accesible en <http://www.fundacionaprocor.com/>]
- [11] LEY ORGÁNICA 4/2007, de 12 de abril, por la que se modifica la Ley Orgánica 6/2001, de 21 de diciembre, de Universidades.
- [12] Delors, J. (1996). La educación encierra un tesoro. Informe a la UNESCO de la Comisión Internacional sobre la Educación para el Siglo XXI. Paris: Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura. ED-96/WS/9(S)
- [13] Martínez, M., Buxarrais, M.R. y Esteban, F. (2002). La universidad como espacio de aprendizaje ético. *Revista Iberoamericana de Educación*, 29, 17-44.
- [14] La responsabilidad de los profesores universitarios como agentes del cambio social. Domínguez, F.; Jerome, C.; Rodríguez, R.M. VI Jornadas Internacionales de Innovación Universitaria. ISBN: 978-84-692-1681, 3-4 Septiembre 2009, Madrid.
- [15] La Cooperación al desarrollo en las aulas de la Universidad Europea de Madrid. Rodríguez, R.M.; Domínguez, F.; Ruiz, M. IV Congreso Universidad y Cooperación al desarrollo. ISBN:978-84-490-2572-3, 12-14 Noviembre, 2008, Barcelona.
- [16] Oficina de responsabilidad social de la UEM, <http://www.uem.es/es/conoce-la-uem/responsabilidad-social>

Sesión 8A:
Calidad y evaluación de la docencia

Análisis Empírico del Papel de las Competencias Generales en el Marco de los Estudios Superiores

Jessica Diaz

Jennifer Perez

Agustin Yagüe

Santiago Alonso

Javier Gil

Universidad Politécnica de Madrid (UPM). E.U. Informática

Departamento de Organización y Estructura de la Información

Ctra. Valencia Km. 7 E-28031 Madrid, Spain

yesica.diaz@upm.es

jenifer.perez@eui.upm.es

agustin.yague@upm.es

salonso@eui.upm.es

jgil@eui.upm.es

Resumen

En plena implantación de los nuevos planes de estudio de acuerdo al EEES, las universidades se enfrentan a un nuevo modelo educativo basado en competencias: competencias específicas y competencias generales. Las competencias específicas están asociadas a la adquisición y desarrollo de conocimientos de un área en particular, mientras que las competencias generales son transversales al plan de estudios y definen capacidades, habilidades y/o aptitudes que el alumno debe desarrollar para aplicarlas a lo largo de su carrera profesional. El objetivo de este trabajo es proporcionar una guía al docente sobre las posibles mejoras para tratar el mayor número de competencias generales satisfactoriamente. Concretamente, se ha analizado la manera en la que los docentes están promoviendo y desarrollando las competencias generales con el objetivo de detectar carencias, mejoras y necesidades. El análisis se ha realizado sobre el profesorado de la Titulación de Graduado en Ingeniería del Software de la Universidad Politécnica de Madrid.

Palabras clave

Modelo educativo basado en competencias, Competencias generales, Análisis Empírico.

1. Introducción

En los últimos años, las universidades españolas han tenido y tienen, todavía a día de hoy, que adaptarse al Espacio Europeo de Educación Superior (EEES). El EEES marca unas directrices a seguir, que afectan, no solo a la forma en que el alumno debe aprender, también afectan a la forma en que el profesor debe enseñar. Estas directrices determinan el sistema de créditos ECTS, la aplicación de evaluación continua, y el uso de métodos de apren-

dizaje activos que no se centren exclusivamente en la lección magistral. Pero además, el EEES marca una nueva tendencia en el proceso de aprendizaje del alumno, según la cual no es suficiente con aprender los contenidos de las materias, sino que además, los alumnos deben desarrollar competencias y habilidades que puedan aplicar a lo largo de su carrera profesional. La adquisición de competencias, no sólo afecta al alumno, también afecta a la enseñanza que se le ha de impartir, ya que el profesorado no sólo debe preocuparse de que el alumno adquiera el conocimiento que se imparte en las materias, sino que también debe preocuparse porque el alumno desarrolle las competencias y habilidades que se requieren en el marco de su materia.

Este artículo se centra en el desarrollo y adquisición de competencias en el ámbito docente. Antes de profundizar en ello, se ha de establecer exactamente qué se espera y se entiende por desarrollo y adquisición de competencias y por ende, qué se entiende por competencia. El término competencia ha sido definido de diferentes maneras por distintos autores. Así, [1] define este concepto como el “conjunto de conocimientos y habilidades que los sujetos necesitamos para desarrollar algún tipo de actividad”. En [2] se define competencia como “el conjunto de conocimientos, habilidades y aptitudes necesarios para desempeñar una ocupación dada y la capacidad de movilizar y aplicar estos recursos en un entorno determinado, para producir un resultado definido”. En este sentido, las competencias son capacidades individuales que permiten realizar tareas u obtener ciertos logros de forma eficiente y eficaz. Sin embargo, la mayoría de las definiciones de competencia no se limitan al conjunto de habilidades o destrezas requeridas para desempeñar tareas adecuadamente en un determinado contexto, sino que también involucran una combinación de atributos con respecto al *saber, saber hacer, saber estar y saber ser* [6]. Las competencias poseen un

fuerte fundamento teórico psicológico, integrando procesos cognitivos y afectivos e involucrando la motivación.

El proceso de adaptación al EEES ha requerido la reconversión de las titulaciones de las universidades españolas para homogeneizar la duración de éstas y favorecer el intercambio entre los estudiantes y profesores de países europeos. De esta reconversión han surgido los nuevos títulos de grado, para los que se ha establecido una duración de 4 años, y en los que entre otras novedades, se han incorporado las competencias como parte de la consecución de objetivos del alumno, tanto las que se deben desarrollar en el marco de la titulación, como las específicas que se desarrollan en cada una de las materias que cursan. Concretamente dichas competencias se han clasificado en dos tipos: *competencias específicas* y *competencias generales*. Las competencias específicas son aquellas que se definen específicamente para un área de conocimiento y normalmente están asociadas a la adquisición y desarrollo de conocimientos. Mientras que las competencias generales son competencias transversales al plan de estudios, y por tanto, a todas las asignaturas que lo componen (aunque esto no significa que cada asignatura deba cubrir todas las competencias generales).

Por tanto, hoy en día, las universidades se enfrentan a un nuevo *modelo educativo basado en competencias*. El papel de los docentes dentro de este nuevo modelo no es tanto enseñar (explicar y examinar) unos conocimientos que tendrán una vigencia limitada y que actualmente son altamente accesibles a través de Internet, sino ayudar a los estudiantes a “aprender a aprender” de manera autónoma y promover su desarrollo cognitivo y personal. El docente debe promover el desarrollo de competencias generales de sus alumnos. Ambos deben poner su esfuerzo para que cada alumno desarrolle unas capacidades generales: personales, instrumentales e interpersonales, habilidades y destrezas profesionales que hagan del alumno un buen profesional en el futuro. Esto resulta de vital importancia, ya que hoy en día, el gran volumen de titulados universitarios en el mercado laboral hace que las competencias generales sean uno de los factores o criterios que utilizan las empresas para la incorporación de los titulados a su plantilla laboral. Esto también se demuestra en el hecho de que las empresas manifiestan, cada día más, un gran

interés por las competencias personales mediante la creciente demanda de cursos de habilidades dentro de los planes de formación para sus empleados. Por ejemplo, aptitud para el trabajo en equipo, iniciativa, técnicas de liderazgo, resolución de conflictos, técnicas de negociación, técnicas de comunicación, etc. La adquisición de determinadas competencias proporciona mayores oportunidades de éxito en el cumplimiento de los objetivos profesionales. Esta importancia de las competencias generales a nivel empresarial, todavía tiene una mayor repercusión en el área de la informática y concretamente en el área de la ingeniería del software, ya que existe tanto una gran oferta y demanda en el mercado, como una gran competencia laboral.

El objetivo de este trabajo ha sido el de analizar cómo los docentes estaban promoviendo y evaluando las competencias generales, para así detectar carencias y necesidades que permitan adoptar soluciones al respecto y mejorar la calidad de la docencia. Concretamente, en este artículo se han analizado las actividades y/o aptitudes desarrolladas por el profesorado de la titulación de Graduado en Ingeniería del Software de la Universidad Politécnica de Madrid (UPM) para cubrir las competencias generales.

La estructura del trabajo es la siguiente: El apartado 2 presenta el análisis realizado de la incorporación de competencias generales mediante la identificación de los objetivos de las competencias, la definición actividades para su consecución y el diseño de una encuesta para realizar una análisis empírico. En el apartado 3 se analizan los resultados obtenidos de la encuesta. Finalmente, en el apartado 4 se presentan las conclusiones y trabajos futuros.

2. Análisis de la Incorporación de Competencias Generales en los nuevos títulos de grado enmarcados en el EEES

El objetivo de este trabajo es analizar cómo y en qué medida, el personal docente promueve las competencias generales. Los docentes se deben plantear cuestiones que les permitan incorporar estrategias didácticas para poder cubrir las competencias generales. Algunas de estas cuestiones son:

- ¿Qué tipo de capacidades/aptitudes debe desarrollar un docente para transmitir a los alum-

nos unas determinadas competencias?

- ¿Qué tipo de actividades podrían ayudar en clase a desarrollar esas competencias?
- ¿Se están transmitiendo bien las competencias a los alumnos?
- ¿Cómo evaluar las competencias? ¿Cómo se puede evaluar el grado de satisfacción de esas competencias?

Las competencias generales se han de aprender en la acción. Por lo tanto los docentes y la institución universitaria deben propiciar situaciones de aprendizaje y proporcionar guías que clarifiquen dónde y cómo el estudiante puede incrementar o mejorar sus competencias generales [5]. El primer paso consiste en establecer las competencias generales desarrolladas en cada asignatura del plan de estudios. Por lo tanto, las preguntas arriba definidas se pueden refinar para formular dos cuestiones básicas a las que el profesor debe dar respuesta:

- ¿Cómo contribuye mi asignatura al desarrollo de las competencias generales y específicas?
- ¿Qué competencias estoy logrando con las técnicas utilizadas para la impartición de los contenidos que incluye mi asignatura?

Pero ¿cómo se articulan las competencias generales con las específicas, propias de la asignatura? A partir de las competencias definidas para una asignatura, es necesario considerar el tipo de contenido, metodologías y formas de evaluación que se van a desarrollar. Este paso es básico para lograr con éxito el modelo educativo por competencias:

- Para el logro de las competencias, ¿qué me propongo?
- ¿Qué contenidos, dinámicas de aula y formas de evaluación son las más adecuadas?

Mediante la articulación de esta última cuestión, el proceso de enseñanza basado en competencias alcanza una madurez considerable, pues obliga a la reflexión pedagógica y didáctica del profesor. Esta reflexión está relacionada con los cambios en las metodologías de enseñanza que utiliza el profesor universitario, transformando el modelo tradicional de *metodologías transmisoras a metodologías centradas en el alumno y en su proceso de aprendizaje* [5]. El modelo educativo basado en competencias implica buscar y usar metodologías de aprendizaje innovadoras que incorporen nuevas experiencias

en métodos y técnicas de enseñanza-aprendizaje y prácticas formativas. Entre éstas destacamos las siguientes: proceso de enseñanza-aprendizaje basado en problemas, estudio de casos, proyectos compartidos, trabajo en equipo, análisis y diagnóstico de necesidades, expectativas y conflictos, talleres y seminarios, experiencias y planes en empresas, laboratorio, etc. También son necesarias técnicas de retroalimentación que permitan obtener información útil de los alumnos para la mejora continua de las técnicas empleadas en el aula.

Este conocimiento es clave para analizar el desarrollo de competencias en el marco de un plan de estudios. Es por esta razón por la que en las siguientes subsecciones se analiza en detalle las competencias generales del Título de Graduado de Ingeniería del Software de la UPM y las actividades planteadas para su consecución.

2.1. Competencias Generales del Título de Graduado de Ing. del Software de la UPM

El Plan de Estudios de la titulación de Graduado en Ingeniería del Software por la Escuela Universitaria de Informática define una serie de competencias generales acordadas por la UPM, y que están especificadas por el Real Decreto y el acuerdo del Consejo de Universidades de fecha 3 de marzo de 2009¹ (ver Cuadro 1).

2.2. Actividades asociadas a las Competencias Generales

Para analizar el grado de consecución y desarrollo de las competencias generales del Título de Graduado de Ingeniería del Software de la UPM (ver subsección anterior), es necesario conocer qué actividades se pueden desarrollar en el marco de cada competencia. Sin embargo, resultaba difícil definir las actividades asociadas a cada competencia debido a la generalidad del título. Por esta razón, fue necesario profundizar en cada una de las competencias para conocer qué objetivos comprendía cada una. La identificación de dichos objetivos se realizó a

¹“Consejo de Universidades por el que se establecen recomendaciones para la propuesta por las universidades de memorias de solicitud de títulos oficiales en los ámbitos de la Ing. Informática, Ing. Técnica Informática e Ing. Química” de fecha 8 de junio de 2009 <http://www.boe.es/boe/dias/2009/08/04/pdfs/BOE-A-2009-12977.pdf>

Cuadro 1: Competencias generales

INTRUMENTALES G1. Capacidad de análisis y síntesis G2. Capacidad de organización y planificación G3. Comunicación oral y escrita en la lengua nativa G4. Uso de lengua inglesa G5. Uso de las tecnologías de la información y las comunicaciones G6. Resolución de problemas
PERSONALES G7. Trabajo en equipo G8. Reconocimiento a la diversidad y la multiculturalidad
SISTÉMICAS G9. Razonamiento crítico G10. Aprendizaje autónomo, adaptación a nuevas situaciones y motivación por el desarrollo profesional permanente G11. Creatividad G12. Liderazgo de equipos G13. Iniciativa y capacidad emprendedora G14. Motivación por la calidad G15. Respeto por el medioambiente

partir de los trabajos de [1, 3, 4, 5]. El Cuadro 2 describe las competencias generales del Título de Graduado de Ingeniería del Software de la UPM y los correspondientes objetivos que permiten su consecución. Una vez definidos los objetivos de cada competencia general, fue posible definir una serie de actividades que satisficieran el desarrollo y consecución de dichos objetivos, y por ende de sus correspondientes competencias generales. El resultado que obtuvimos de dicho trabajo se muestra en el Cuadro 3, el cual relaciona las actividades con las competencias generales que se pretenden desarrollar y promover con las mismas.

2.3. Encuesta

A partir del análisis de actividades que favorecen la consecución de cada competencia, se realizó una encuesta para detectar el grado de satisfacción de dichas competencias, carencias, necesidades, etc. Dicha encuesta se realizó a una muestra de 11 profesores de la E. U. Informática de la UPM que imparten docencia en diferentes asignaturas del Título de Graduado en Ingeniería del Software, quienes se ofrecieron voluntarios. La encuesta consta de 35 preguntas². Cada pregunta cubría una serie de competencias generales y todas las competencias fueron analizadas a través de las preguntas. Cabe destacar

²<https://syst.eui.upm.es/sites/default/files/JENUI2011anexo.pdf>

Cuadro 2: Objetivos de las competencias generales

G1. Capacidad de análisis y síntesis: Lectura comprensiva. Subrayado / esquema. Crítica. Resumen.
G2. Capacidad de organización y planificación: Planificación a largo plazo / Planificación a corto plazo. Gestión de recursos, estimación. Control y monitorización.
G3.a) Comunicación escrita en la lengua nativa: Riqueza de vocabulario. Empleo de tecnicismos. Estructuración clara y correcta.
G3.b) Comunicación oral en la lengua nativa: Velocidad. Entonación. Volumen. Pausas. Contacto visual. Vocalización. Postura corporal. Gestos de apoyo, ilustrativos, etc.
G4. Uso de lengua inglesa: Lectura, escritura y conversación.
G5. Uso de las tecnologías de la información y las comunicaciones: Uso de recursos web, moodle, software de edición (textos, hojas de cálculo, presentaciones, etc.).
G6. Resolución de problemas: Comprender el problema. Trazar un plan para resolverlo. Poner en práctica el plan. Comprobar los resultados. Técnicas: Ensayo-error, resolver un problema semejante más sencillo. Experimentar. Simplificar. Experimentar y extraer pautas (inducir). Resolver problemas por analogía. Utilizar un método de expresión adecuado mediante esquemas, tablas, modelos, etc.
G7. Trabajo en equipo: Escucha activa. Empatía. Argumentación. Resolución de conflicto.
G8. Reconocimiento a la diversidad y la multiculturalidad: Comprensión y aceptación de la diversidad interindividual (sexo, capacidad física o intelectual), lingüística, social y cultural (religión, ideología o etnia) como un componente enriquecedor personal y colectivo para desarrollar la convivencia entre las personas.
G9. Razonamiento crítico: Reflexionar. Opinar. Argumentar.
G10. Aprendizaje autónomo, adaptación a nuevas situaciones y motivación por el desarrollo profesional permanente
G11. Creatividad: Capacidad para dar respuestas variadas y novedosas a una propuesta o problema dado. Realización de asociaciones con sus conocimientos previos, transformando lo conocido, combina conceptos, objetos o fenómenos para crear otros nuevos.
G12. Liderazgo de equipos
G13. Iniciativa y capacidad emprendedora
G14. Motivación por la calidad: Orientación hacia la mejora del propio aprendizaje con el fin de afrontar eficazmente problemas y/o dificultades en el ámbito educativo y lograr la formación integral de la persona. Interés por ampliar los contenidos programados.
G15. Respeto por el medioambiente

que las preguntas están formuladas en términos de las actividades que se deberían llevar a cabo en el aula para desarrollar las competencias generales establecidas (ver Cuadro 3) y las respuestas se realizaban en los siguientes términos: 0- Nada de acuerdo, 1- Poco de acuerdo, 2- Bastante de acuerdo, y 3- Completamente de acuerdo.

Cuadro 3: Definición de actividades asociadas a competencias generales

Actividades	Competencias
Clases teóricas: sesiones expositivas, explicativas y/o demostrativas de contenidos.	G1
Trabajos/proyectos en grupo: resolución de problemas, análisis de datos, investigaciones, etc. (el éxito de cada estudiante dependerá de que el conjunto hayan alcanzado las metas fijadas, de manera que los objetivos no son individuales sino grupales). Evaluación final o mediante hitos.	G2, G7, G8, G11, G12
Estudio y trabajo autónomo: actividades individuales que realiza el alumno como preparación de exámenes, lecturas, etc.	G10
Exposición de trabajos en el aula en lengua nativa. Retroalimentación a los alumnos (se presupone de contenido) pero también de comunicación oral/corporal etc.	G3.b
Exposición de trabajos en el aula en lengua inglesa. Conversación. Transparencias en inglés	G4
Entrega de informes. Seminario sobre cómo escribir informes	G3.a, G5
Trabajo de recopilación de información (estado del arte de un tema en particular)	G1, G10
Proposición de temas y reuniones de ideas. Defensa y debate de las ideas ante un grupo (argumentación, escucha, empatía, etc.) Resolución de conflictos.	G3.b, G7, G8, G9
Valoración de exposiciones de otros compañeros (actitud/análisis crítico).	G9
Planificación de tareas de un problema de una duración media/larga.	G2
Estudio de casos.	G6, G11
Proceso de enseñanza-aprendizaje basado en problemas.	G6
Talleres y seminarios: sesiones monográficas supervisadas con participación compartida. Intercambio de experiencia, crítica, diálogo y participación activa de los asistentes.	G3.b, G10, G13
Experiencias y planes en empresas	G13
Clases prácticas: aplicación de los conocimientos a situaciones concretas y adquisición de habilidades básicas y procedimientos: prácticas de laboratorio, de campo, resolución de problemas, etc.	G6, G11
Uso de recursos didácticos haciendo uso de TICs	G5
Uso de recursos de búsqueda bibliográfica (safari, IEEE, ACM, etc.)	G5, G10
Tutorías: atención personalizada para facilitar y orientar a los alumnos en el proceso formativo (consultas, revisión de trabajos en desarrollo, resolución de problemas de comprensión, supervisión de prácticas en empresa, orientación en la elección de asignaturas optativas, etc.)	G14
Durante la realización de trabajo largos. Hitos de entrega. Supervisión.	G2
Tutorías resolución de conflictos trabajo en grupos	G7

3. Análisis de Resultados

Esta sección presenta el análisis de los resultados obtenidos en la encuesta realizada. Las Figuras 1, 2 y 3 muestran a través de tres gráficos la media del grado en el que las competencias instrumentales, personales y sistémicas, respectivamente, son cubiertas por las asignaturas que han participado en la encuesta. A partir de los gráficos, es posible deducir que las competencias que han sido cubiertas en mayor medida son aquellas relacionadas con la capacidad de análisis y síntesis, la resolución de problemas, el trabajo en equipo, el razonamiento crítico y el aprendizaje autónomo. Mientras que las menos cubiertas son aquellas que tienen que ver con

el uso de idiomas tanto los nativos como extranjeros, concretamente lengua inglesa, el reconocimiento a la diversidad y a la multiculturalidad, así como, las competencias relacionadas con la iniciativa y capacidad emprendedora. La Figura 4 muestra un gráfico que refleja las mismas deficiencias. En este caso el gráfico expresa el grado en el que un conjunto de 7 asignaturas, impartidas por los profesores que participaron en la encuesta, cubren las competencias generales. Se observa como la competencia del respeto por el medio ambiente no está cubierta por ninguna de las asignaturas, y de nuevo la comunicación oral y escrita, el uso de la lengua inglesa y el reconocimiento de la multiculturalidad tienen índices más bajos que el resto de competencias.

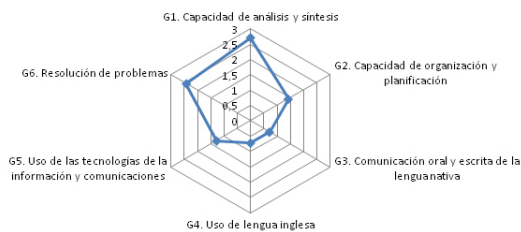


Figura 1: Capacidades Instrumentales

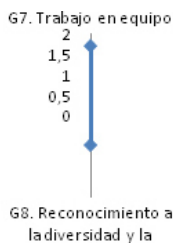


Figura 2: Capacidades Personales

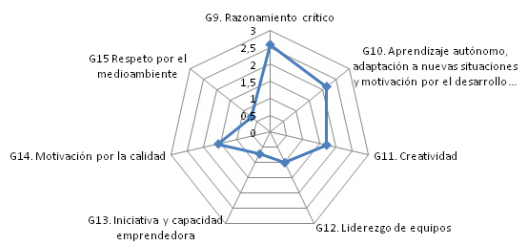


Figura 3: Capacidades Sistémicas

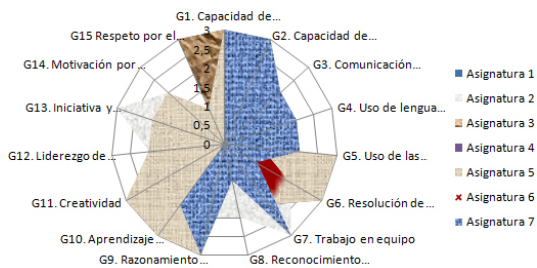


Figura 4: Competencias Generales

A continuación, se muestran los resultados en detalle para un subconjunto de competencias generales. El eje de abscisas de los gráficos de las Figuras 5–10 representa el conjunto de preguntas

directamente relacionadas con la competencia a analizar; el eje de coordenadas representa la puntuación a las preguntas de la encuesta, que como ya se ha definido en la sección 2.3 han sido valoradas en un rango de 0–3. La Figura 5 muestra que, para la competencia general *G1:Capacidad de análisis y síntesis*, la media de las respuestas del total de profesores encuestados, únicamente para aquellas preguntas que cubren la competencia G1 (eje de abscisas). El gráfico muestra que la media de implantación de la competencia G1 es satisfactoria, ya que los resultados de las respuestas de los profesores alcanzan una puntuación de 2,5 (sobre 3).

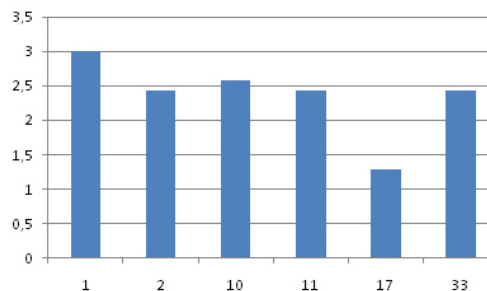


Figura 5: Competencia G1

Otras competencias generales muestran también el mismo nivel satisfactorio en cuanto a su implantación en la docencia. Los resultados de las respuestas de los profesores de las competencias *G2:Capacidad de organización y planificación* (ver Figura 6), *G9:Razonamiento crítico* (ver Figura 7) y *G10:Trabajo autónomo* (ver Figura 8) alcanzan una puntuación de 2 (sobre 3). Al igual que estas competencias, la competencia *G6:Resolución de problemas* ha obtenido unos buenos resultados en la encuesta, pero se ha de matizar que se ha detectado que es necesario mejorar la aplicación de la técnica de estudio de casos y/o simulaciones para potenciar la integración de la teoría y la práctica.

Sin embargo, la Figura 9 muestra que la media de implantación de la competencia *G3:Comunicación oral y escrita en lengua nativa* presenta algunos puntos débiles, ya que, p.e., la escucha, la expresión oral y comunicación oral en forma de debates no se promueven. Aunque se fomenta la exposición de trabajos en el aula ante el resto de compañeros, no se proporcionan pautas para realizar tales exposiciones,

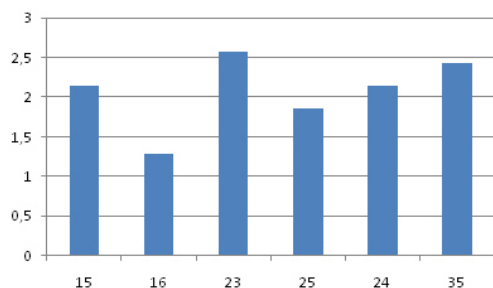


Figura 6: Competencia G2

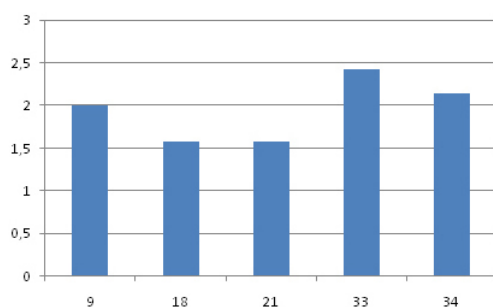


Figura 7: Competencia G9

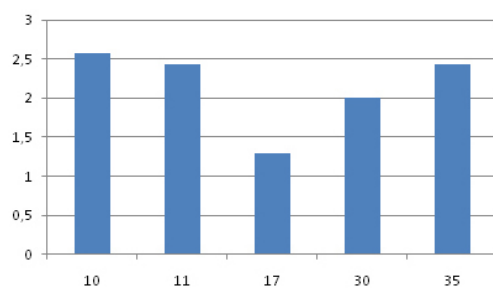


Figura 8: Competencia G10

ni se proporciona retroalimentación a los alumnos sobre cómo han ido las exposiciones, no sólo en cuanto a contenidos, sino en cuanto a la calidad de la técnica expositiva.

La Figura 10 muestra que la media de la competencia *G4: Uso de la lengua inglesa* está poco aplicada. Únicamente se promueve la lectura de texto en lengua inglesa, mientras que las exposiciones, redacciones y conversaciones en inglés no están contempladas.

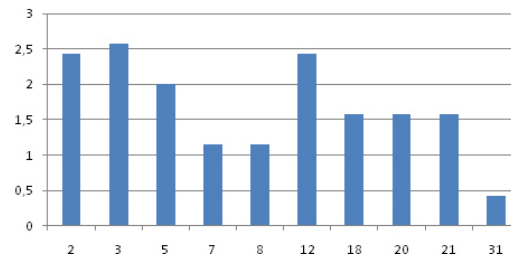


Figura 9: Competencia G3

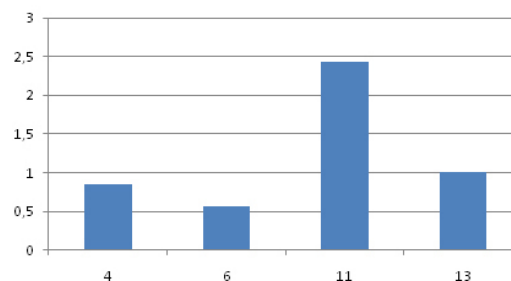


Figura 10: Competencia G4

Finalmente, existe un grupo de competencias que se satisfacen de forma moderada. Estas son las competencias *G5: Uso de las TICs*, *G7: Trabajo en equipo*, y *G14: Motivación por la calidad*. Concretamente, el análisis de la competencia *G7* deja de manifiesto algunos puntos aún por cubrir, como por ejemplo la resolución de conflictos, o el fomento de objetivos grupales y no individuales. Así mismo, el análisis de la competencia *G14* muestra la existencia de importantes puntos débiles, por ejemplo el hecho de que no se enseñan pautas y/o recomendaciones para la realización de informes de calidad, ni se enseñan pautas para realizar exposiciones claras y efectivas. Además, una vez realizados los informes o exposiciones, apenas existe retroalimentación.

4. Conclusiones

Las nuevas titulaciones de grado dentro del EEES han de seguir un modelo educativo basado en competencias. Por ello, el docente no sólo debe preocuparse de que el alumno adquiera el conocimiento que se imparte en la materia, sino que también debe preocuparse de que el alumno desarrolle las compe-

tencias y habilidades que se le requieren en el marco de su materia. Con el objetivo de proporcionar una guía al docente sobre qué mejorar, para tratar el mayor número de competencias satisfactoriamente, se ha realizado un análisis empírico sobre el desarrollo y consecución de competencias en la Titulación de Graduado de Ingeniería de Software de la E.U. Informática de la UPM. De este modo ha sido posible detectar aquellas competencias que sufren de carencias, mejoras y necesidades.

Este análisis ha constado de cuatro etapas: (i) Identificación de objetivos específicos de cada competencia general, (ii) Definición de actividades a realizar para la consecución satisfactoria de competencias, (iii) Realización de una encuesta al profesorado para detectar a través de las actividades que competencias son implantadas y en qué medida, y (iv) Análisis de los resultados de la encuesta realizada.

De este análisis es posible concluir que hay competencias generales que a día de hoy se satisfacen plenamente como son: la capacidad de análisis, la capacidad de organización y planificación, el razonamiento crítico y el trabajo autónomo. Sin embargo, existen competencias que se deben mejorar como son: la comunicación oral y escrita en lengua nativa, el uso de la lengua inglesa, el uso de las TICs, la motivación por la calidad y el trabajo en equipo. Por tanto, es posible concluir que hay una clara tendencia al abandono de la clase magistral clásica, pero que todavía existen muchos aspectos a mejorar en la forma de impartir las materias, para que los alumnos consigan la consecución de competencias.

Este trabajo ha sido la base para definir las directrices de un Proyecto de Innovación Educativa que comienza su implantación en el segundo cuatrimestre del curso académico 2010-2011 y que será el marco en el que se seguirá trabajando para una mejora en el desarrollo y evaluación de competencias generales del alumnado de la Titulación de Graduado en Ingeniería del Software de la E.U. Informática de la UPM.

Agradecimientos

Los autores agradecen el esfuerzo de P.P. Alarcón, P. Fernández, J. Garbajosa, J.L. Sánchez, y J. Tejedor por su colaboración en este trabajo, el cual ha sido parcialmente financiado por el Programa de Innovación Educativa 2010/2011 de la Universidad Politécnica de Madrid (proyecto Aprendizaje Ágil, *Agile Learning*) y por el programa de Personal Investigador en Formación de la Universidad Politécnica de Madrid. Finalmente, agradecer al Instituto de Ciencias de la Educación de la Universidad Politécnica de Madrid, que a través de sus cursos de Formación del Profesorado en el marco del EEES, proporciona la base formativa para el desarrollo de la profesión docente.

Referencias

- [1] Zabalza Beraza y Miguel Ángel, *Competencias docentes del profesorado universitario: calidad y desarrollo profesional*. 2 Edición, Narcea, S.A. de Ediciones, 2003..
- [2] Yániz, C. y Villardón, L., *Planificar desde competencias para promover el aprendizaje. El reto de la sociedad del conocimiento para el profesorado universitario*, Universidad de Deusto, 2006.
- [3] Rodríguez A., Lozano C. y Ochoa de Eribe, *Aproximación al logro de una metodología para el desarrollo de un proceso enseñanza-aprendizaje a través de competencias*, Encuentro Anual Virtual Educa, 2009.
- [4] Goñi, J.M., *El espacio europeo de Educación Superior, un reto para la Universidad. Competencias, tareas y evaluación, los ejes del currículum universitario*, Ed. Octaedro, 2005.
- [5] Bozu, Z. y Canto, P.J., *El profesorado universitario en la sociedad del conocimiento: competencias profesionales docentes*, Revista de Formación e Innovación Educativa Universitaria, vol 2 n 2, 2006, pp. 87-97
- [6] Proyecto Tunning (2003). *Tunning Educational Structure in Europe. Informe final*. Bilbao (España): Universidad de Deusto., 2003.

En busca del estudiante perdido: caracterización de los ‘no presentados’

Ismael Sanz, María J. Aramburu, Lledó Museros, María Pérez, Carmen Barrachina

Departamento de Ingeniería y Ciencia de los Computadores

Universitat Jaume I

Avg. Vicent Sos Baynat s/n

12071 Castellón de la Plana

{i.sanz,aramburu,museros,mcatalan,carmen.barrachina}@uji.es

Resumen

El abandono de asignaturas por parte de los estudiantes es un problema cada vez más importante. Este trabajo describe una metodología y una herramienta informática para entender qué lleva a los estudiantes a abandonar una asignatura, y así poder realizar acciones preventivas y paliativas.

Nuestro objetivo inicial era ser capaces de determinar empíricamente unos criterios de clasificación que permitieran dar pistas a priori sobre el riesgo de abandono de cada estudiante matriculado, y así poder poner en marcha medidas preventivas desde el principio del curso. Se han realizado pruebas piloto en asignaturas del ámbito de las Bases de Datos, con resultados prometedores.

El trabajo realizado nos condujo al desarrollo inicial de un sistema de información integrado, con la flexibilidad suficiente para servir de apoyo a futuras tareas de tomas de decisiones relativas a la gestión de la docencia.

Palabras clave

Abandono estudiantil, sistema de apoyo a la decisión.

1. Introducción

1.1. Contexto y motivación

La motivación inicial del presente trabajo surgió de la experiencia acumulada durante las realización de tres proyectos de mejora educativa, dos de ellos asociados a la asignatura Bases de datos avanzadas (optativa de cuarto curso de Ingeniería Informática y de tercer curso de Ingeniería Técnica en Informática de Gestión), y el otro asociado a la asignatura

Bases de datos (obligatoria de segundo curso de Ingeniería Técnica en Informática de Gestión en modalidad semipresencial). Estos proyectos consistieron globalmente en la implementación de una nueva metodología de enseñanza centrada en el estudiante y basada en un sistema de evaluación continua. Todos los proyectos alcanzaron su objetivo principal, que era mejorar el número de aprobados, de manera incluso más satisfactoria de lo esperado. Sin embargo no cambió en absoluto, e incluso empeoró, otra de las variables que nos habíamos propuesto mejorar: el número de estudiantes presentados.

Desde nuestro punto de vista, habíamos conseguido que quien llevara la asignatura al día superara la asignatura, pero al precio de una tasa alta de abandonos. Esto se podría achacar a la nueva metodología que se implantó. Sin embargo, entendemos que la situación no es tan simple: cada curso encontramos estudiantes matriculados que no llegan a tener ningún contacto con la asignatura (no consta un solo acceso al curso correspondiente en el Aula Virtual, no asisten ni siquiera a la clase inicial de presentación, ni se presentan a los exámenes). Naturalmente, entre estos dos extremos (estudiantes que por cualquier circunstancia abandonan la asignatura tras intentarlo, y estudiantes que simplemente no aparecen) existe toda una serie de casuísticas intermedias.

Como un último intento de paliar esta situación, en la asignatura Bases de Datos Avanzadas incluimos un sistema de evaluación adicional. Consistía en un itinerario específico para estudiantes que no pudieran realizar los entregables realizados en clase, bajo la suposición implícita de que la imposibilidad de asistir a las clases presenciales era el motivo principal de las cifras de abandono; hay que tener en cuenta que por las características de estas asignaturas muchos de los estudiantes matriculados están realizando simultáneamente su estancia en prácticas

o trabajando en una empresa. Los estudiantes que eligieran este itinerario debían realizar unas tareas previamente programadas, disponían de tutorías físicas (en el despacho del profesor) y virtuales (por e-mail o a través del Aula Virtual), y tenían la obligación de presentarse al examen final de la asignatura. Sin embargo, la experiencia ha mostrado que este itinerario no es seguido por prácticamente ningún estudiante, y que muchos estudiantes que encuentran trabajo o están haciendo la estancia en prácticas no tienen problema alguno en seguir la evaluación continua.

Debía haber, pues, otras causas subyacentes que desconocíamos. Conversaciones informales con los estudiantes y con profesores en cuyas asignaturas ocurre el mismo problema nos convencieron de que no podíamos plantear nuevas propuestas de soluciones paliativas sin antes descubrir esas causas subyacentes. Nuestra sensación era que muchos estudiantes abandonaban nuestras asignaturas y no sabíamos por qué, ni podíamos hacer nada para evitarlo, lo que no nos parecía aceptable. En una búsqueda bibliográfica en algunos de los principales foros de investigación en innovación educativa encontramos pocos trabajos que arrojaran luz sobre los “no presentados”, que se interpretan más bien como un hecho inevitable – algo con lo que no podemos estar de acuerdo sin una completa investigación previa que no parece haberse realizado hasta ahora [2][6].

Decidimos que una buena manera de abordar esta investigación era recopilar datos suficientes para permitir estudiar el problema utilizando herramientas informáticas de análisis multidimensional de datos. Se trata de herramientas de inteligencia de negocio de amplia utilización para el apoyo de decisiones en entornos empresariales y financieros, hemos considerado que sería muy interesante utilizarlas para analizar nuestra problemática y de esta manera realizar una investigación mucho más completa que la que permiten otras herramientas más simples pero de uso mucho más habitual, como pueden ser las hojas de cálculo. Además, disponer de una herramienta con estas características podría llegar a ser muy útil para múltiples tareas relacionadas con la docencia.

Una aproximación similar había sido ya ensayada por [7], que mostró la utilidad de aplicar análisis estadísticos sobre información almacenada en un *Data Warehouse* específicamente construido para estudiar el absentismo en la Universidad Industrial de San-

tander. Este trabajo se basaba exclusivamente en la información disponible en el sistema de información universitario, e ignoraba otros factores referentes a la situación personal y las percepciones subjetivas de los estudiantes que, sospechábamos, resultaban decisivos en el abandono o no de una asignatura. Como veremos, esta hipótesis de trabajo se ha visto confirmada por los resultados obtenidos.

Para asegurarnos de que nuestros resultados fueran lo más generalizables posible, decidimos realizar un trabajo coordinado entre tres asignaturas que comparten área de conocimiento (las bases de datos), pero que se imparten de manera muy diferente: *Bases de datos* (IG18S) es una asignatura semipresencial y armonizada con el EEES; *Bases de datos avanzadas* (IG52/II52) se evalúa, como se ha comentado, de forma continua, mientras que *Sistemas de información inteligentes* (II73) está basada en clases magistrales y se evalúa en un examen final. Por tanto, de esta manera podemos tener en cuenta la manera de impartir la asignatura como un factor más de nuestro análisis.

1.2. Objetivos

El objetivo general de nuestro trabajo es obtener información sobre los estudiantes que abandonan las asignaturas, para ser capaces de entender qué les lleva a ello y poder realizar acciones preventivas y paliativas en el futuro. Según nuestra experiencia y la de otros trabajos relacionados [2][6], entendemos que es muy improbable que todos los casos respondan a las mismas causas, y que es muy importante que seamos capaces de descubrir estas causas para entender mejor a nuestros estudiantes y así favorecer su progresión. Por esta razón, nuestro método consiste en la implementación de herramientas para intentar *agrupar* los estudiantes que comparten ciertas características comunes y así *caracterizar* cada uno de esos distintos “grupos de riesgo”, que probablemente necesitarán soluciones específicas. De esta manera sería posible determinar unos criterios de clasificación que permitan determinar *a priori* el riesgo de abandono de un estudiante matriculado, y poder así poner en marcha medidas preventivas desde el principio del curso.

Por tanto, los objetivos de este trabajo pueden desglosarse de la siguiente manera:

- La recopilación de un conjunto de datos que

incluyan las características potencialmente relevantes de los estudiantes matriculados en las asignaturas objeto del proyecto (con datos sobre su situación laboral, académica y personal), debidamente mantenidos en soporte electrónico en forma de base de datos, y pre-procesados para respetar el anonimato de los estudiantes (anonimización).

- La implementación y aplicación de un conjunto de herramientas de análisis de esos datos que nos permita (i) caracterizar a los estudiantes que han abandonado la asignatura, encontrando características que los separen de los que han continuado hasta el final (ii) encontrar los grupos de estudiantes no presentados que presentan características similares (es decir, los “grupos de riesgo”) y (iii) estudiar si hay diferencias entre las distintas maneras de impartir asignaturas.

Un objetivo adicional, que fue creciendo en importancia conforme avanzábamos en el trabajo, es el diseño y la implementación de un sistema de información integrado de apoyo al docente, y la evaluación de su potencial utilidad. Este sistema integraría toda la información académica de las asignaturas y de los estudiantes matriculados, y proporcionaría herramientas adecuadas para su análisis multidimensional. Como resultado obtendríamos una herramienta de apoyo a la toma de decisiones para docentes y gestores académicos.

1.3. Resultados

Como resultado de este trabajo hemos creado una infraestructura básica consistente en:

1. Un estudio sistemático de las variables que pueden influir en el abandono de una asignatura.
2. Métodos que permiten obtener datos relativos a esas variables; hemos implementado métodos para extraer información de las bases de datos corporativas de la universidad, y también hemos creado una encuesta electrónica para los estudiantes.
3. Técnicas para producir modelos predictivos, que permiten determinar el grado de riesgo de cada estudiante.
4. Una herramienta de análisis multidimensional de datos que permite estudiar los datos des-

de diversas perspectivas y realizar agrupaciones específicas a las necesidades concretas de cada caso de estudio.

Los puntos 2, 3 y 4 constituyen el embrión de un sistema de información de apoyo para el docente.

El resto del artículo se organiza de la siguiente manera: la sección 2 presenta la metodología desarrollada para la consecución de los objetivos anteriormente planteados, en la sección 3 se realiza un análisis de los resultados obtenidos y finalmente en la sección 4 se exponen conclusiones, se repasa el impacto que están teniendo los resultados preliminares obtenidos y se plantean ideas para trabajos futuros.

2. Metodología desarrollada

Nuestra hipótesis inicial de trabajo es que hay dos tipos de factores fundamentales que afectan en la decisión de que un estudiante abandone una asignatura: por un lado, factores relativos a la situación académica de cada estudiante (por ejemplo, el número de créditos superados o el número de convocatorias a que se ha presentado ya) y, por otro, otros relativos a las circunstancias ajenas (por ejemplo, laborales y familiares) y a preferencias personales del estudiante respecto a la asignatura (por ejemplo, el interés que le genere la metodología docente). Deberemos, por tanto, recopilar información sobre ambos aspectos.

Es importante tener en cuenta desde el principio el diferente carácter de ambos tipos de factores: en el primer caso se trata de datos objetivos, de los que podremos disponer de forma exacta y numérica. Sin embargo, los factores personales de los estudiantes son mucho más genéricos y abiertos: preferencias, circunstancias personales, etc. Esto motivará que necesitemos utilizar estrategias de análisis diferenciadas para cada uno de estos factores.

En todo caso, a partir de la hipótesis, será necesario decidir qué información es potencialmente de interés para el análisis, y cómo es posible acceder a ella. Dado el carácter abierto de esta investigación, intentamos determinar un amplio abanico de variables potencialmente interesantes. Como paso inicial se realizaron tres acciones:

- Una búsqueda bibliográfica en foros de innovación educativa. El trabajo más cercano que encontramos fue [6], en el que se describe una

encuesta realizada en la universidad de Oviedo en un proyecto de innovación similar a éste.

- Entrevistas informales con estudiantes (antiguos y actuales) de las asignaturas objeto del proyecto.
- Un estudio de la información disponible sobre el historial académico de cada estudiante en el sistema de información interno de la universidad.

A partir de estos datos, diseñamos la siguiente estrategia:

- Para analizar la información relativa al historial académico de los estudiantes, se aplicaron técnicas de clasificación automática, para intentar determinar la relevancia de cada variable y comprobar si era posible encontrar patrones relevantes para cada asignatura.
- Para analizar las circunstancias personales y las preferencias de cada estudiante, mucho más complejas, optaremos por un análisis exploratorio de los datos, mediante un interfaz gráfico apropiado.

Como algoritmos de clasificación automática se seleccionaron dos métodos sencillos de generación de árboles de decisión: C4.5 [4] y la función genérica *rpart* disponible en el software estadístico R [5], que sigue las técnicas descritas en [1]. Dado que el volumen de datos disponibles no era elevado, no cabía esperar un gran poder predictivo de los modelos obtenidos; sin embargo, este primer paso proporcionaría información muy precisa sobre la relevancia de los factores académicos en cada asignatura.

Para estudiar los factores relativos a las circunstancias y las preferencias personales, diseñamos una encuesta anónima a partir de [6], y usamos el software QlikView [3] para visualizar, explorar y analizar los datos de manera interactiva. La Figura 1 muestra una de las pantallas desarrolladas para el análisis.

2.1. Obtención de los datos

Utilizamos la información recopilada para adaptar el contenido de la encuesta propuesta por [6] y generar nuestra propia encuesta. En resumen, se tuvieron en cuenta factores académicos, indicadores del rendimiento, circunstancias personales, y otros factores inherentes a la asignatura (horarios, relevancia para el estudiante...) y al profesorado.

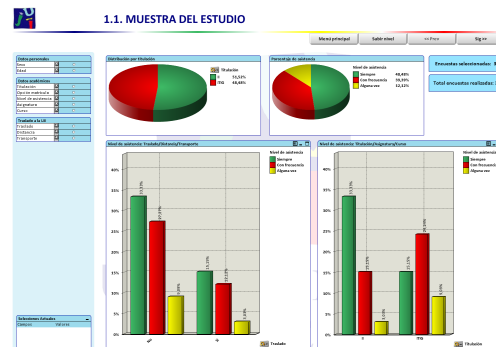


Figura 1: Análisis de resultados en QlikView

Además de diseñar la encuesta, se buscaron otras fuentes de datos de las que podíamos obtener información relevante, a partir de la información disponible en el sistema de información interno de la universidad. En particular, esta información incluye los resultados de cursos anteriores (incluyendo los estudiantes repetidores) y los comentarios de los estudiantes realizados en años anteriores como parte de la evaluación del profesorado.

Una vez determinada qué información debíamos capturar directamente de los estudiantes, en esta fase completamos el diseño de las encuestas a realizar y la automatización de la obtención de datos. Para la implementación de la encuesta consideramos prioritario facilitar lo más posible a los estudiantes su cumplimentación, lo que implicaba los siguientes principios: (1) la encuesta debía ser completamente anónima, para evitar cualquier sospecha y (2) la encuesta debía implementarse mediante una plataforma electrónica tan fácil de usar como fuera posible. El primer principio descartaba el uso del Aula Virtual de la universidad, ya que requiere autenticación. Por ello, realizamos un estudio de varias plataformas de encuestas electrónicas gratuitas disponibles en Internet. Finalmente nos decidimos por *e-encuesta*, ya que proporcionaba la mayor simplicidad de uso a los estudiantes. Se decidió enviar a los estudiantes el enlace tras la finalización de las revisiones de examen de la convocatoria de junio. La figura 2 muestra una pantalla de dicha encuesta.

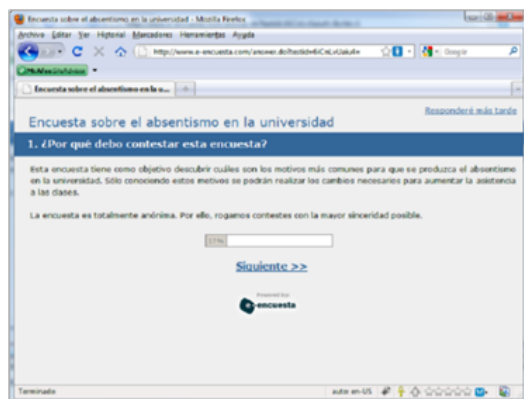


Figura 2: Pantalla de bienvenida de la encuesta

3. Puesta en marcha y resultados

En cuanto a la información obtenida del sistema de información interno de la universidad, para su extracción se realizaron los siguientes pasos para cada asignatura involucrada y para cada curso disponible:

1. Extraer la lista de estudiantes del curso.
2. Eliminar los apartados con información personal y los no relevantes. Los campos considerados relevantes son: *Curso*, *Asignatura*, *Convocatoria*, *Titulación*, *Créditos superados*, *Créditos troncales superados* y *Media del expediente*.
3. Añadir información sobre si el estudiante se ha presentado o no (codificada como presentado=1 y no presentado=0). Esto se obtuvo de: (i) para el curso actual, la información de seguimiento de la asignatura, y (ii) para los cursos anteriores, de las actas.

Los datos obtenidos fueron los siguientes: 80 registros asociados a la asignatura II73 (desde el curso 2005/2006), 184 registros asociados a la asignatura IG52/II52 (desde el curso 2007/2008, en el que cambió el sistema de evaluación de la asignatura) y 59 registros asociados a la asignatura IG18S (desde el curso 2008/2009, curso en el que comenzó a impartirse de manera semipresencial). En total se han utilizado 323 registros.

Una vez obtenida toda esta información se comenzó con su análisis.

3.1. Análisis de datos

Por un lado se analizaron los datos obtenidos sobre la situación académica de los estudiantes, y por otro lado, y de manera paralela, los datos obtenidos con la realización de la encuesta, para lo cual se utilizó la herramienta de análisis multidimensional de datos QlikView.

El objetivo del análisis de los datos provenientes del sistema de información interno de la universidad fue identificar, de entre las variables disponibles, cuáles son relevantes para determinar si un estudiante se ha presentado finalmente o no. Para ello, aplicamos las dos técnicas antes mencionadas, que intentan determinar la relación entre una variable aleatoria y diversos factores. En este caso, la variable objetivo era si el estudiante se presentó o no a la asignatura; los factores son el resto de variables disponibles. A partir de este análisis se obtuvieron los siguientes resultados:

- Modelo predictivos para II73. El particionamiento recursivo obtuvo un modelo muy simple (figura 3a). Si la media del expediente era inferior a 6.485 clasifica al alumno como No presentado, y si la media es igual o superior como presentado. La clasificación es correcta en un 90%. Es notable que la única variable relevante para este caso encontrada por el algoritmo ha sido la media del expediente. Intuitivamente parece un resultado razonable, ya que al tratarse de estudiantes que están prácticamente acabando la carrera hay poca variación en el resto de variables.
- Modelos predictivos para IG52/II52 (figura 3b). Como en el caso anterior, depende básicamente de la media del expediente (siendo aquellos estudiantes con media más alta más probable que se presenten, y aquellos con la media más baja menos probable). Sin embargo, en el rango intermedio de puntuaciones entran en juego también el número de créditos troncales superados. El error de clasificación es alto (25%), lo que apunta al hecho de que hay más factores determinantes además de los estudiados.
- Modelos predictivos para IG18S (figura 3c). La partición recursiva obtiene un modelo en el que la variable principal es la media del expediente y secundariamente se utilizan los datos de créditos superados.

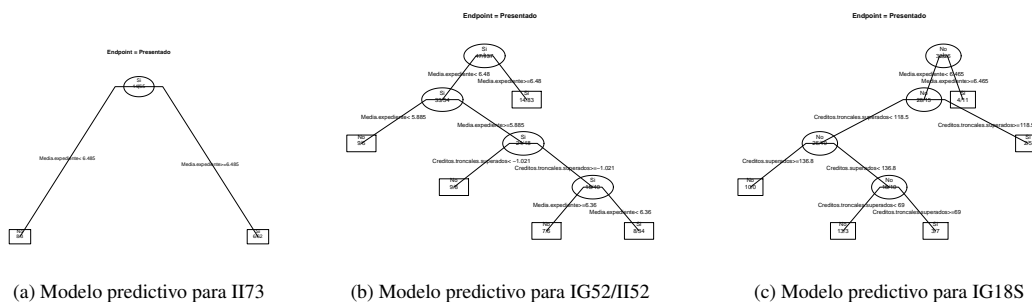


Figura 3: Modelos predictivos desarrollados

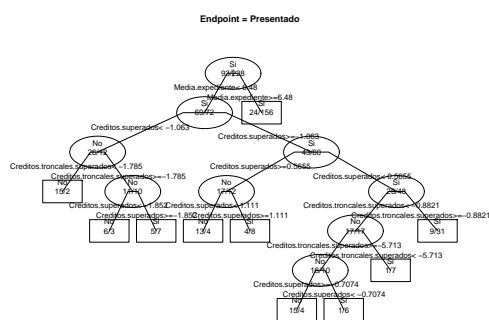


Figura 4: Modelo predictivo para los datos agregados

- Modelos predictivos para los datos agregados (figura 4). A pesar de presentar un error de predicción superior al 30%, son interesantes para mostrar tendencias globales (los datos se han normalizado para limitar las diferencias entre las asignaturas).

A continuación se detallan las conclusiones globales derivadas del análisis de los resultados de las encuestas a través de la herramienta QlikView, resumidas en la figura 5.

En primer lugar, los motivos de absentismo en relación a los horarios de las asignaturas son:

- *Solapamiento de horarios con otras asignaturas*: Una posible solución consistiría en comprobar entre qué asignaturas se producen mayores solapamientos e intentar evitarlos.

- *Muchas clases seguidas*: El hecho de tener muchas clases seguidas sin descanso entre ellas provoca muchas veces que el estudiante decida dejar de asistir a alguna de ellas para poder descansar. Se debería intentar dejar espacios de tiempo entre clases de asignaturas, al menos de las que pertenecen al mismo curso.

- *Muy temprano*: Cabría pensar que esto lo opinen los estudiantes que residen fuera de la ciudad y que deben desplazarse hasta la universidad, requiriendo esto un tiempo determinado. Sin embargo, los que han contestado que las clases son demasiado temprano, en su mayoría residen en la misma ciudad en la que se encuentra la Universidad.

En segundo lugar, los motivos de absentismo más comunes obtenidos de las preguntas abiertas nos dicen que:

- *Organización*: Estos temas son los más problemáticos a la hora de poder asistir a clase, sobre todo lo referente al solapamiento de asignaturas. Se podrían tomar las mismas medidas mencionadas anteriormente.
- *Motivos de salud*: El siguiente motivo más común es el relacionado con motivos de salud, pero en este caso no hay nada que la parte del profesorado puede hacer para solucionar este problema.
- *Solapamiento con el trabajo*: Al igual que antes, a pesar de tratarse de otro motivo común, el profesorado no puede intervenir.
- *Clases aburridas*: En este caso el profesorado sí que puede tomar medidas: intentando impartir clases que sean del interés del estudiante, en las

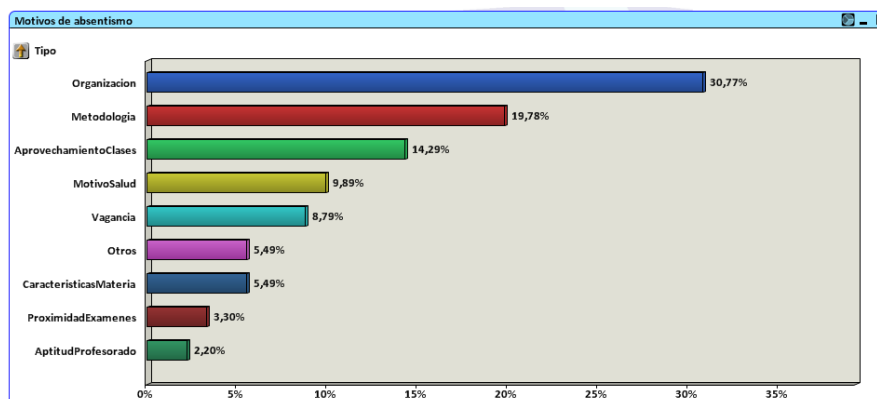


Figura 5: Resumen de los motivos de absentismo manifestados en la encuesta

que éste sea más partícipe, clases más dinámicas en las que se evite entrar en una monotonía que pueda interferir en la atención del alumno.

- *Horario adecuado*: El motivo más popular entre los encuestados hace referencia a tener un horario adecuado de las clases, que no comience demasiado temprano ni acabe demasiado tarde, que no sea en horarios de comida.
- *Puntuar asistencia*: El hecho de premiar la asistencia es importante para los estudiantes, ya que con ello tienen un incentivo extra que les “obligue” a acudir a las clases.
- *Buenas explicaciones*: Otro hecho importante es que el profesorado ofrezcan buenas explicaciones durante sus clases.
- *Ejercicios prácticos*: El mismo nivel de importancia le dan los alumnos a las buenas explicaciones y a la realización de ejercicios prácticos durante las clases.

En cuanto a los motivos más comunes para abandonar una asignatura son los siguientes:

- *Difícil*: El hecho de que la asignatura sea demasiado difícil es el factor más importante a la hora de abandonar la asistencia a una materia. Sería conveniente intentar enfocar las clases de manera que la comprensión de los conceptos de esa asignatura, complicada de por sí, sea más sencilla.
- *Descuelgue*: A veces se produce un descuelgue de la asignatura, debido a la falta de tiempo para llevarla al día, o por la dificultad de ésta.

- *Falta de tiempo, exceso de asignaturas*: Estos dos motivos tienen el mismo nivel de importancia para los encuestados. Es, incluso, posible que estén relacionadas entre ellas, ya que un exceso en la matriculación de asignaturas provocará que el alumno no disponga de suficiente tiempo para atenderlas todas.

Finalmente, en relación a la proporción de presentados en las asignaturas bajo estudio en este trabajo, las conclusiones obtenidas son las siguientes:

- Respecto a *créditos superados*: Los alumnos que más se presentan al examen de alguna de estas asignaturas son los que tienen entre el 51 y el 75 % de créditos superados.
- Respecto a *créditos troncales superados*: Los alumnos que más se presentan al examen son los que tienen entre el 51 % y el 75 % de créditos troncales superados.
- Respecto a *la nota media de expediente*: Los alumnos con notable como nota media del expediente tienen un índice mayor de presentados.
- Respecto a *la titulación y a la asignatura*: Los alumnos se presentan más a la asignatura IG52/II52 que al resto.

Estas últimas conclusiones confirman las conclusiones obtenidas con el análisis de los datos del sistema de información interno a través de la herramienta R.

4. Impacto, conclusiones y trabajo futuro

El primer resultado de este proyecto ha sido la creación de una infraestructura para la gestión integrada de la información referida a la docencia que no está limitada a la aplicación concreta que acabamos de describir. A medio plazo, esperamos convertirla en un completo “Sistema de Análisis de Información Académica” que pueda servir de apoyo para la toma de decisiones tanto a los docentes responsables de las asignaturas como a los coordinadores académicos que gestionan el grado. A tal efecto, nuestra línea de trabajo principal va a estar orientada a la integración de la información disponible en el sistema de información institucional de la universidad, con el objeto de explorar las posibilidades que así se abrirán.

En cuanto al objetivo que motivó inicialmente este trabajo, la caracterización de los no presentados, las técnicas creadas nos han permitido aprender más sobre las causas que llevan a algunos estudiantes a abandonar las asignaturas, refinar los modelos predictivos y el sistema de visualización y análisis mediante QlikView, y aplicar su resultado a la mejora docente. Es importante notar que nuestros resultados indican claramente que cualquier sistema de información útil para el apoyo al docente no puede limitarse a contener sólo información extraída automáticamente del sistema de información institucional, sino que se debe enriquecer en la medida de lo posible con datos estadísticos que sean indicadores de la situación personal y las percepciones subjetivas de los estudiantes, y que dependen de cada materia.

Esto implica necesariamente sistematizar la realización de encuestas a los estudiantes. En la primera aproximación realizada, quisimos imponer el anonimato total en las encuestas, lo que hizo imposible cruzar los datos recogidos con los obtenidos del sistema de información interno de la universidad, limitando enormemente las posibilidades de análisis automático. Por ello, en este curso hemos querido mejorar el sistema de encuestas para poder incrementar los datos disponibles para su análisis automático, limitando las encuestas a los temas más “sensibles” en los que el anonimato tenga sentido.

A pesar de estas limitaciones, los datos obtenidos durante el curso pasado ya han motivado algunos cambios relevantes en la docencia. En la asignatura

Sistemas de información inteligentes (la única que ha acabado en el momento de enviar esta contribución) se han aplicado algunos pequeños cambios en la metodología docente motivados por los resultados de las encuestas, que han redundado en una mejora clara: este curso ningún estudiante ha abandonado la asignatura.

Agradecimientos

Este trabajo se ha realizado gracias al apoyo de la *Unitat de Suport Educatiu* de la Universitat Jaume I (a través de proyectos de mejora educativa en los cursos 2009/10 y 2010/11) y del Departamento de Ingeniería y Ciencia de los Computadores de la Universitat Jaume I, que ha financiado ayudas para Carmen Barrachina e Ismael Sanz a través de su Plan Estratégico.

Referencias

- [1] Breiman, L.; Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). Classification and regression trees. Wadsworth & Brooks/Cole Advanced Books & Software
- [2] Agustín Cernuda del Río, Sonia Hevia Vázquez, María del Carmen Suárez Torrente, Daniel Gayo Avello, “Un estudio sobre el absentismo y el abandono en asignaturas de programación”. XIII Jornadas de Enseñanza Universitaria de la Informática (2007).
- [3] <http://www.qliktech.com>
- [4] Quinlan, J. R. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, 1993.
- [5] <http://www.r-project.org/>
- [6] Rodríguez, R. et al. “El absentismo en la Universidad: resultados de una encuesta sobre motivos que señalan los estudiantes para no asistir a clase”. Aula Abierta, 82, ISSN 0210-277, págs. 117-146 (2003)
- [7] Salazar, A.; Gosálbez, J.; Bosch, I.; Miralles, R.; Vergara, L.; A case study of knowledge discovery on academic achievement, student desertion and student retention. 2nd International Conference on Information Technology: Research and Education, UTRE 2004 (2004).

La mentoría como elemento de mejora en el proceso de enseñanza - aprendizaje

José Antonio Gómez Hernández
Dpto. de Lenguajes y Sistemas Informáticos
Universidad de Granada
C/ Periodista Daniel Saucedo Aranda s/n
18071 Granada
jagomez@ugr.es

Eduardo M. Eisman Cabeza
Dpto. de Ciencias de la Computación e I.A.
Universidad de Granada
C/ Periodista Daniel Saucedo Aranda s/n
18071 Granada
eisman@decsai.ugr.es

Resumen

Este artículo describe una experiencia de mentoría llevada a cabo dentro de la Universidad de Granada con el objetivo de mejorar la calidad de la enseñanza que en ella se ofrece. Se han realizado diferentes actividades (grabaciones en vídeo y seminarios de formación) que han dado lugar a acciones de mejora concretas de la docencia, la tutoría y la evaluación de los alumnos. A tenor de los resultados de diferentes encuestas, la experiencia ha sido muy positiva no sólo para el profesor mentor y el profesor novel, sino también para sus alumnos.

Summary

This paper describes a mentoring experience carried out in the University of Granada with the aim of improving the quality of the education that this offers. Different activities have been performed (video recording and training seminars) which have given rise to concrete measures for the improvement of the teaching, the tutoring, and the evaluation of the students. According to the results of different surveys, the experience has been very positive not only for the mentor and the beginning teacher, but also for their students.

Palabras clave

Mentoría, grabaciones, tutoría, rúbricas, evaluación.

1. Introducción

Durante muchos años, nuestro principal objetivo como docentes ha sido que los alumnos adquirieran un conocimiento sólido de las disciplinas que impartimos, por lo que hemos centrado nuestros esfuerzos

en mejorar aspectos como la planificación, los contenidos, o los materiales. Sin embargo, al recaer todo el trabajo de nuestro lado, esta mejora no se solía ver reflejada de forma sustancial en los resultados de los alumnos. Por este motivo, no debemos actuar como meros transmisores de conocimiento, sino que es de vital importancia hacer que el alumno participe activamente en su aprendizaje. Como se sugiere en [15], se trata de introducir al alumno en un camino del que no pueda salir sin aprender.

En esta línea de promover una mejora en la calidad del proceso de enseñanza - aprendizaje, la Universidad de Granada ha puesto en marcha un 'Curso de iniciación a la docencia universitaria' para profesores con menos de cinco años de experiencia docente que culmina en una etapa de mentoría donde el profesor novel es guiado y comparte experiencias con un profesor mentor con el objetivo de favorecer su inserción laboral.

El contenido de este artículo es parte de la experiencia de mentoría que tuvo lugar a lo largo del curso académico 2009/2010, y en la que participaron dos profesores de la Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada con docencia en las asignaturas 'Sistemas Operativos II' y 'Diseño de Sistemas Operativos' de Ingeniería Informática (profesor mentor), y 'Fundamentos de Informática' de la Diplomatura en Estadística (profesor novel).

2. El proceso de mentoría

Durante la etapa de mentoría se pretendían alcanzar dos objetivos. Por un lado, potenciar los aspectos positivos y subsanar las carencias existentes en la labor docente tanto del profesor mentor como del profesor novel, para lo cual se realizarían una serie de grabaciones. Por otra parte, analizar y profundizar en temas docentes de interés mediante la realiza-

ción de varios seminarios. Todo el trabajo realizado y los resultados obtenidos se irían reflejando en una memoria. A continuación se detalla cómo fue la experiencia a lo largo de este proceso.

3. Observaciones de la labor docente de los profesores

La grabación en vídeo de las clases impartidas por el profesorado permite, tras un posterior análisis y una crítica constructiva, mejorar la calidad de la docencia. Por ello, las observaciones entre compañeros deberían ser una práctica habitual en la universidad, aunque no sean necesariamente grabaciones en vídeo. No hay que tener miedo a ellas ni a las aportaciones de los compañeros, ya que no se trata de evaluar a nadie, sino de ser conscientes de la imagen que realmente se transmite (que en ocasiones difiere de la que se cree transmitir) y reflexionar sobre ella.

En este sentido se realizaron dos grabaciones por profesor, dejando suficiente tiempo entre medias para poder analizar la primera grabación y poner en práctica todos los consejos durante la segunda. Después de cada grabación, cada uno la veía por separado y la analizaba individualmente. Finalmente, se hacía una puesta en común para comentar las cosas que cada uno había apreciado.

Como modelo de análisis de las grabaciones se utilizó el Test de Florida [13], que permitía evaluar gran cantidad de aspectos docentes como la organización y desarrollo de la enseñanza, la presentación del tema, la comunicación verbal y no verbal, o el comportamiento de los alumnos. Experiencias similares se encuentran en [7]. Algunos de los aspectos a considerar en las grabaciones fueron los siguientes:

- Planificar el proceso de enseñanza - aprendizaje: preparar un material atractivo, equilibrado, y con una redacción adecuada, teniendo en cuenta nuestro objetivo y a quién va dirigido.
- Ser flexibles en el desarrollo de la asignatura: aunque la responsabilidad del aprendizaje debe recaer en los alumnos, no debemos estar atados a una planificación rígida, sino abiertos a introducir modificaciones o mejoras en función de los resultados de su aprendizaje.
- Optimizar el uso de herramientas y medios didácticos: estructurar los contenidos de la pizarra, no tapar a nadie, y apoyarse en vídeos o

presentaciones con diagramas y esquemas.

- Transmitir con credibilidad los conocimientos: tener autoconfianza, mostrar entusiasmo y convicción, usar anotaciones si es necesario pero sin leerlas textualmente, contextualizar los contenidos y presentarlos progresivamente en cuanto a complejidad, resaltando lo importante y los problemas más comunes.
- Mejorar la expresión oral: hablar con seguridad y dominio de la situación, sin usar muletillas o un tono de voz bajo y monótono, con el fin de mantener la atención de los alumnos e incidir en las cosas importantes.
- Mejorar la expresión corporal: evitar meterse las manos en los bolsillos, moverse por la clase para mantener a los alumnos atentos, acercarse a ellos y mirarles a los ojos para ganar credibilidad, demostrar que se les presta atención y favorecer su participación.
- Realizar preguntas sobre cosas concretas: limitar las preguntas a cuestiones específicas o dirigirlas a una persona determinada pues las generales o al aire no suelen tener respuesta, aunque pedir voluntarios a veces funciona, sobre todo si ese esfuerzo se ve reflejado en la nota.
- Proporcionar una realimentación académica: hacer un seguimiento de los alumnos (observación directa, actividades, trabajo en grupo, exámenes parciales...) y entregar los resultados en un plazo breve de tiempo para que puedan mejorarse de cara al futuro.

Con respecto a los detalles técnicos de las grabaciones, costó conseguir una cámara, grabar el vídeo y convertirlo, puesto que no se tenía experiencia. La imagen resultó ser buena, aunque algo oscura, y el sonido permitía entender lo que se decía a pesar de existir cierto ruido de fondo.

Al ser los profesores de diferentes áreas de conocimiento, el novel tenía que averiguar cómo explicar cada cosa de la mejor manera y dónde existía una mayor dificultad. Si bien es cierto, esto permitió centrarse en aspectos más generales como la forma de desenvolverse en las clases.

Aunque la presencia de la cámara sorprendía al principio, los alumnos y los profesores se comportaron con normalidad ya que no era la primera clase del curso y había cierta confianza.

3.1. Grabaciones al profesor novel

Se realizaron grabaciones en dos grupos de prácticas y aulas diferentes para tener en cuenta el tamaño del grupo y del aula, y el número de repetidores.

En la primera grabación, la densidad de puestos de trabajo dificultaba la movilidad del profesor por los pasillos. Además, las torres de los ordenadores estaban sobre la mesa, aumentando el ruido y llegando en ciertos puestos a tapar la pizarra, que tenía algún problema de iluminación. Sobre la labor docente del profesor novel, los ítems observados en el Test de Florida [13] daban unos indicadores correctos: explicación clara, pausada y estructurada, utilizando la pizarra para reforzar las respuestas a preguntas de los alumnos. Aún así, se podía mejorar en la formulación de preguntas concretas y la expresividad oral y corporal, enfatizando las partes importantes y estructurando mejor los contenidos de la pizarra.

La segunda grabación se realizó en un aula mejor iluminada y más espaciosa, y se produjo una mejora sustancial en los aspectos propuestos: mayor soltura, mejor entonación de la voz, más alegre, mayor acercamiento a los alumnos, preguntas más concretas, pizarra más estructurada, y resolución de los problemas más comunes. Incluso se propuso potenciar el aprendizaje activo para que el alumno sea más responsable de su aprendizaje.

3.2. Grabaciones al profesor mentor

Las dos grabaciones se realizaron para el mismo grupo de alumnos y aula, la cual tenía una mala acústica y hacía difícil el trabajo en grupo.

En la primera grabación se observó como, a pesar del aula, el profesor fomentaba con acierto el trabajo en grupo de los alumnos, les orientaba en su aprendizaje, y en ocasiones se metía las manos en los bolsillos, aunque debido al frío. Se acordó mejorar el orden de la pizarra, utilizar gráficas que favorecieran la discusión en clase, y promover la participación voluntaria del alumnado reservando un 10% de su calificación a tal fin. Finalmente, una expresión algo más alegre podría acercar más a los alumnos.

Estas mejoras se observaron en la segunda grabación: pizarra más ordenada, gráficas para la discusión de conceptos, preguntas de comprensión, y participación máxima de los alumnos evitando que trabajasen aislados.

4. Seminarios de mejora

En relación a los seminarios monográficos, el profesor novel planteó, a instancias de su mentor, tres temas que podían resultar de interés: metodologías docentes, tutorías, y evaluación.

Para la preparación de cada seminario, el mentor seleccionaba un material común de estudio que era analizado individualmente por ambos profesores y que servía como punto de partida. A lo largo de cada seminario se aclaraban las dudas surgidas y se analizaba la posibilidad de aplicar alguna propuesta en sus asignaturas. A continuación se comentan los aspectos concretos de cada uno de los seminarios.

4.1. Seminario sobre metodologías docentes

El primer seminario consistió en una revisión de las modalidades y metodologías de enseñanza más utilizadas [3, 10, 12]. Por modalidades de enseñanza nos referimos a las distintas maneras de llevar a cabo los procesos de enseñanza - aprendizaje, tanto presenciales (clases teóricas, prácticas, seminarios, talleres, tutorías...) como no presenciales (trabajo autónomo, individual o en grupo). Lo ideal es distribuir bien el volumen de trabajo del profesor y de los alumnos para poder trabajar con varias de ellas. Estas modalidades se concretan en procedimientos, estrategias o técnicas específicas constituyendo las diferentes metodologías de enseñanza (lección magistral, resolución de problemas...). El seminario finalizó con una descripción, a petición del profesor novel, de los métodos usados por el mentor en sus clases, indicando cuáles recomendaba y cuáles no.

4.2. Seminario sobre tutorías

A pesar de que las tutorías permiten mejorar el aprendizaje de los estudiantes, atendiendo a sus diferentes perfiles, son un recurso muy poco utilizado. Por esta razón se decidió realizar un seminario sobre tutorías, analizándolas [1, 4] desde dos puntos de vista diferentes. La primera como orientación de la formación académica integral del estudiante, facilitando su adaptación e integración en su formación como futuro profesional, y la segunda como estrategia didáctica orientada a facilitar el aprendizaje en el ámbito disciplinar de la asignatura. Nosotros creemos que las tutorías deberían centrarse princi-

palmente en la formación académica del estudiante, redirigiendo las tutorías fuera del ámbito de nuestra docencia a lugares especializados de la universidad. Así trataríamos aspectos sobre el trabajo individual o autónomo del estudiante y en pequeño grupo.

Por último indicar que cada vez más alumnos prefieren realizar sus consultas por correo electrónico o a través de plataformas de apoyo a la docencia en lugar de hacerlo presencialmente, por la comodidad de no tener que desplazarse al centro.

Como acción concreta se propuso establecer un plan de tutorías, que se describe en la Sección 5.1, para supervisar los trabajos y las presentaciones de los alumnos de la asignatura 'Diseño de Sistemas Operativos' mediante la corrección de borradores.

4.3. Seminario sobre formas de evaluación

El último seminario estuvo orientado a la adquisición de nuevas estrategias de evaluación. En contraposición a la evaluación sumativa, centrada en obtener una calificación final para el trabajo del alumno, la evaluación formativa [2, 6, 8, 9, 11, 14] identifica las debilidades del proceso de enseñanza - aprendizaje para poder actuar en consecuencia.

Llegamos a la conclusión de que la mejor forma de realizar una evaluación formativa era usando una carpeta de aprendizaje o portafolio, en la que el alumno anotaría las dificultades que encontrase en cada tema, sus ejercicios resueltos, trabajos, presentaciones... Así, el profesor supervisaría tanto el trabajo del alumno en clase como su trabajo autónomo.

Para la asignatura 'Sistemas Operativos II' se propuso utilizar los 'one minute papers', que consisten en dedicar unos minutos al final de la clase para que los alumnos contesten por escrito, y poniendo su nombre, a muy pocas preguntas, abiertas o de tipo test: qué ha sido lo más importante de esa clase, qué ha quedado confuso, o explicar algún concepto. Así los alumnos prestan más atención, llevan todo al día, reflexionan sobre lo que han estudiado, se sabe si han entendido todo, si hay errores individuales o generalizados a corregir, e incluso se pasa lista.

De forma complementaria, para la asignatura 'Diseño de Sistemas Operativos' se planteó la posibilidad de utilizar rúbricas para que los alumnos conociesen los criterios exactos de evaluación de sus trabajos (grado de elaboración del tema, aplicación práctica, uso de fuentes documentales contrastadas,

originalidad del enfoque, extensión y formato) y presentaciones (actitud, expresión oral, orden en la exposición, material audiovisual atractivo, trabajo en grupo, tiempo de exposición, y respuesta a las cuestiones planteadas). Esta medida, junto con el plan de tutorías, debía mejorar la calidad de los mismos. Además, las rúbricas podían ser utilizadas también para la evaluación entre los propios alumnos.

5. Aplicación práctica de los seminarios

El objetivo de los seminarios no sólo era realizar un estudio teórico sobre temas docentes, sino que estaba enfocado a obtener resultados prácticos. Por ejemplo, los seminarios de tutorías y evaluación se aplicaron en la mejora docente de la asignatura 'Diseño de Sistemas Operativos', una optativa de 5º curso de Ingeniería Informática. En dicha asignatura se exige que los alumnos realicen, en la parte teórica, dos trabajos y dos presentaciones de los mismos en clase. Los trabajos están destinados a que los alumnos amplíen, consoliden y profundicen en el diseño y construcción de sistemas operativos, mejoren la redacción y presentación de informes extendidos, sintetizen de manera adecuada las lecturas realizadas, y emitan juicios personales en un apartado de conclusiones. Por su parte, las presentaciones pretenden desarrollar cierta práctica en las exposiciones orales, dado que hoy en día se nos exige un dominio del discurso oral en el ámbito profesional (grupo de compañeros de la empresa, presentación de proyectos o productos...). Además, las presentaciones, más exhaustivamente que los trabajos, obligan a realizar un aprendizaje más profundo para sintetizar ideas y expresarlas de forma clara y concisa.

Hasta el curso pasado, el profesor sólo supervisaba el índice del trabajo y una bibliografía mínima de partida. A partir de aquí, el alumno elaboraba su trabajo y se evaluaba el producto final. En general, los resultados producían trabajos de una calidad intermedia, tanto por su contenido como en el formato.

Como parte del proyecto de mentoría, se diseñaron dos acciones de mejora y un mecanismo de evaluación del proceso. Las acciones de mejora comprendían un plan de tutorías de trabajos, y la elaboración de rúbricas para evaluar los trabajos y las presentaciones. Para la evaluación, se diseñó una encuesta para recabar la visión de los alumnos sobre las propuestas de mejora, las cuales se describen a

continuación. Los resultados de las encuestas se detallan en la Sección 6.

5.1. Plan de tutorías

El objetivo ha sido integrar las tutorías en el desarrollo de los trabajos y las presentaciones. De esta forma, a los alumnos se les propone el siguiente plan de trabajo en el que las tutorías son parte del mismo:

1. Localización y primera lectura de los documentos de partida para determinar la estructura del tema, que se materializa en un índice y una bibliografía. Para supervisar la calidad de los contenidos es obligatorio consultar al profesor sobre la adecuación de dicho índice y bibliografía.
2. Desarrollo del primer borrador de los apartados del índice: exposición de datos o información de los documentos leídos, argumentación y exposición de los propios puntos de vista, y presentación de conclusiones. Se establece una fecha para presentar el borrador al profesor, y se pueden ir consultando las dudas que surjan.
3. Confección del documento final usando las sugerencias del profesor que crea oportunas.
4. Elaboración de una presentación de 15 minutos sobre el tema, a partir del documento final.
5. Antes de realizar la presentación, y también de forma voluntaria, se puede consultar con el profesor los contenidos de la presentación, formato, etc. para tener en cuenta posibles mejoras.
6. Una vez elaborada la presentación definitiva se envía al profesor para colgarla en la página web.
7. Antes de realizar la presentación en clase conviene ensayar con los compañeros de grupo.

Como se puede observar, la tutoría está integrada en el plan de trabajo: revisión del índice y del borrador del trabajo y de la presentación. Esto es importante pues no sólo es valorable el resultado final sino también el proceso. Además, las diferentes revisiones del profesor constituyen una importante información de retorno (*feedback*) que transmite a los alumnos los puntos fuertes y débiles de su trabajo.

Podría pensarse que estas revisiones consumen mucho tiempo del profesor. En realidad no es así. El coste en tiempo por trabajo y presentación es bajo ya que se trata de una revisión global y no en detalle. Con cierta experiencia del profesor, se conocen los errores generales típicos de los trabajos en grupo

y se puede incidir en ellos: el tema se trata de forma superficial, no hay ejemplos o aplicaciones, etc.

Puesto que los trabajos se exigen en formato electrónico, esta tutoría se puede hacer en gran medida a través del correo, lo que proporciona una gran inmediatez en la respuesta y evita desplazamientos innecesarios de los alumnos al centro, cosa que agradecen bastante pues hay un número importante de ellos que trabajan. Además, las correcciones utilizando el control de cambios o anotaciones en el propio documento facilitan el proceso de revisión.

5.2. Uso de rúbricas

La calificación de la asignatura se realiza a partir de la valoración de los trabajos y presentaciones. Antes de aplicar las acciones de mejora sólo se disponía de unos criterios generales para la evaluación: grado de elaboración de la información aportada, trabajo teórico o con aplicaciones prácticas, cantidad y actualidad de las fuentes documentales utilizadas, originalidad en el enfoque, extensión y formato.

La realización de trabajos y presentaciones junto con el plan de tutorías pretende establecer un sistema que favorezca el aprendizaje de los alumnos, usando un esquema de evaluación formativa [14] orientada no sólo a calificar sino a corregir los errores en el momento en el que se producen. Así se consigue un aprendizaje más profundo y duradero, y se evita que estos errores tengan carácter sancionador (calificación negativa) [2].

Para ello, se han transformado los criterios anteriores en rúbricas o matrices de evaluación. Se han convertido unos criterios generales en un sistema analítico que permite las dos vertientes de la evaluación: las rúbricas se utilizan como forma de evaluación integral y formativa. Por un lado permiten calificar los trabajos y por otro sirven como instrumento de orientación de la práctica educativa. El profesor unifica los criterios de evaluación de diferentes estudiantes, y el alumno aborda su aprendizaje sabiendo los objetivos y metas del mismo. No sólo se mejora el resultado final, sino todo el proceso de realización de los trabajos y presentaciones. Además, el uso de criterios de evaluación mejora las calificaciones y se acercan más a la distribución normal [6].

De forma resumida, la rúbrica consiste en tratar de fijar una escala descriptiva de acuerdo a unos criterios establecidos. Se materializa como una tabla de

doble entrada. El primer eje, de categorías, recoge los elementos a evaluar, es decir, los más importantes según los objetivos. El segundo eje establece el nivel de calidad con los valores asociados.

Como puede verse en la Tabla 1 (rúbrica para los trabajos) y la Tabla 2 (rúbrica para las presentaciones orales), en el eje de categorías se desglosan los criterios establecidos en el plan de tutorías propuesto para la asignatura. A cada categoría se le añade un peso en la calificación final según su importancia. En el eje de valoraciones se establecen cuatro niveles (insuficiente, aceptable, buen nivel, y excelente) con su rango de puntuación asignado. En ambos casos, trabajos y presentaciones, se tiene en cuenta que para la asignatura los primeros están valorados sobre 1.75 puntos y los segundos sobre 0.75 puntos.

6. Resultados obtenidos

Esta mentoría ha supuesto una gran oportunidad para el intercambio de ideas. La grabación de las clases en vídeo y su análisis ha permitido mejorar aspectos de la comunicación con los alumnos como la necesidad de realizar preguntas concretas y personalizadas, enfatizar con el gesto y la voz los puntos importantes, y una adecuada estructuración de los contenidos de la pizarra. Sin embargo, si se combinase con demostraciones de incidentes que el profesor deba resolver (alumnos que no dejan de hablar o piden un aplazamiento de la evaluación), se conseguirían mejores resultados.

Sobre los seminarios, se realizaron dos encuestas para que los alumnos evaluaran (de 0 a 5) si las acciones realizadas (tutorías y rúbricas) les habían servido para hacer sus trabajos y presentaciones.

El plan de tutorías fue utilizado por 13 de los 19 alumnos (Figura 1), y para la mayoría fue bastante útil (Figura 2). Los aspectos más destacados fueron la mejora en la calidad de los trabajos, la orientación sobre la búsqueda de información, y la colaboración del profesor. Sobre los aspectos a mejorar, alguien propuso una mayor rapidez en la respuesta, aun reconociendo que se le contestó en fines de semana o Semana Santa. En cuanto a los que no usaron las tutorías, los motivos fueron la falta de tiempo (había que enviar un borrador al profesor), y que no creyeron que fuese necesario.

En relación a las rúbricas, sólo la mitad de los alumnos (10 de los 19) las utilizaron, aunque para

Sesión 8A: Calidad y evaluación de la docencia

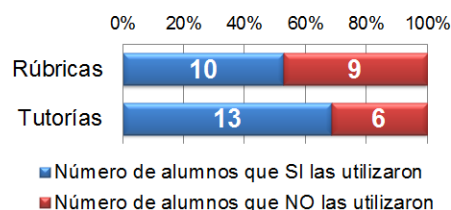


Figura 1: Uso de tutorías y rúbricas.

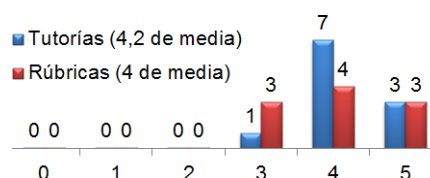


Figura 2: Valoraciones del plan de tutorías y rúbricas.

estos fueron de bastante utilidad. Consideraron que era una buena metodología para generar trabajos de calidad, y permitían saber los aspectos en los que centrarse. En contra, se podían diferenciar más algunas categorías y hacerlas más objetivas. Las razones de no utilizar las rúbricas fueron que se entregaron tarde (la idea surgió iniciado el curso), no se conocía su existencia, se habían utilizado las correcciones del profesor, o no se tenía claro cómo utilizarlas.

La comparación de las notas obtenidas entre los primeros trabajos/presentaciones y los segundos, indica un ligero aumento de las calificaciones, pero no todo el que deseáramos. Esto es justificable por una parte debido a que las acciones de mejora comenzaron una vez iniciado el curso. Por otra, en base a los resultados de las encuestas, vemos que es necesario emplear cierto tiempo en enseñar a utilizar las rúbricas y mostrar ejemplos de buenos y malos trabajos.

Remitiéndonos a los resultados de los 'Cuestionarios de opinión del alumnado sobre la actuación docente del profesorado' que realiza anualmente la universidad, si comparamos los del curso actual (con las acciones de mejora) con los del curso anterior, hay que resaltar que prácticamente todos los ítems de la encuesta mejoran en un 10% de media, y especialmente los más relacionados con esas acciones. Así, el alumno se siente mejor informado sobre los criterios de evaluación (11%), cree que esta es adecuada, y estima que se le atiende más y mejor en tu-

Valoración		Insuficiente [0-0.5 puntos]	Aceptable [0.6-1 puntos]	Buen nivel [1.1-1.4 puntos]	Excelente [1.5-1.75 puntos]
Criterio					
(50%) Grado de elaboración de la información aportada	Desarrollo	Ninguna consistencia en el fluir del tema, elaboración y profundidad.	Muy poca consistencia en el fluir del tema, elaboración y profundidad.	Consistencia parcial en el fluir del tema, elaboración, y profundidad.	Consistente en el fluir del tema elaboración y profundidad.
	Contenido	Organización de ideas deficiente y sin coherencia (confuso, incompleto y sin una dirección clara). Ausencia de conocimiento previo y corrección conceptual. No hay gráficos o estos no ayudan a la comprensión. Información a un nivel muy simplista. El contenido no demuestra que se haya entendido lo que se ha trabajado.	Algunos errores en la organización de ideas y párrafos (se han cubierto las diferentes secciones pero no hay conexión ni transición entre ellas). Falta de conocimientos previos y corrección conceptual. Falta algún gráfico que ayude a la comprensión de algún concepto. El contenido demuestra que se ha entendido lo que se ha trabajado.	Las ideas están elaboradas y en general los párrafos son coherentes (la organización es adecuada y está relacionada entre si). Presencia parcial de conocimiento previo y corrección conceptual. Ayudas gráficas para los conceptos, pero algunas no son las mejores. La información es clara y muestra cierta reflexión sobre el tema.	Muy buena elaboración de ideas y coherencia del documento (muestra una planificación cuidadosa y una secuenciación lógica y clara). Presencia completa de conocimientos previos y corrección conceptual. Magnífica selección de ayudas gráficas. La información es excelente: se ha entendido el tema, y se ha reflexionado.
	Redacción	La redacción no está cuidada. Hay faltas de ortografía.	Redacción cuidada pero con errores. Algunas frases confusas y de difícil comprensión.	No hay errores de redacción, ni faltas de ortografía.	La redacción es excelente por su claridad y cuidado.
Trabajo teórico con aplicaciones	No existe ningún ejemplo de aplicabilidad o uso de los conceptos	Establece aplicabilidad o usos prácticos de alguno de los conceptos teóricos	Parte, pero no todos los elementos teóricos, tienen su ejemplo	Establece un equilibrio perfecto entre la teoría y su aplicación	

Tabla 1: Fragmento de la rúbrica de trabajos escritos [5].

torías (13%). También cabe resaltar un incremento del 13% en la motivación por la disciplina, resultado que mejora notablemente si se compara con las medias del área de conocimiento y de la titulación.

7. Conclusiones

En este artículo se ha presentado una experiencia de mentoría realizada en el curso 2009/2010, la cual ha sido muy enriquecedora, no sólo para el profesor novel, sino también para el mentor y los alumnos de sus asignaturas, ya que todos se han beneficiado de las acciones llevadas a cabo, como reflejan las encuestas realizadas al final del proceso. Los profesores han mejorado aspectos de la docencia encaminados a corregir la puesta en escena de las clases, y a disponer de un amplio abanico de metodologías docentes y de evaluación. En concreto, se han integrado las tutorías como parte de la actividad docente y se ha suministrado un mecanismo de evaluación

analítica, las rúbricas. Por su parte, los alumnos no sólo valoran positivamente las acciones realizadas, sino que aumentan su motivación por la asignatura y sienten que se beneficia su aprendizaje.

De cara a los próximos cursos hay dos puntos por mejorar. Uno estará encaminado a enseñar al alumno a utilizar las rúbricas aplicándolas a trabajos de ejemplo de diferente calidad. Otro estará orientado a hacer participar a los alumnos en el proceso de evaluación de los compañeros, dado que las rúbricas facilitan y hacen más objetiva esta labor.

Agradecimientos

Ambos profesores desean agradecer a sus alumnos su predisposición y participación en las sesiones de grabación y en las diferentes iniciativas que se han realizado. Igualmente agradecer al Vicerrectorado de Garantía de la Calidad por llevar a cabo tan grata y fructífera experiencia.

Valoración Criterio	Insuficiente [0-0.1 puntos]	Aceptable [0.2-0.3 puntos]	Buen nivel [0.4-0.5 puntos]	Excelente [0.6-0.75 puntos]
Actitud (15%)	Toda la exposición se ha realizado sin dirigirse a la audiencia. Tono de voz inadecuado. Lectura de anotaciones excesiva.	En ocasiones no se dirige a la audiencia y baja el tono de voz de forma que no se entiende. Puntualmente lee alguna nota para recordar.	La presentación es fluida. Los compañeros la siguen con interés.	Siempre se dirige a la audiencia. Los gestos, voz y compostura son correctos. Capta el interés.
Expresión oral y estructura de las frases (15%)	Incorrecta (uso inadecuado de los términos). Las frases son siempre confusas y no las termina.	Correcta, pero con algunas imprecisiones, y en algunos casos emplea frases entrecortadas.	Lenguaje claro pero en ocasiones no es conciso o adecuado.	Se usa un lenguaje claro, conciso, y adecuado. La construcción de las frases es correcta.
Relación de ideas (20%)	Exposición desordenada y con	Exposición ordenada, pero se muestran	Exposición ordenada, pero a veces no	Exposición ordenada, precisa y correcta.

Tabla 2: Fragmento de la rúbrica de presentaciones [5].

Referencias

- [1] Álvarez, V.B. *La tutoría: elemento clave en el Modelo Europeo de Educación Superior*. Salamanca, 2004.
- [2] Casanova, M.A. *La evaluación educativa*. SEP-Muralla, México, 1998.
- [3] De Miguel, M. *Modalidades de enseñanza centradas en el desarrollo de competencias. Orientaciones para promover el cambio metodológico en el Espacio Europeo de Educación Superior*. Ediciones Universidad de Oviedo, 2005.
- [4] Delgado, J.A. *Líneas básicas de la intervención en tutoría universitaria*. Ed. Método, 2005.
- [5] Gómez, J.A. *Los trabajos y presentaciones de la Asignatura Diseño de Sistemas Operativos: objetivos, normas y evaluación*. 2010. Disponible en línea [Mayo, 2011] <<http://lsi.ugr.es/jagomez/disisoparchivos/trabajosDSO/Evaluacion.html>>.
- [6] Kuisma, R. *Criteria referenced marking of written assignments*. *Assesment & evaluation in higher education*, 24(1):27-39, 1999.
- [7] León, M.J.; López, M.C. (Coords.) *Experiencias de Mentorización a Profesorado Principiante en la Educación Superior*. Actas Primeras Jornadas de Intercambio de Experiencias de Mentorización en la Educación Superior. Universidad de Granada, 2009. Disponible en línea [Mayo, 2011] <http://calidad.ugr.es/pages/secretariados/form_apoyo_calidad/programa_iniciacion_docencia/jornadas-intercambio/experienciasdementorizacion>.
- [8] López, M.C. *Evaluación de los procesos de enseñanza-aprendizaje en la universidad y su adaptación al Espacio Europeo de Educación Superior*. Editorial Universidad de Granada, 2007.
- [9] Morales, P. *La evaluación formativa*. Publicado en Morales Vallejo, P., *Ser profesor: una mirada al alumno*. Guatemala: Universidad Rafael Landívar, 41-98, 2009.
- [10] Navarro, J.; Valero, M. *Taller de formación: aprendizaje basado en proyectos (project based learning)*. Univ. Politécnica de Madrid, 2007.
- [11] Olmos, S. *Evaluación formativa y sumativa de estudiantes universitarios: aplicación de las tecnologías a la evaluación educativa*. Tesis doctoral, Universidad de Salamanca, 2008.
- [12] Prieto, L. *La enseñanza universitaria centrada en el aprendizaje*. Octaedro, 2008.
- [13] Smith, B.O. y otros. *Evaluation and Professional Improvement Aspects of the Florida Performance Measurement System*. *Educational Leadership*, 7(44) 16-19, 1987.
- [14] Smyth, K. *The benefits of students learning about critical evaluation rather than being summatively judged*. *Assessment & evaluation in higher education*, 29(3): 370-378, 2004.
- [15] Valero, M. *¿Cómo se adapta una asignatura al sistema ECTS?* Taller previo a las XI Jornadas de Enseñanza Universitaria de la Informática (JENU), Universidad Europea de Madrid, 12 de Julio, Villaviciosa de Odón, Madrid, 2005.

Sesión 8B:
Fundamentos teóricos de la Informática
Inteligencia Artificial

Sistema de Generación Personalizada de Ejercicios de Lógica Computacional

Jose M. Alcaraz Calero^{1,2}, Luis Daniel Hernandez Molinero¹,
Santiago Paredes Moreno¹

¹Departamento de Ingeniería de la Información y las Comunicaciones
Universidad de Murcia
30100 Murcia
{jmalcaraz, ldaniel, chapu}@um.es

²Cloud and Security Lab
Hewlett-Packard Laboratories
BS34 8QZ Bristol
{jose.alcaraz-calero}@hp.com

Resumen

El presente artículo describe el diseño, desarrollo e implementación de un framework que permite a los docentes la generación tanto de boletines entregables de ejercicios como de tutoriales interactivos de ejercicios con su explicación y solución. Estos recursos son personalizados para cada alumno porque se utiliza un repertorio de ejercicios adecuado al nivel del conocimiento que cada cual tenga sobre la materia. Este framework también dificulta algunos malos hábitos entre los alumnos como la copia de ejercicios o la distribución de los mismos, posiblemente con soluciones incorrectas. Es especialmente útil en asignaturas donde se requiere una fuerte componente de notación matemática tales como son la asignatura de lógica computacional o de matemáticas ya que está basado en el procesador de textos *LaTeX*. Además, esta herramienta da soporte a nuevas metodologías docentes más personalizadas tanto en el aula como fuera de ella.

Summary

This article describes the design, development and implementation of a framework which enables teachers the generation of both home works and interactive tutorials of exercises with explanation and solution. These resources are personalized for each student since it is used a exercises repository adapted to the knowledge level of every student have about the subject. This framework makes difficult some bad habits such as plagiarism and distribution of such exercises, maybe with

incorrect solutions. It is especially useful in subjects in which is required a strong components of mathematical notation such as computational logics and mathematics since it is based on *Latex* word processor. Moreover, this tools offer support for new teaching methodologies more personalized in both classroom and autonomous learning.

Palabras clave

Ejercicios, Notación, Personalización, TIC, Recurso Metodológico, Interactivo.

1. Motivación

En cualquier asignatura que requiere de un lenguaje formal como son las asignaturas de lógica computacional y matemáticas se presenta una problemática importante en el diseño de actividades para el alumno y su posterior corrección. En efecto, la propuesta de actividades de dificultad creciente requiere de un gran dedicación en su diseño y puede que aún más en su escritura formal, además si éstas son de desarrollo, p.e. la demostración de un teorema, y debido a su naturaleza, es muy posible que requiera de mucho esfuerzo para su corrección y si son de ejercitación, p.e. calcular una tabla de verdad, es muy fácil el plagio entre alumnos. Tradicionalmente, este tipo de asignaturas están asociadas al uso de la pizarra como principal mecanismo de interacción alumno-profesor en clase por lo que se plantea una problemática

especialmente acentuada cuando se desea hacer uso de las tecnologías de la información y las comunicaciones (TIC) para el diseño de actividades en formato electrónico, pues la usual interfaz hombre-máquina no está precisamente adaptada a notaciones formales, complicando mucho la realización de actividades basadas en ordenador, tanto en clase como fuera de ella.

La mayoría de las herramientas disponibles hoy en día, especialmente las orientadas a la evaluación, crean ejercicios demasiado estrictos en la introducción de datos del usuario [1]. Si bien es cierto que existen herramientas, que recurriendo a tecnologías estándares, ayudan a mostrar el lenguaje formal matemático de forma on-line, p.e. [2], no es menos cierto que éstas son difícilmente parametrizables a las preferencias de cada profesor o al nivel de dificultad de las necesidades formativas propias de los alumnos.

Por ello, este artículo describe un framework capaz de proporcionar al profesor la posibilidad de generar tanto boletines entregables de ejercicios personalizados para cada alumno como tutoriales interactivos de ejercicios con sus respectivas explicaciones y soluciones proporcionando un nuevo mecanismo de interacción alumno-profesor y dando pie a un posible cambio de metodología de enseñanza tanto en el aula como en el diseño de las actividades que conforman el trabajo autónomo del alumno, necesidad que pone de manifiesto el EEES [3].

El resto de este artículo está distribuido de la siguiente manera: La sección 2 describe los trabajos relacionados con la utilización de las TIC para la personalización de ejercicios y tutoriales. La descripción de la arquitectura del sistema propuesto está detallada en la sección 3. La sección 4 describe los lenguajes utilizados para describir la batería de ejercicios y boletines utilizados en el sistema. La sección 5 expone algunos detalles de implementación. Finalmente, la sección 6 proporciona algunas conclusiones y trabajos futuros.

2. Trabajos Relacionados

En los últimos años han ido apareciendo numerosos trabajos sobre el uso de las TIC como mecanismo mediador de comunicaciones alumnos-profesor dentro de las metodologías docentes.

Así, *Castel de Haro et al.* [4] proporciona un enfoque sobre la utilización de las TIC para hacer más interesante el estudio de la lógica computacional. Mientras que esta es una primera aproximación a las metodologías docentes adecuadas al EEES, todavía existe una carencia importante en el soporte de herramientas que permitan la personalización de ejercicios y tutoriales al alumno.

En este sentido, *Lu* [5] proporciona un sistema recomendador de material de e-learning personalizado para el alumno de acuerdo a sus necesidades basado en el feedback proporcionado por el alumno. En la misma línea, *Holohan et al* [1] proporciona un mecanismo para llevar a cabo la generación de ejercicios para e-learning basados en ontologías.

No es menos cierto que en el intento de avanzar tecnológicamente nos encontramos con ciertos efectos colaterales como indica *Lathrop and Foss* [6] al poner de manifiesto los nuevos retos contra el plagio y el engaño en la nueva era de Internet, enfatizando la necesidad de diseñar sistemas capaces de identificar estos hechos.

Si bien la idea de *Lu* y *Holohan et al* está muy cercana al procedimiento que presentamos en este artículo para la selección del material de e-learning ofrecido al alumno en su proceso de auto-aprendizaje, también consideramos imprescindible la personalización de los boletines de prácticas para dificultar el plagio y el engaño de los alumnos. La consecuencia de todo ello es la creación de una infraestructura que permite la creación de tutoriales de aprendizaje interactivos.

3. Arquitectura del Sistema

La Figura 1 representa una visión global de la del sistema de entrega de ejercicios y/o tutoriales.

El sistema de entrega presenta una arquitectura cliente-servidor en la que el estudiante hace uso de cualquier navegador web para acceder al sistema. Una sesión normal en esta arquitectura es la siguiente:

- 1.- Inicialmente, el estudiante accede al módulo de autenticación mediante su dirección de correo electrónico, o DNI, y contraseña. Este mecanismo de seguridad restringe el uso del sistema y permite establecer un entorno controlado del mismo, de tal manera que solo los

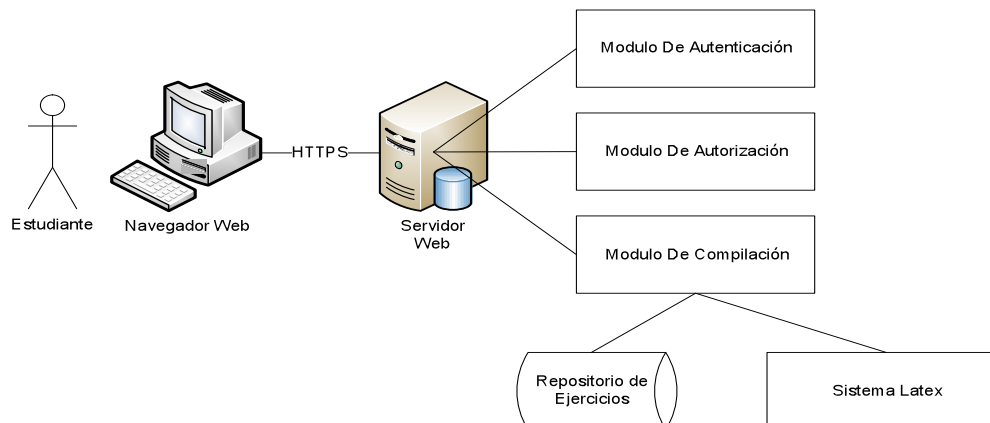


Figura 1. Visión general de la arquitectura del sistema

usuarios que el profesor decida puedan utilizar la aplicación.

2.- Autenticado el usuario, se detectan los boletines y/o tutoriales a los que puede acceder en función del grupo al que pertenece utilizando el módulo de autorización. Esto permite a la aplicación la versatilidad suficiente para gestionar diferentes grupos de usuarios/ asignaturas. Además, mediante este mecanismo de autorización se controla la privacidad entre los contenidos de diferentes asignaturas de tal manera que los usuarios no puedan acceder a boletines de otras materias con el fin de que se centren solo y solo en una asignatura.

3.- Determinados los boletines a los que el alumno tiene acceso, de entre todos los disponibles en el repositorio de boletines, se le da la posibilidad de seleccionar un boletín sobre uno o varios tipos de ejercicio. Entendemos por boletín al compendio de enunciados de ejercicios que deben cumplimentarse por el alumno y pueden, o no, presentarse con su solución. Obviamente, este boletín es totalmente personalizado y particularizado para ese alumno. Para determinar cómo se lleva a cabo este proceso de personalización véase la sección 4 en la que se describe el diseño del repertorio de ejercicios.

4.- Generado el boletín de ejercicios, se le entrega al usuario que puede, si así lo desea, descargarlo en su disco local.

La clave de todo el proceso está en disponer de una buena base de ejercicios y el formato en que éstos deben mostrarse, y es en lo que incidimos en los siguientes apartados.

4. Repositorio de Ejercicios

La arquitectura utiliza diferentes fuentes de datos para representar el repositorio de ejercicios. Se utiliza un único fichero denominado *fichero de boletines* para definir los diferentes boletines y/o tutoriales que se administran en la aplicación. De ahora en adelante se referirá indistintamente a boletín y a tutorial, pues aunque atiende a conceptos distintos la aplicación soporta ambos por igual. Cada boletín está estructurado necesariamente por una cabecera, un pie de página y un conjunto de ejercicios. Este fichero se explica con detenimiento en la sub-sección 4.1.

Aunque la cabecera, el pie de página y los ejercicios sean elementos diferentes, todos se gestionan de la misma manera dentro de la aplicación. La aplicación posee tres directorios denominados *headers*, *footers*, *exercises* y ellos contienen todas las posibles cabeceras, pies de página y ejercicios que pueden ser manejados por la aplicación. Cada uno de estos directorios posee un par de ficheros denominado *fichero de plantilla* y *fichero de instancias* para cada pie de página, cabecera y tipo de ejercicio (asistido o no) que puede aparecer en un boletín/tutorial. Estos ficheros definen una plantilla del tipo de ejercicio a generar en la aplicación y una batería de ejemplos para esa plantilla, respectivamente. Ambos serán explicados en las subsecciones 4.2 y 4.3. La Figura 2 muestra la estructura de directorios y ficheros utilizada en la arquitectura. En dicha figura se pueden observar los tres directorios con sus respectivos pares de ficheros

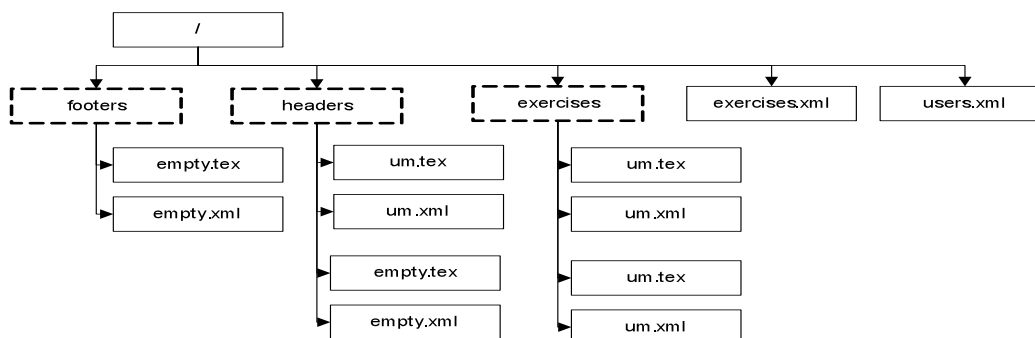


Figura 2. Estructura de ficheros propuesta

plantilla y de instancias. Además, el fichero de boletines (*exercices.xml*) y el fichero en el que se indican los usuarios, contraseñas y roles de cada usuario del sistema (*users.xml*).

4.1. Fichero de Boletines

El fichero de boletines contiene todos los boletines que el profesor desea gestionar en la aplicación (*exercices.xml*). Éste está estructurado necesariamente por cabecera, pie de página y un conjunto de ejercicios. El Listado 1 muestra un ejemplo de definición de boletín que se encuentra dentro de este fichero:

```

<bulletin bulletinId="entrega1"
  description="1º Entrega"
  randomOrder="false"
  groupsId="g1-logica">
  <header headerId="um"/>
  <exercise exerciseId="L0-n0-a"
    randomEntry="true"/>
  <exercise exerciseId="L0-n0-a"
    randomEntry="false"
    exerciseEntryId="1"
    shuffleCases="true"/>
  <footer footerId="empty"/>
</bulletin>
  
```

Listado 1. Ejercicio de definición de boletín

Como puede observarse, un boletín (identificado con la etiqueta XML *boletín*) puede

pertenecer a uno o más grupos de alumnos (atributo *groupsId*). Un grupo de alumnos es una agrupación lógica que decide el profesor y que puede atender tanto al concepto usual de grupo teórico, como al de subgrupo de prácticas, así como a cualquier otro tipo de grupo con el número de alumnos que considere adecuado el docente para aplicar cualquier metodología e-blending o activa [7][8]. Solo los alumnos del grupo al que está asociado el boletín pueden ver el mismo. Además un boletín puede configurarse en modo *orden aleatorio*, o no, indicando si los ejercicios que componen el boletín se presentan en el orden preestablecido por el profesor o por el contrario los ejercicios se presentan en un orden aleatorio, dificultando así el plagio, copia o engaño por parte del alumno. Esta opción es especialmente útil si desea generarse boletines de prácticas muy individualizados o bien incluso para generar exámenes que afecten a la calificación final de la asignatura.

Note que cada boletín está compuesto de una cabecera, una serie de ejercicios y un pie de página, todos ellos indicados mediante un *identificador*, las expresiones terminadas en *Id*. Este identificador ha de coincidir con el nombre de los ficheros que contienen los ejercicios que se desea utilizar. Este manera de enlazar boletines con ejercicios permite a los profesores mucha versatilidad y sencillez en los cambios, pues añadir un nuevo ejercicio o una nueva cabecera solo consiste en crear un nuevo fichero, colocarlos en la ubicación correspondiente y usar el nombre de ese fichero como identificador.

Cuando nos referimos a un ejercicio hay que distinguir dos niveles de abstracción. Por un lado

está el tipo de ejercicio para hacer referencia a aquel que se diseña para conseguir un objetivo formativo de la asignatura. Por otro lado están las instancias de ese tipo de ejercicios para hacer referencias a casos particulares de ejercicios. Por ejemplo, un tipo de ejercicio es calcular tablas de verdad, y es una instancia aquel que pide determinar la tabla de verdad para una expresión lógica particular.

El identificador *exerciseld* denota el tipo de ejercicio que se va a incluir en el boletín pero no así la instancia concreta de dicho ejercicio. Es decir, los ficheros de boletines especifican el tipo de ejercicios que deben generarse y no entran en qué ejercicios concretos (instancias) se desean incluir.

El motivo de este diseño es para conseguir la suficiente versatilidad para poder extender fácilmente la batería de ejercicio a la vez que se consigue la separación de niveles de dificultad de los mismos, la ordenación de los diferentes tipos de ejercicios y la fácil modificación de todos ellos (a través de la plantilla) por tanto el sistema se ha diseñado de manera que cada tipo de ejercicio tiene asignada la misma dificultad en todas sus instanciaciones del repertorio y por tanto dos ejercicios diferentes podrían tener exactamente el mismo enunciado (fichero plantilla) pero las instancias proporcionan un nivel de dificultad mayor o menor.

Además, la característica de aleatoriedad sobre los tipos de ejercicios hace que cada alumno puede tener potencialmente ejercicios análogos a los de sus compañeros pero no iguales, obligándolo, en cierta medida, a la necesidad de realizarlo si desea realmente aprender y dificultando notablemente el plagio de los mismos. Por otro lado, el profesor, si así lo desea, puede también generar siempre el mismo ejercicio para todos los alumnos (ver el segundo ejercicio del Listado 1), para ello ha de desactivar la característica de aleatoriedad (atributo *randomEntry*) e indicar el identificación de la instanciación que desea utilizar (versión sección 4.3 para más información).

Otra característica diferenciadora de este sistema es la posibilidad de barajar (*aleatorizar*) las diferentes opciones que se den en la instancia de un ejercicio concreto (atributo *shuffleCases*). Obviamente, esta opción solo tienen sentido cuando la respuesta es de elección múltiple, V/F,

etc. Esto se utiliza para evitar situaciones en las que todos las instancias de ejercicios del mismo tipo tiene un patrón predecible para mostrar la solución y los alumnos podrían detectarlo y intuir siempre la solución correcta.

4.2. Fichero de Plantilla

Este fichero se escribe utilizando el lenguaje *LaTeX*, y es el resultado de unir la cabecera, los ejercicios instanciados y el pie de página de un boletín. Una vez generado, es compilado con *pdfLaTeX* para obtener el fichero PDF correspondiente y que es el enviado al alumno.

El contenido de estos ficheros plantilla es totalmente personalizable y la única exigencia real es que el resultado de las concatenaciones de estos ficheros ha de ser un fichero *LaTeX* que compile correctamente. Aun así, para el contexto de la generación de ejercicios interactivos y tutoriales, se ha optado por la utilización de *AcroTeX* [9], un paquete de ficheros para *LaTeX orientados a la docencia*. Este paquete permite hacer ejercicios utilizando toda la potencia hipermédia, formularios y javascript embebido en ficheros PDF. Así, por ejemplo, se pueden diseñar ejercicios con soluciones interactivas, enlaces, animaciones gráficas, ejercicios de múltiple sección, ejercicios con gráficos y un largísimo etcétera. *AcroTeX* es la piedra angular de la arquitectura presentada en este trabajo de docente. Concretamente, *Acrotex* define un entorno *sortquiz* en el que se indican los ejercicios y un entorno *solution* en el que se indican las soluciones a dichos ejercicios y las explicaciones de dichas soluciones (para hacer tutoriales).

El Listado 2 muestra un ejemplo de fichero de plantilla que, utilizando los entornos *sortquiz* y *solution* de *AcroTeX*, generará un ejercicio que se auto-corrige de forma automática cuando el alumno introduce su solución. Responde a un sencillo ejercicio del tipo V/F en el contexto de la lógica computacional: se muestra al alumno una fórmula y se pregunta si es una fórmula bien formada o no.

En el Listado 2 tiene una mención especial los *anclajes*. Un anclaje es un patrón de la forma $\{variable\}$ donde *variable* es el nombre del anclaje (en el ejemplo hay cinco). Cada uno de estos anclajes serán posteriormente rellenados con los valores de las diferentes instancias de ejercicios que se introduzca en el fichero de

instancias. El número de anclajes depende totalmente de la plantilla que se esté diseñando (del tipo de ejercicio) y el sistema ha sido diseñado para soportar cualquier número de anclajes. Este número determinará cuantas variables deberá tener el fichero de instancias que se defina para este ejercicio.

Note también que el Listado 2 está dividido en 3 partes, que es el único patrón estructural que, por conveniencia y no por limitaciones de programación, conviene respetar en el diseño de los boletines. La primera de ellas se ha realizado con la única finalidad de separar al máximo las variables que han de instanciarse del resto de la plantilla. En esta sección pueden verse 3 comandos (*\expresion* *\valoruno* *\valordos*) que se redefinen con los valores que serán instanciados con el fichero de instancias y que permiten referenciar a estos valores antes de ser instanciados. Luego se define el enunciado (en el que se usa la expresión anteriormente definida) y luego se indica la solución, indicando que *\valoruno* es la solución correcta si el valor de la variable/anclaje *\$R1_solucion* es 1 o incorrecta si es 0.

```
%---- Variables del ejercicio
\renewcommand{\expresion}{
${expresion}}

\renewcommand{\valoruno}{
${R1_valor}}

\renewcommand{\valordos}{
${R2_valor}}

%---- Ejercicio

%---- Enunciado
\begin{shortquiz}
Observa la siguiente expresión y
pincha sobre la opción correcta:
$$\expresion$$

%---- Soluciones
\begin{answers}{2}
\Choices[random]
  \Ans${R1_solucion}
  \valoruno \eAns
  \Ans${R2_solucion}
  \valordos \eAns
\end{answers}
\end{shortquiz}
```

\newpage

Listado 2. Ejemplo de plantilla de ejercicio

4.3. Fichero de Instancias

Este fichero contiene todas las instancias de un tipo de ejercicio y tiene formato XML. Cada una de estas instancias (definida con la etiqueta *exerciseInstance*) está compuesta por tantas variables como puntos de anclaje tenga la plantilla asociada. La asociación entre fichero de plantilla y de instancias se hace a través del nombre del fichero (ambos se llaman igual, pero con diferente extensión), lo que dota de versatilidad al sistema.

```
<exerciseInstance id="1">
  <v name="expresion">p \neg q</v>
  <v name="R1_valor">No es fbf</v>
  <v name="R1_solucion">1</v>
  <v name="R2_valor">Es fbf</v>
  <v name="R2_solucion">0</v>
</exerciseInstance>
```

Listado 3. Ejemplo de instancia del ejercicio.

El Listado 3 muestra un ejemplo de instancia para la plantilla anteriormente descrita en el Listado 2. Nótese como las 5 variables disponibles en la plantilla son reemplazadas por sus correspondientes valores. Además, nótese la facilidad en la que un profesor añade un nuevo ejercicio: simplemente indica que una expresión cualquiera, tiene dos valores: “No es Formula Bien Formada (FBF)” y “Es FBF” y que uno de ellos es correcto. Fruto de la experiencia se ha comprobado que simplemente escribiendo el enunciado del ejercicio como comentario XML al principio del documento, este fichero se puede rellenar sin necesidad de recurrir a la plantilla para nada, ya que un simple *copy & paste* del ejercicio anterior te da la información suficiente para hacer otra nueva instancia para el repositorio.

Cada instancia tiene un identificador que puede ser utilizado para indicar en los boletines que es esa instanciación concreta la que se desea que aparezca en los mismo (atributo *id*). Además, merece la pena mencionar que el sistema seleccionar la instancia del ejercicio (o al azar o una concreta) y reemplazara los valores en la plantilla, *instanciándola* y generando la porción

de fichero látex que será concatenada a las otras porciones que componen el boletín.

Aunque el nombre que se puede dar a los puntos de anclaje es totalmente libre para el usuario, nótese que en el Listado 3, algunos tienen el patrón: “ R ” + *Numero* + “_” + *Nombre de variable*”. Esta es la manera de indicar que si se desean reordenar las opciones de forma aleatoria, solo se va a aleatoria aquellas que empiecen con R + *Numero* + “_” y además, todas las que tienen el mismo número han de mantener el mismo remplazo de aleatoriedad. Por ejemplo, para el Listado 3, si el ejercicio del boletín tiene la opción (*shuffleCases* activada), puede ocurrir que se haga el reemplazo ($1 \leftrightarrow 1$) o ($1 \leftrightarrow 2$). Suponiendo el último caso, el ejercicio quedaría finalmente como se muestra en el Listado 4.

```
<exerciseInstance id="1">
  <v name="expresion">p \neg q</v>
  <v name="R1_valor">Es fbf</v>
  <v name="R1_solucion">0</v>
  <v name="R2_valor">No es fbf</v>
  <v name="R2_solucion">1</v>
</exerciseInstance>
```

Listado 4. Ejemplo de instancia del ejercicio después de ser aleatorizado

5. Implementación

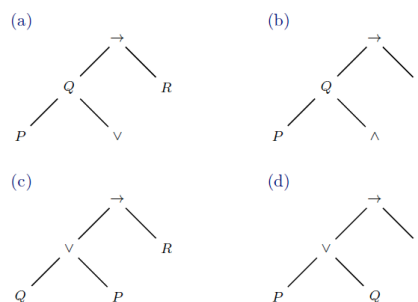
El sistema ha sido implementado utilizando *MiKTeX* como motor de compilación de *LaTeX*, en su versión portable, lo que permite hacer una aplicación web que pueda ser distribuida sin necesidad de dependencias del sistema operativo. Además, la aplicación web ha sido implementada en Java *Servlet* y *JSP* para llevar a cabo la implementación de la interfaz del usuario. Para la gestión de plantillas y reemplazo de variables se ha utilizado una librería Java llamada *FreeMarket*. La aplicación ha sido publicada bajo el nombre de *Acrologic* con licencia GPL en el portal *sourceforge* y puede ser descargada gratuitamente en la siguiente dirección: <https://sourceforge.net/projects/acrologic/>. Los ejercicios diseñados para la asignatura de lógica computación no son distribuidos, ni han sido publicados bajo ninguna licencia.

La Figura 3 muestra un ejemplo de ejercicio que ha sido generado automáticamente para el usuario registrado en el sistema y entregado al

mismo a través de Internet mediante un fichero PDF que se abre automáticamente en el ordenador del alumno usando el tipo mime apropiado.

EJERCICIO 1. Observa cada árbol y explica si puede ser un árbol sintáctico que se corresponda con la siguiente expresión¹:

$$(P \vee Q) \rightarrow R$$



¹Pincha sobre la letra de cada uno para conocer la respuesta.

Figura 3. Ejemplo de ejercicio generado

6. Discusión

La personalización hacia los alumnos y la creación de materiales que fomenten el trabajo autónomo del alumno son dos de los pilares fundamentales en los que se basa el EEES. El sistema propuesto en el presente artículo permite ser utilizado en un gran abanico de metodologías docente que contemple actividades en clase o fuera de ella, la realización de prácticas y boletines, la construcción de tutoriales o ejercicios de autoevaluación en función del nivel de cada alumno o la realización de exámenes tanto on-line como en papel. Sin duda es un sistema flexible que permite ahondar en innovaciones docentes que utilicen las nuevas tecnologías dentro de la enseñanza, de forma que conformen una parte activa dentro de la metodología de aprendizaje del alumno. Nótese que esto es especialmente relevante en carreras como la Ingeniería Informática en la que, en general, conviene mostrar con el ejemplo la utilización de los ordenadores como medios de enseñanza.

7. Conclusiones

Se ha propuesto un sistema para la generación de boletines y tutoriales de forma personalizada para

los alumnos utilizando para ello un repositorio de ejercicios extensible. El sistema tiene una fuerte componente de seguridad para el plagio y el engaño de los alumnos. Además, se ha propuesto un mecanismo de representación de los repositorios de ejercicios que ha sido probado con éxito y validado en la implementación que ha sido llevada a cabo del sistema.

Como trabajo futuro se pretende diseñar e implementar un sistema para recibir las respuesta de los ejercicios de los alumnos (aquellas que no sean tutoriales) e insertarlas automáticamente en una base de datos con dos fines: disponer información de todos los alumnos sobre el tipo de preguntas que son capaces de resolver (y obtener así un perfil de la clase) y disponer información individualizada (y poder así tener el perfil de alumno). Estas informaciones permiten generar nuevas versiones de los boletines incluyendo una componente más pragmática que permita adaptar los nuevos boletines generados a las necesidades reales de los alumnos. También esperamos aumentar el sistema con nuevas funcionalidades como son el acceso al PDF con contraseña, el establecimiento de tiempo límite para realizar el boletín (después del cual los resultados se enviarán automáticamente al profesor o a una base de datos), la imposibilidad de abrir el fichero PDF más de un vez, y, en definitiva, todas aquellas característica que sería deseables para adaptar el sistema de tal forma que contribuya en la metodología docente del aula y el uso de las TIC en la realización de exámenes con un cierto grado de seguridad.

Agradecimientos

Los autores quieren agradecer a la Universidad de Murcia y a la Comunidad Autónoma de la Región de Murcia su soporte para llevar a cabo esta investigación bajo la convocatoria de ayudas para proyectos de innovación docente con TIC para el curso 2010/2011. Además, los autores también quieren agradecer a la Fundación Seneca por soportar el trabajo de Jose Maria Alcaraz Calero bajo las beca post-doctoral 15714/PD/10.

Referencias

- [1] Holohan Edmond, Melia Mark, McMullen Declan, Pahl Claus. *The Generation of E-Learning Exercise Problems from Subject Ontologies*. Sixth IEEE International Conference on Advanced Learning Technologies (ICALT'06). pp. ISBN: 967-969. 0-7695-2632-2
- [2] Soler Masó, Josep; Boada i Oliveras, Imma; Prados Carrasco, Ferran; Fabregat, Ramon.; Poch Garcia, Jordi. Experiencia docente en diseño de bases de datos con la ayuda de herramientas de e-learning. Jornadas de Enseñanza universitaria de la Informática. JENUI 2009
- [3] Rué Joan. *Enseñar en la Universidad: El EEES como Reto para la Educación Superior*. Narcea, 2007.
- [4] Castel de Haro M^a Jesús, Villagrà Arnedo Carlos, Palomino Benito Juan Alejandro. *Materiales Docentes Con Los Que Aumentará Tu InterEEES Por Estudiar Lógica*. Jornadas de Investigación en Docencia Universitaria : la construcción colegiada del modelo docente universitario del siglo XXI. pp 164. 2008. ISBN 978-84-691-4559-3
- [5] Lu Jie. *Personalized e-learning material recommender system*. International conference on information technology for application. pp. 374-379, 2004.
- [6] Lathrop Ann, Foss Kathless. *Student cheating and plagiarism in the Internet era: A wake-up call*. Libraries Unlimited. 2000.
- [7] González Boticario Jesús, Gaudioso Vásquez Elena. *Aprender y formar en Internet*. Thmoson Learning. 2001
- [8] Valero García, Miguel; Vaquerizo García, Belén. Puzzles mejorados con mapas conceptuales. Jornadas de Enseñanza universitaria de la Informática. JENUI 2009.
- [9] Story D. P. The AcroTEX eDucation Bundle. 2006. Available at <http://www.acrotex.net/>

Enseñanza-aprendizaje en Visión por Computador: un enfoque motivador hacia el futuro profesional

María Guijarro Mata-García	Gonzalo Pajares Martinsanz	P. Javier Herrera Caro	Jesús M. de la Cruz García
Dpt. Ingeniería del Software e Inteligencia Artificial	Dpt. Ingeniería del Software e Inteligencia Artificial	Dpt. Ingeniería del Software e Inteligencia Artificial	Dpt. Arquitectura de Computadores y Automática
Facultad de Informática	Facultad de Informática	Facultad de Informática	Facultad de Informática
Universidad Complutense	Universidad Complutense	Universidad Complutense	Universidad Complutense
28040 Madrid	28040 Madrid	28040 Madrid	28040 Madrid
mguijarro@fdi.ucm.es	pajares@fdi.ucm.es	pjherrera@pdi.ucm.es	jmcruz@fis.ucm.es

Resumen

En este trabajo se exponen los diferentes enfoques didácticos aplicados en relación al proceso enseñanza-aprendizaje en educación superior cuyo eje central gira en torno a la materia Tratamiento de imágenes y Visión por Computador, estructurándose según dos itinerarios: grado y postgrado. Alrededor de esta materia se han planteado una serie de actividades en base al interés del alumno y orientados hacia su futuro profesional, con la participación y colaboración activa de diversas instituciones, organismos de investigación y empresas del sector productivo. El alumno acaba relacionando los contenidos formativos con aplicaciones prácticas reales, incluyendo actividades y desarrollos del mundo profesional. De esta manera se ha proporcionado una perspectiva de aproximación entre el mundo universitario y profesional, que sin duda ha contribuido a dotar a los contenidos de una utilidad realista.

Summary

In this paper we describe the different teaching approaches applied in relation to the teaching-learning process in higher education whose central axis is structured around image processing and computer vision, structured along two approaches: grade and postgrade. Around this area have been raised a number of activities based on student interest and oriented to his professional future, with the active participation and collaboration of various institutions, research organizations and production companies. The student finish relating

learning contents to real practical applications, including activities and developments from the professional world. This has provided a perspective of approximation between the academic and professional world, which has undoubtedly helped to give the contents of a realistic value.

Palabras clave

Enseñanza-aprendizaje, aprendizaje activo, pbl, visión, aprendizaje colaborativo

1. Motivación

Hoy en día, los docentes, nos enfrentamos a una palpable evidencia, como es la acusada desmotivación de los alumnos y la consecuente disminución de su rendimiento académico, tanto en la formación pre-universitaria como universitaria, viéndose reflejada dicha carencia en su futuro profesional. Esta tendencia permite plantearnos si las metodologías de enseñanza-aprendizaje tradicionales, que en otros momentos han dado buenos resultados, son actualmente apropiadas para afrontar dicha problemática. Quizás el auge tecnológico esté en el punto de mira del proceso, ya que en la actualidad los alumnos disponen de abundantes recursos tecnológicos, sintiéndose muy cómodos con su uso, pues participan en su vida cotidiana de comunicación y ocio. Por otra parte, suele ser frecuente la pregunta ¿esto para qué sirve?, suscitada quizás por la falta de horizonte claro respecto de los contenidos que se pretenden transmitir. En este trabajo planteamos un modelo educativo orientado a alumnos universitarios, en

el que el aprovechamiento de los recursos tecnológicos junto con la perspectiva de utilidad otorgada a la materia, gracias a la colaboración institucional y de empresas trata de incrementar el interés en el proceso enseñanza-aprendizaje, mejorando considerablemente el rendimiento. El proceso global se plantea desde una aproximación de aprendizaje colaborativo basado en problemas reales [3], con el horizonte de la utilidad de los aprendizajes en el mundo laboral gracias al apoyo institucional y de las empresas, que consiguen la suficiente motivación del alumno y su positiva predisposición al aprendizaje. El paradigma pedagógico planteado se fundamenta en las teorías sobre el constructivismo y el aprendizaje activo de Piaget [5], el socio-constructivismo de Vygotsky [7] y las teorías del aprendizaje en situación de Lave y Wenger [2].

Los manuales pedagógicos presentan el proceso enseñanza-aprendizaje desde dos puntos de vista bien diferenciados: colectivo e individualizado. Desde el punto de vista colectivo, todos los alumnos reciben y realizan el mismo tipo de procesos de enseñanza y aprendizaje, mientras que desde el individual los procesos son específicos para cada alumno. El Espacio de Educación Superior (EEES) incide de forma importante en este último.

Pues bien, bajo el planteamiento de la investigación desarrollada, se ofrecen a los alumnos una serie de actividades con propuestas diferenciales, que tienen como finalidad reforzar el aprendizaje individualizado a la vez que se simultáneamente también se muestran perspectivas de futuro profesional.

Este trabajo se diseña de forma que el alumno al estudiar los contenidos ve su aplicación práctica, relacionándolos con las posibles salidas profesionales que conllevan. Simultáneamente se ofrece una doble orientación de futuro profesional en dos ámbitos concretos: industria e investigación. Se cuenta con la colaboración externa de dos tipos de instituciones con los que se viene trabajando desde hace varios años, en concreto: dos Centros de Investigación y dos Empresas Tecnológicas. Los Centros de Investigación son: a) el Centro de Automática y Robótica (CAR) del Consejo Superior de Investigaciones Científicas a través del Grupo de Percepción Artificial [9]; b) el Centro de Investigación Forestal (CIFOR) del Instituto de

Investigación Agraria y Alimentaria (INIA) [10]; c) el Centro de Domótica Integral (DeDint) [11]. Las empresas son: a) TCP Sistemas e Ingeniería [12] y b) Digital Image Pro (DIMAP) [13].

Como se ha mencionado previamente, el presente trabajo se centra en una materia específica, cual es la Percepción Computacional (PC) desde un doble punto de vista: el Tratamiento de Imágenes y la Visión por Computador. Estas dos materias se ofertan respectivamente en los actuales estudios de Ingeniería Superior (futuro Grado, término al que nos referiremos de aquí en adelante) y en el Master de Posgrado dentro de los estudios oficiales de Informática.

Un alumno en el nivel de Grado dispone de tres itinerarios posibles: la salida exclusiva hacia el mundo empresarial, la salida hacia el mundo empresarial con perspectivas de realización de estudios de Posgrado o la salida hacia la investigación, cuya continuación ha de ser necesariamente el Posgrado.

Por otra parte, al finalizar los estudios de Posgrado existen otros dos nuevos itinerarios: uno en el que el alumno opta por una salida exclusivamente empresarial o un segundo en el que el alumno opta claramente por la investigación con la incorporación al programa de doctorado para la realización de su Tesis Doctoral.

En este artículo se presentan, junto con la descripción del experimento, los resultados preliminares obtenidos hasta la fecha en relación con los objetivos anteriores. La estructura del artículo es la siguiente. La sección 2 realiza una descripción completa sobre la metodología en relación al escenario planteado. Se introducen las asignaturas que han participado en la experiencia, así como las instituciones participantes. En la sección 3 se plantean los objetivos relativos a la experiencia. En la sección 4 se revisa la evaluación de resultados, describiendo primeramente los recursos con los que se ha contado, así como las tareas que han surgido, para finalmente mostrar los resultados obtenidos. La sección 5 se dedica al análisis de la utilidad práctica de este trabajo bajo la perspectiva tanto del alumno como del profesor. Finalmente en la sección 6 se discuten algunas conclusiones y el trabajo futuro.

2. Metodología

La estructura de este trabajo mantiene las dos vertientes del proceso enseñanza-aprendizaje: global e individual. Se ha diseñado de forma que el alumno al estudiar los contenidos ve su aplicación práctica, relacionándolos con las posibles salidas profesionales que conllevan. Se ofrece una doble orientación de futuro: industria e investigación.

Las colaboraciones pasadas y futuras con las instituciones con las que trabajamos tienen en común el hecho de que en todos los proyectos, el Tratamiento de Imágenes y la Visión por Computador constituyen una parte esencial de los mismos. Por ello, la experiencia innovadora docente que se plantea se centra en dichas materias curriculares. Así, se proporciona a los alumnos una visión de futuro profesional tanto hacia la investigación como hacia la industria.

2.1. Contextualización de las asignaturas

Los estudios y materias sobre los que se han desarrollado los trabajos experimentales del nuevo modelo educativo han sido:

- Master oficial de investigación en Informática: Percepción Visual Artificial (PVA) y Proyecto Fin de Master (PFM)
- Master oficial en Ingeniería de Sistemas y de Control: Visión por Computador (VPC), conjuntamente con la UNED.
- Ingeniería Informática: Tratamiento de Imágenes Digitales (TAID) y Sistemas Informáticos (SI)
- Título Propio UCM: Experto Universitario en Tratamiento y Aplicaciones de Imágenes digitales y Visión por Computador (ETVC)
- Becas colaboración del Ministerio de Educación (BCME)

Las materias PVA y TAID se han sometido a evaluación voluntaria en el curso 2009/10 en el programa DOCENTIA [14] habiendo sido de las mejor evaluadas en esta universidad y recomendadas para su inclusión en un proyecto de innovación docente en la UNED por recomendación del propio Vicerrectorado de Desarrollo y Calidad de la Docencia de la Universidad Complutense.

Aunque la Visión por Computador constituye la materia central de la experiencia, existe una

relación interdisciplinaria con otras materias encuadradas en diferentes planes de estudio:

- Másteres oficiales de investigación en Informática y de Ingeniería de Sistemas y Control en la UNED: Optimización en Control de Procesos (OCP), Control Inteligente (CIN), Sistemas Bioinspirados (SIB) y Proyecto Fin de Master (PFM).
- Máster oficial en Físicas UCM: Modelado y Simulación (MyS)
- Ingeniería Informática UCM: Modelado y Simulación de Sistemas (MSS), Arquitectura de Computadores (ARC), Inteligencia Artificial Aplicada al Control (IAC)
- Grado Gestión Empresarial y Tecnología Informática: Programación Orientada a Objetos (POO), Laboratorio de Programación de Sistemas (LPS).

En la Figura 1 se muestra la relación entre instituciones y materias según las opciones de salida: industrial y profesional. En la parte central aparecen las materias que constituyen el núcleo en torno a la Visión por Computador (PVA y TAID).

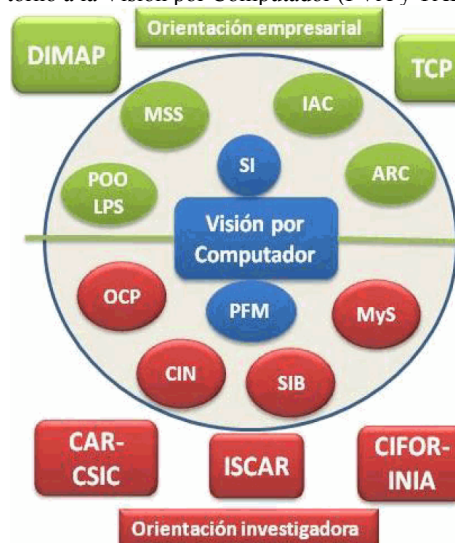


Figura 1. Relación entre materias principales y secundarias: salidas profesionales

3. Objetivos del proyecto

Los objetivos que se plantean bajo la aplicación de esta nueva metodología están todos encaminados a motivar al alumno en relación a su

proceso enseñanza-aprendizaje, y también a adquirir por parte del profesorado buena práctica en lo que respecta a la transmisión del conocimiento en contacto con la realidad del mercado laboral tanto desde el punto de vista de la investigación como desde la perspectiva industrial. Dentro de estos objetivos se encuentra el aprovechar la experiencia realizada con los alumnos en años anteriores, valorando los resultados y optimizando los procesos de enseñanza-aprendizaje a través de los guiones de prácticas ya elaborados. Un segundo objetivo consiste en aprovechar hasta donde sea posible los recursos tanto materiales como humanos, puestos por las instituciones participantes en el proyecto, para facilitar la tarea de aprendizaje. Un tercer objetivo de importancia consiste en aprovechar la experiencia profesional proveniente tanto por parte del profesorado involucrado como de los profesionales pertenecientes a las mencionadas instituciones con el fin de orientar al alumno en relación a las posibles salidas profesionales, propiciando una visión de futuro a la vez que se les proporcionan determinadas pautas para su inserción en el mundo laboral. Para esto se aplican las técnicas de aprendizaje basado en problemas, PBL, puesto que el contacto con la realidad de problemas y ejercicios permite al alumno aprender a entender y a enfocar las futuras situaciones a las que se va a enfrentar. Para ello se proporcionan al alumno pautas de trabajo en relación a las actividades relacionadas con los proyectos en las empresas e instituciones, situándole muy cerca de la realidad profesional. La finalidad es la consecución del interés motivacional individual por medio de las actividades programadas, así como por el diseño de prácticas adaptadas a los intereses individuales.

Tanto las charlas como las prácticas han sido adaptadas según el temario de cada asignatura, relacionando así, los conceptos teóricos con su aplicación práctica, bien sea en la empresa o en centros de investigación. Con esta metodología se consigue el objetivo fundamental de despertar el interés por parte del alumnado.

Con carácter específico, los objetivos que se plantean se concretan en los que se mencionan a continuación, distinguiendo entre los orientados puramente hacia la investigación, cuya motivación viene impuesta por los proyectos en colaboración con CAR y CIFOR y aquellos otros cuya

orientación tiene un marcado carácter industrial propiciado por la colaboración con TCP y DIMAP. Por tanto, en lo que sigue se distinguen entre objetivos de investigación e industriales, identificados con la nomenclatura INV e IND respectivamente, seguidos de un indicador numérico.

A nivel de investigación se centran en:

INV-1: consiste en adaptar las prácticas obligatorias y las opcionales propuestas en las asignaturas orientándolas a los métodos y técnicas previstas en los proyectos de investigación con CAR y CIFOR. Este objetivo está orientado específicamente hacia las materias PVA, VPC, PFM y ETVC.

INV-2: se propicia la participación de los alumnos en las reuniones de seguimiento de los proyectos con CAR y CIFOR, exclusivamente para PFM en los que el tema seleccionado constituye el objeto de la reunión.

INV-3: se organizan conferencias y charlas sobre un tema de investigación, que se imparten por investigadores de CAR y CIFOR. Orientadas principalmente hacia las asignaturas de los Másteres y PFM, si bien incentivando y propiciando la participación del resto de alumnos en las demás asignaturas involucradas en la experiencia así como la libre asistencia, que se ofrece al conjunto de la comunidad universitaria.

INV-4: se organizan charlas-coloquio sobre las salidas profesionales en investigación, impartidas por antiguos alumnos, ahora trabajando como investigadores en CAR. Abiertas a toda la comunidad universitaria.

INV-5: se programan visitas a las instalaciones y los sistemas bajo investigación en el CAR.

A nivel de la industria se distinguen:

IND-1: se adaptan las prácticas obligatorias y opcionales de las asignaturas y se proponen otras nuevas en relación a los métodos y técnicas previstas en los proyectos con TCP. Este objetivo está orientado específicamente hacia las materias TAID, SI, ETVC y BCME.

IND-2: se propicia la participación de los alumnos en las reuniones técnicas de seguimiento de los proyectos con TCP, exclusivamente para alumnos de SI y BCME que se encuentren trabajando en los temas objeto de la reunión.

IND-3: se organizan charlas-coloquio sobre las tecnologías desarrolladas en TCP en relación a

los proyectos mencionados, incidiendo en los métodos de trabajo y las relaciones con empresas del sector a nivel internacional. Orientado a todos los alumnos y a la comunidad universitaria con especial hincapié para los alumnos de TAID, SI y BCME.

IND-4: se organizan charlas-coloquio sobre las perspectivas profesionales, actividades desarrolladas o búsqueda de financiación en DIMAP. Abiertas a toda la comunidad universitaria.

IND-5: se programan visitas a las instalaciones y se facilitan demostraciones de sistemas propios de DIMAP. Orientadas a alumnos principalmente de TAID, SI y BCME debido a las limitaciones de espacio.

4. Evaluación de resultados

En esta sección, en primer lugar se enumeran los recursos materiales con los que se han contado para el desarrollo de las actividades. Seguidamente, se describe la metodología seguida junto con el plan de trabajo, basada en las tareas derivadas de los objetivos.

4.1. Recursos disponibles.

La relación entre las materias y los responsables de las mismas es la que se muestra a continuación, distinguiendo entre el grupo de materias que constituyen el núcleo que se presenta, denominado grupo principal y las que orbitan en torno a ellas, que se identifican como grupo secundario. Las siglas que identifican a los responsables de las materias no tienen mayor finalidad que la mera asignación de materias con los miembros del equipo docente involucrado en el proyecto.

Grupo principal:

GP es el coordinador responsable de las materias que constituyen el núcleo del mismo, desarrollando actividades docentes en PVA, TAID, SI, VPC, ETVC y BCME. En ETVC participan todos los miembros solicitantes. A este grupo pertenecen los PFM en el que participan todos los miembros del equipo. SI se coordina entre dos grupos de alumnos y se diseña según los proyectos de navegación autónoma de robots.

Grupo secundario:

- JC es responsable directo de OCP.

- MS es responsable de CIN, SIB y MyS, compartiendo con PH las materias MSS e IAC.
- JR es responsable de las materias ARC y SI.
- MG es responsable de las materias POO y LPS.

Se cuenta con los recursos propios de las Facultades de Físicas e Informática donde se desarrollan las actividades relacionadas con dichas materias: Aulas, Laboratorios, Sala de Grados para conferencias, Sala de Reuniones para seguimiento de proyectos con empresas.

Se utilizan los recursos disponibles y adquiridos bajo la cobertura de los proyectos de investigación, destacando: equipos de visión artificial dotados con las respectivas cámaras, computadores y programas informáticos, Figura 2.

Se cuenta con los recursos que las empresas y centros de investigación ponen a disposición en las visitas programadas, incluyendo espacios y personal técnico-científico que explica el funcionamiento de los equipos y los métodos de desarrollo.

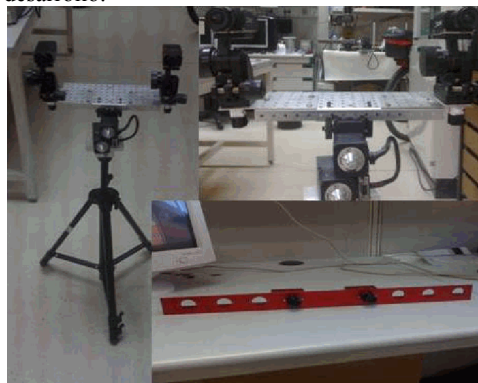


Figura 2: Equipamiento en Laboratorio

4.2. Tareas: plan de trabajo

La planificación temporal de tareas se establece en función de la duración de un curso académico, especificando en cada una de ellas la duración temporal de la misma en relación al curso académico 2009/2010.

Tarea 1: Exposición a los alumnos de los programas de las materias pertenecientes al grupo principal, así como las relaciones interdisciplinarias entre éstas y las pertenecientes al grupo secundario. Esta tarea se lleva a cabo en todas las materias involucradas en el proyecto.

Duración: 4 a 8 de Octubre

Responsables: GP

Tarea 2: Exposición a los alumnos de los programas de las materias pertenecientes al grupo secundario, así como las relaciones interdisciplinarias entre éstas y las del grupo principal.

Duración: 4 a 8 de Octubre

Responsables: JC, JR, MS, MG, PH

Tarea 3: Presentación de los proyectos de investigación desarrollados y la relación de las tareas a desarrollar en ellos con las materias del grupo principal.

Duración: 11 a 22 de Octubre

Responsables: GP, MG, PH

Tarea 4: Realización de un sondeo por escrito al finalizar las exposiciones anteriores sobre el contenido de lo expuesto y el grado de interés de los alumnos.

Duración: 11 a 22 de Octubre

Responsables: GP, MG, PH

Tarea 5: Adaptación de las prácticas obligatorias desarrolladas para este proyecto en función de la mayoría de intereses detectados en los alumnos relativo a las siguientes materias del grupo principal: PVA, VPC, TAID y ETVC. Esta tarea se inicia al finalizar las anteriores y se prolonga hasta completar el conjunto previsto para el curso.

Duración: 25 Octubre hasta 26 de Noviembre

Responsables: GP, MG, PH

Tarea 6: Definición de los trabajos a desarrollar por los alumnos que han elegido PFM en relación a los proyectos conocidos y los intereses individuales, teniendo en cuenta su relación interdisciplinar con las materias del grupo secundario.

Duración: 25 Octubre hasta 5 de Noviembre

Responsables: GP, JR, JC, MS, MG, PH

Tarea 7: Definición de los proyectos de SI en conjunto con los grupos asignados. Esta tarea se realiza de forma individualizada a nivel de tutorías.

Duración: 4 a 8 Octubre

Responsables: GP, JR

Tarea 8: Definición y diseño de las prácticas personalizadas, es decir, las que opcionalmente elegirán los alumnos como complemento a las obligatorias. Esta tarea contempla la relación interdisciplinar entre las materias PVA, VPC, TAID y ETVC del grupo principal que son las

destinatarias directas y las materias del Grupo secundario. Se fomentará la realización de prácticas en grupos con el fin de proporcionar pautas de trabajo en grupo tal y como es habitual en empresas y centros de investigación, facilitando así la inserción profesional.

Duración: 29 Noviembre a 10 Diciembre.

Responsables: GP, MG, PH

Tarea 9: Definición de los trabajos a desarrollar en BCME mediante la orientación tutorizada directa.

Duración: 1 a 5 Noviembre (su ejecución comenzó en Enero).

Responsables: GP, MG, PH

Tarea 10: Ejecución de las prácticas y trabajos en función de la planificación de las diferentes materias involucradas, dependiendo de su carácter cuatrimestral o anual.

Duración: todo el curso.

Responsables: GP, PH

Tarea 11: Participación en las charlas, coloquios o conferencias impartidas por los profesionales de las empresas o centros de investigación involucrados en los proyectos anteriormente mencionados.

Duración: según programación en el periodo de ejecución de las materias.

Tarea 12: Participación en reuniones de seguimiento de los proyectos, se hará de forma limitada en cuanto al número de alumnos y en función de sus opciones elegidas. Sujetas a las cláusulas de confidencialidad de las empresas.

Duración: según planificación de proyectos.

Tarea 13: Realización de visitas guiadas, según el siguiente plan: a) laboratorio en Físicas; b) laboratorios de robots del centro CAR en Arganda del Rey; c) CeDint para tomar contacto con las instalaciones de Realidad Virtual; d) instalaciones del CIFOR-INIA; e) DIMAP para conocer un simulador de visión 3D.

La responsabilidad de las tareas 11 a 13 recae en todos los miembros del grupo.

Formando parte de la metodología, conviene resaltar que al inicio del curso se exponen los objetivos globales, que en cada tema se definen los objetivos específicos y las tareas a realizar fijando el plazo de entrega de las mismas, y que la evaluación de los trabajos será continua procurando así la motivación del alumno.

5. Utilidad Práctica

La utilidad práctica se centra en el proceso enseñanza-aprendizaje dentro del EEES tanto desde el punto de vista del alumno como del profesor, poniendo el énfasis en la motivación del alumno y el trabajo y esfuerzo individualizado.

5.1. Perspectiva del alumno

El alumno relaciona los contenidos formativos con aplicaciones prácticas reales, que le proporcionan un grado de satisfacción por la utilidad de los contenidos [6], bajo una toma de contacto con el mundo laboral en dos de sus vertientes: empresa e investigación.

Adquieren hábitos de trabajo individual y en equipo como preparación a su futura incorporación al mundo profesional, donde asumen pautas importantes para desarrollar sus tareas de forma autónoma a la vez que son capaces de colaborar e integrar sus desarrollos en los equipos de trabajo.

Se proporciona una perspectiva de aproximación entre el mundo universitario y profesional, dotando a los contenidos de una utilidad real y se orienta al alumno en su futura inserción en el mundo laboral.

5.2. Perspectiva del profesor

El profesor debe realizar un estudio entre las posibilidades que ofrecen los métodos de trabajo tradicionales dentro del proceso de enseñanza y la incorporación de nuevas metodologías que traspasan los límites del aula. De esta manera se facilita la toma de contacto con la realidad empresarial o de investigación, permitiéndole trasladar al grupo de alumnos los contenidos derivados de la experiencia profesional en el mundo empresarial e investigador.

El profesor permite la adaptación de ciertos contenidos según las necesidades empresariales y de investigación, e inversamente, posibilita y orienta a los centros involucrados sobre las posibilidades de aplicación de nuevas tecnologías. Posibilita la tan ansiada transferencia tecnológica Universidad-Empresa-Investigación.

Se facilita la interdisciplinariedad de materias y la colaboración inter-departamental e incluso inter-centros.

En el grupo participan dos miembros recientemente doctorados (MG, PH), que

proceden de las disciplinas hacia las que se orienta el proyecto, que les permite completar su formación y ofrecen una garantía de “proximidad” a los alumnos. La participación de recién posgraduados supone un elemento fuertemente motivador hacia el alumno en cuanto que ve la proximidad de su situación con respecto a personas muy cercanas en el tiempo.

A su vez, la experiencia de haber involucrado al alumno en el mundo tanto empresarial como investigador ofrece al profesor un punto de vista diferente al originado por las enseñanzas clásicas de esta materia. Se puede evaluar el grado de satisfacción por parte del alumnado en las encuestas realizadas al final del curso académico, donde las puntuaciones ofrecidas a los profesores responsables así como a la propia asignatura han sido satisfactorias. La mayor parte de los alumnos agradece la involucración en el mundo empresarial, al ser ese el camino que posteriormente eligen, y resultado de esta experiencia la contratación de alumnos en nuevos proyectos bajo esta temática.

6. Conclusiones y trabajo futuro

Este modelo se ha aplicado previamente en los cursos académicos 2008/09 y 2009/10, estando actualmente en el tercero donde las estadísticas respecto de la matriculación han resultado ser: en TAID, al tratarse de un seminario con laboratorio, el número máximo de alumnos previsto es de 40, habiendo conseguido esta cifra en los primeros días de matrícula, siendo la primera asignatura cerrada por llegar a dicho cupo. En PVA se ha batido récord de matrícula con 25 alumnos en 2009/10 y 21 alumnos en 2010/11, siendo también una de las asignaturas más demandadas en el Master. En PFM se han presentado 7 trabajos en 2009/10 y en el curso 2010/11 existe una previsión de otros tantos. En estas dos últimas se pretende mantener esta línea. En SI se han beneficiado 6 alumnos por curso. En VPC, se ha conseguido una matrícula de 20 alumnos en 2010/11 siendo de nuevo una de las más demandadas. En ETVC se han matriculado 8 alumnos y finalmente en BCME la concurrencia es de un alumno por curso. La participación a través de las materias del grupo secundario se dirige a unos 150 alumnos, de los cuales la participación en trabajos interdisciplinares y

actividades complementarias se sitúa en torno a un 30%, haciendo un total de unos 145 alumnos. Las actividades de visitas, charlas y conferencias se ofertan abiertas al resto de la comunidad educativa.

El modelo propuesto es fácilmente exportable a otras materias, asignaturas o titulaciones. El único requisito es disponer de soporte externo relacionado con el mundo laboral o investigador. Se pretende dar continuidad a esta experiencia en cursos sucesivos.

Se plantea una experiencia multidisciplinar en la que concurren diversas materias y diversas áreas además de las relaciones con las entidades participantes. Al tratarse de alumnos que se encuentran en sus últimos cursos, la motivación en ellos es bastante alta. Cabe destacar que en determinados temas de investigación, los alumnos no muestran tanto interés, cosa que no suele ocurrir con los proyectos dirigidos por empresas.

Se está colaborando con un grupo de la Facultad de Educación de la UNED sobre perspectivas docentes de futuro para elevar una propuesta al Ministerio de Educación sobre la base del proceso enseñanza-aprendizaje dentro del EEES. Esta participación ha sido promovida por este Vicerrectorado de Desarrollo y Calidad de la Docencia.

Las asignaturas centrales de la experiencia se han sometido a evaluación dentro del programa DOCENTIA (PVA y TAID) durante los dos cursos previos con resultados satisfactorios, estando en el presente curso sometida a evaluación dentro del mismo programa. A su vez se pretende evaluar de manera continuada durante el curso, las asignaturas así como las actividades realizadas en ellas, de manera particular.

Agradecimientos

Agradecer al proyecto DPI2009-14552-C02-01, del Ministerio de Educación y Ciencia del Plan Nacional de I+D+i, que ha servido de base para la realización de las actividades docentes que se desarrollan en este trabajo así como la colaboración y disponibilidad de todo el personal que ha hecho y hace posible la aplicación de este modelo innovador enseñanza-aprendizaje que tan buenos resultados está generando.

Referencias

- [1] Garrison, D.R. *Quality and theory in distance education: theoretical consideration*, in D. Keegan (Ed.), *Theoretical principles of distance education*, Routledge, New York, USA, 1993.
- [2] Lave, J. and Wenger, E. *Situated learning: legitimate peripheral participation*. Cambridge: Cambridge University Press. 1991.
- [3] Johnson, D.W., Johnson, R. T., & Smith, K. A., (1991). *Cooperative learning: Increasing college faculty instructional productivity*. ASHE-ERIC Higher Education Report No. 4. Washington DC, USA: School of Education and Human Development, The George Washington University.
- [4] Oakley, B., Felder, R. M., Brent, R., Elhajj, I., *Turning student groups into effective teams*. New Forums Press, Inc. P.O. Bos 876, Stillwater, OK, 2004.
- [5] Piaget, J. *Science of education and the psychology of the child*. New York: Orion Press. 1970.
- [6] Springer, L., Stanne, M.E., and Donovan, S., *Effects of Small Group Learning on Undergraduates in Science, Mathematics, Engineering and Technology: A Meta Analysis*. Madison, WI: National Institute of Science Education, 1997.
- [7] Vygotsky, L. S. *Mind in society: The development of higher psychological process*. Harvard. Harvard University Press. 1978.
- [8] Sancho-Thomas, P., Fuentes-Fernandez, R., Fernandez-Majon, B. "Learning teamwork skills in university programming courses" *Computers & Education*. 53, 2, 517-531. 2009
- [9] CAR-CSIC: http://www.iai.csic.es/asp/personal_iai.asp
- [10] CIFOR-INIA <http://www.inia.es/inia/>
- [11] CeDint <http://www.cedint.upm.es/>
- [12] TCP Sistemas e Ingeniería <http://www.tcps.com/>
- [13] DIMAP <http://www.dimap.es/>
- [14] <http://www.aneca.es/Programas/DOCENTIA>

Pósteres y Recursos Docentes

COETEST: Corrección óptica de exámenes de test en papel automática, rápida y económica

Darío Álvarez Gutiérrez, Lucas Díaz Sanzo

Departamento de Informática
Universidad de Oviedo
Edificio de Ciencias, Calvo Sotelo s/n
33007 Oviedo
darioa@uniovi.es

Resumen

La realización frecuente de evaluaciones de tipo test en papel a grandes grupos se ve restringida por el esfuerzo necesario para su corrección. COETEST es una herramienta que permite minimizar este esfuerzo reduciendo el trabajo del profesor a la preparación de las preguntas. Con escáneres multihoja económicos, las hojas se digitalizan y las respuestas marcadas se reconocen usando técnicas de visión artificial, reduciendo el tiempo de corrección de varias horas a unos minutos. La herramienta elimina el coste, el retraso y la dependencia de un servicio central especializado, se acelera la publicación de resultados y se reduce el tiempo invertido por el profesor en la realización de estos exámenes.

[COETEST puede descargarse en
<http://petra.euitio.uniovi.es/~darioa/coetest/>]

Summary

Frequent test exams for large groups are sometimes impeded because of the effort needed for the grading process. COETEST is a tool that minimizes this effort, reducing the work of the teacher to the development of the questions. Using budget multi-page scanners, test pages are digitized, and the marks are detected using computer vision techniques. The time used for grading is reduced from hours to minutes. The tool removes the cost, the delay and the dependency on a specialized central service, results are published earlier, and the global effort put by the teacher in test exams is diminished.

Palabras clave

Test, visión por computador, herramienta, corrección automática, grandes grupos.

1. Introducción

Los exámenes de tipo test siguen siendo una de las principales herramientas de evaluación. Requieren un esfuerzo previo por parte del profesor en la preparación de las preguntas, y uno “mecánico” posterior para su corrección. El principal trabajo de corrección es la comparación de las respuestas del alumno con las respuestas correctas, y varía dependiendo del sistema utilizado.

Aunque existen numerosas herramientas para realizar exámenes de test mediante ordenador (incorporadas ya en plataformas de aprendizaje como Moodle [4] o Desire2Learn [1]), la realidad es que no pueden ser aplicadas en la práctica para grandes grupos al no disponer de los recursos informáticos para ello (100 ordenadores / aula), o físicos (disponibilidad de varios laboratorios de ordenadores, habitualmente ya sobrecargados).

En estos casos se sigue utilizando el sistema tradicional de repartir hojas de papel impreso en las que los alumnos marcan las respuestas, que son recogidas por el profesor y corregidas posteriormente (de manera manual o mediante un sistema de detección de marcas ópticas). Estos sistemas requieren un esfuerzo relativamente grande por parte del profesor, lo que hace que el número de pruebas de test realizadas se restrinja.

COETEST es una herramienta que elimina esta barrera de esfuerzo mecánico del profesor en la corrección de exámenes de tipo test tradicionales para los grandes grupos. De esta manera el profesor podrá realizar más exámenes de tipo test si lo estima conveniente al no requerir un esfuerzo mecánico adicional.

El resto del artículo se estructura de la siguiente manera. En primer lugar se discuten los inconvenientes de los sistemas de corrección de test manuales y los de detección óptica de marcas. A continuación se describirá el método de trabajo con la herramienta COETEST y cómo está

implementada. Se presenta luego una estimación del ahorro de tiempo que supone para el profesor su utilización, así como algunos parámetros de su funcionamiento. Se finaliza con las previsiones de trabajo futuro y las conclusiones.

2. Sistemas de corrección de exámenes de test tradicionales

2.1. Corrección manual

En la corrección manual el profesor compara las marcas de cada hoja del test con las respuestas correctas, y lo anota. Habitualmente se hace uso de una plantilla superpuesta en la que se recortan las respuestas correctas.

Los inconvenientes de este sistema se derivan del tiempo invertido en la preparación de las plantillas, y en la tediosa tarea de ir pasándolas por cada uno de los exámenes. Además, el tedio provoca equivocaciones en la corrección

2.2. Corrección por detección automática de marcas con lector específico

Se trata de que un dispositivo automático detecte las marcas del examen. Usualmente se utilizan dispositivos especiales denominados “Lectores Ópticos de Marcas”. Un inconveniente es el elevado precio de estos dispositivos, y la necesidad de utilizar hojas de test especiales para el dispositivo (que tienen un coste apreciable). Adicionalmente, debido al coste, habitualmente existe un servicio centralizado que se encarga del sistema. Esto establece una gran dependencia que introduce sobre todo un retardo incierto en la obtención de resultados (hay que enviar las hojas al servicio, esperar a que toque el turno, que el personal esté disponible, etc.).

2.3. Corrección por detección automática de marcas mediante software

Es una variante más reciente del sistema anterior, en el que se utilizan hojas normales para imprimir el test, que son escaneadas y luego se detectan las marcas mediante un software. COETEST es una herramienta que pertenece a este grupo.

Aunque existen herramientas comerciales (Remark Office OMR [3], PC OMR [6]) y también *Open Source* (Udai Waterloo OMR [7] o queXF [5]) se ha optado por desarrollar una nueva herramienta como proyecto fin de carrera

fundamentalmente por la posibilidad de construirla a medida de las necesidades del autor.

3. Realización de tests con COETEST

COETEST está implementado como aplicación multiplataforma Java de escritorio. El proceso de realización de un examen de test apoyado en la herramienta comprendería los siguientes pasos:

1. *Preparación del examen (profesor)*. El profesor determina el enunciado del examen de test propiamente dicho.
2. *Generación de la hoja de contestación del test (COETEST)*. Usando el apartado de Generación de plantillas de la herramienta se indican unos datos de cabecera, el número de preguntas del test, de respuestas, de modelos diferentes y de casillas para el identificador numérico del alumno (Fig. 1).

EXAMEN DE TEST
Asignatura en cuestión
HOJA DE CONTESTACION DE TEST

Apellidos, Nombre: _____

Modelo: 0 1 2 3

	A	B	C	D	E
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					

Figura 1. Ejemplo de hoja de test en papel

3. *Impresión del examen y de las hojas de contestación (profesor)*. El profesor imprime en papel normal las copias del enunciado del examen y de la hoja de contestación.
4. *Realización del examen (alumno)*. El alumno marca en la hoja de contestación su identificador, el modelo del examen y las respuestas a cada pregunta (Fig. 2).
5. *Recogida y escaneo de los exámenes (profesor)*. El profesor recoge las hojas de contestación y las escanea. Los ficheros JPEG generados se llevan al ordenador.

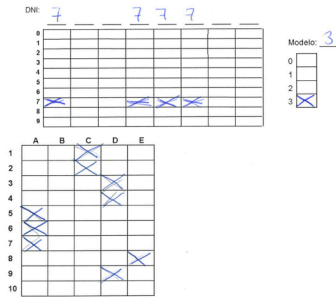


Figura 2. Hoja de test rellena por el alumno

6. *Análisis (COETEST)*. Con el apartado de Análisis, se indican el número de preguntas, de modelos, etc. coincidentes con los originales, así como el directorio de las hojas escaneadas (Fig. 3). El sistema analiza cada hoja y detecta las marcas realizadas, representando las respuestas con caracteres separados por comas, y un guión si no hay respuesta (ej: "a,ab,-,ce,-,-,e").

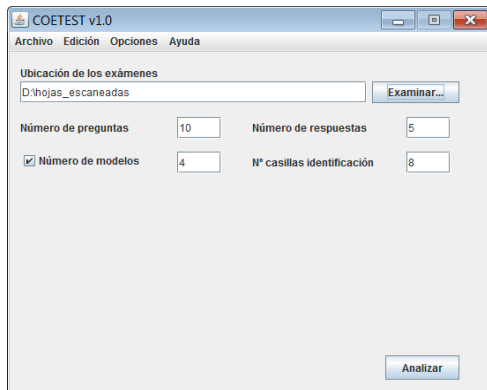


Figura 3. Configuración del análisis

Se genera un fichero de hoja de cálculo (estándar CSV) con los datos detectados (identificador, fichero, respuestas detectadas) (Fig 4).

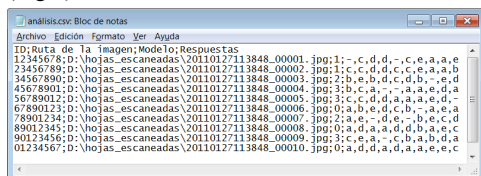


Figura 4. Hoja de cálculo con respuestas detectadas

Existe un panel de control para calibrar el sistema de detección de marcas en función del material escaneado de origen (Fig. 5).

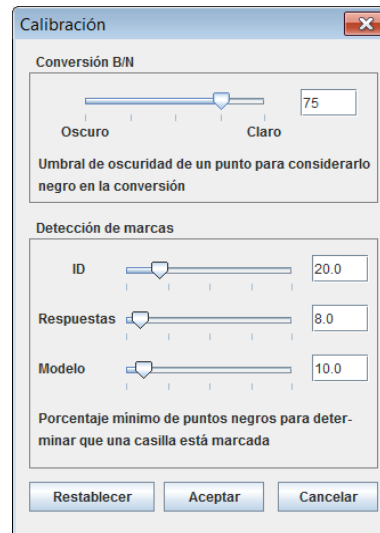


Figura 5. Calibración de la detección de marcas

7. *Corrección (COETEST)*. Con el apartado de Corrección se especifica la clave de respuestas correctas de cada modelo (ej: "b,ab,e,ce,d,ae,e").

El sistema genera un fichero CSV con el número de respuestas correctas, incorrectas y en blanco (Fig. 6).

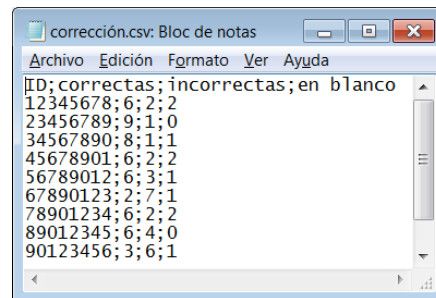


Figura 6. Hoja de cálculo con resultados corregidos

8. *Calificación y presentación de resultados (profesor)*. A partir de estos resultados, el profesor asigna la calificación con los criterios que desee (ej: incorrecta -1 punto, correcta +3), formatea los resultados finales para su presentación y los hace llegar a los alumnos.

4. Proceso técnico de detección de marcas

1. *Recortar zona de marcas en cruz.* En la plantilla se dejan impresas unas cruces en las esquinas para detectar el recuadro donde están los datos y rotaciones en el escaneo de la hoja.
2. *Convertir a blanco y negro.*
3. *Detectar el punto de cruce las marcas.* Se aplica la transformada de Hough [2] para detectar líneas rectas en la imagen, lo que permite luego detectar fácilmente los puntos de cruce que marcan las cuatro esquinas del recuadro de datos.
4. *Corregir rotación.* Los puntos detectados también indican si hay rotación en la imagen, y se corrige aplicando una rotación contraria (hasta 5 grados de desviación).
5. *Recortar casillas.* Sabiendo la configuración del examen y la zona de datos marcada por las cruces, es conocida la posición de cada casilla.
6. *Detección de marcas.* Mediante la evaluación del histograma se decide si una casilla está marcada o no (si el porcentaje de píxeles que superan un determinado umbral de negro dentro de la casilla llega a un valor determinado se da la casilla por marcada).

5. Rendimiento y parámetros de funcionamiento

El ahorro de tiempo frente a una corrección manual es considerable. Como ejemplo, la corrección de 100 exámenes de test de 10 preguntas, que suponía unas 2 horas y 45 minutos, pasa a menos de 10 minutos con COETEST (tiempo de escaneo: menos de 1s. por examen).

Parámetros de funcionamiento del sistema (porcentaje detección correcta cercano al 100%):

- Preguntas por examen: 50 máximo
- Respuestas por pregunta: 6 máximo
- Dígitos de identificación: 10 máximo
- Número exámenes: típicamente ilimitado con imágenes a 200ppp.
- Rotación imágenes: 5° máximo sin error.
- Tiempo de análisis: típicamente 2,5 segundos por examen a 200ppp.

6. Trabajo Futuro

- Acceso al sistema mediante aplicación web.
- Estadísticas de análisis de resultados del test.

- Desarrollo de un algoritmo de calibración del sistema que optimice automáticamente el porcentaje de reconocimiento.
- Mejora del aspecto estético de la hoja generada para el examen.
- Interfaz para la corrección manual de los errores de detección o de origen.
- Incorporación de esquemas de calificación en función de las respuestas acertadas y falladas.

7. Conclusiones

Los exámenes de test tradicionales en papel seguirán siendo necesarios, especialmente en el caso de grandes grupos.

Se ha desarrollado el sistema COETEST, que permite liberar al profesor de las tareas más mecánicas de corrección en la realización de exámenes de test tradicionales en papel. Es significativo el ahorro en tiempo de profesor: casi 3 horas por cada prueba con 100 alumnos.

De esta manera se consigue el objetivo de eliminar barreras externas para la realización de más pruebas objetivas de tipo test.

El rendimiento y funcionamiento alcanzado por el sistema es correcto y sin limitaciones especiales en la práctica.

Referencias

- [1] Desire2Learn. Desire2Learn Measurement and Assessment Tools. <http://www.desire2learn.com/learningenvironment/assessment/>. (8/02/2011)
- [2] Duda, R. O., Hart, P. E. Use of the Hough Transformation to Detect Lines and Curves in Pictures, *Com. ACM*, V15, pp. 11–15, 1972.
- [3] Gravic. Remark Office OMR. <http://www.gravic.com/remark/officeomr/>. (8/02/2011)
- [4] Moodle.org. Moodle Quiz Module. http://docs.moodle.org/en/Quiz_module. (8/02/2011)
- [5] queXF: An Open Source, web based paper form verification and data entry system. <http://quexf.sourceforge.net/>. (8/02/2011)
- [6] SIQI. PC OMR 9.0. <http://www.cai2000.com/>. (8 Febrero 2011)
- [7] UDAI Waterloo OMR. <http://www.cse.iitd.ernet.in/~aseth/udai/OMRProj/README.html>. (8/02/2011)

El placer de las buenas citas

Nieves Pavón Pulido
Dpto. de Ing. Elect., Sist. Inf. y Autom.
Universidad de Huelva
21007 Palos de la Frontera (Huelva)
npavon@dti.uhu.es

Julia González Rodríguez
Dep. de Ing. de Sist. Inf. y Telem.
Universidad de Extremadura
10001 Cáceres
juliagon@unex.es

Joe Miró
Dept. de Matem. e Inform.
Universidad de las Islas Baleares
07122 Palma de Mallorca
joe.miro@uib.es

Resumen

Ya Salomón dijo “No hay nada nuevo bajo el sol”. Quizá no haya ideas nuevas, pero hay gente privilegiada que ha sido capaz de expresarlas de una forma que las hace resonar y que incita reflexionar y profundizar sobre ellas. Hemos escogido, seleccionado y organizado un conjunto de excelentes citas para reflexionar sobre la esencia de la educación.

Summary

Solomon said “There is nothing new under the sun.” There might not be new ideas, but some people have the ability to express them in such a manner that they resound within us and incite us to reflect on them. We have collected, selected, and organized a set of excellent quotations that we hope will allow us to consider carefully about the essence of education.

Palabras clave

Educación, citas, reflexión

1. Introducción

Una buena idea nos mueve, pero una buena idea explicada con claridad especial, con brillo, con ingenio resuena en nuestro interior. Nos incita a reflexionar, a pensar de nuevo. La tenemos siempre presente y resurge inopinadamente una y otra vez, enriqueciendo nuestros argumentos. Una buena cita es un pequeño tesoro.

Por esto muchos recogemos citas de forma formal o informal. Este póster proviene de esta práctica. Queremos compartir algunas de las citas que hemos escogido para nuestro deleite personal ya que aquellas ideas que más nos han movido merecen ser difundidas. Hemos creado una wiki donde hemos ido colocando nuestras citas favoritas (<http://citased.wikispaces.com/>) y hemos escogido

algunas de las mejores para que sirvan como instrumento de reflexión en esta edición de las Jenui.

Es difícil escribir un documento centrado en citas. Por fuerza la estructura es indefinida, y la fluidez se resiente con los continuos cambios de estilo. Creemos que es un precio que vale la pena pagar, que las citas lo merecen. Las citas son las protagonistas de este trabajo. Sólo añadimos lo imprescindible para que reluzcan más.

2. Mundo virtual, Mundo real

Uno de los objetivos del EEES es pasar de conocimientos a competencias, esto es, de formar gente que funciona bien en el mundo virtual del aula a gente que funcione bien en el Mundo Real. Este objetivo implica que el mundo universitario no es real, sino que es la consabida torre de marfil. ¿Es injusta esta concepción que se tiene de nosotros?

No debe serlo cuando gente de gran calibre como Edwin Land fuera del mundo de la informática y Fred Brooks o David Patterson dentro de ella opinan que es un problema que nos debe preocupar. Los profesores somos gente con una forma de pensar especial. Y la lógica es necesaria, pero no lo es todo.

Lo más grave de esto es que esta visión disminuida de lo que es el pensamiento humano aleja a los jóvenes del mundo de la informática. Debemos reconocer que hay vida más allá de nuestro mundo cerebral e intelectual y trabajar para promover el instinto, la creatividad, la comunicación, el conocimiento social. Los libros son necesarios, pero la práctica y experiencias reales, también.

«¿A qué me refiero cuando digo que [en la universidad] un hombre es tratado como un niño? Me refiero a que se le dice, desde el mismo momento de su llegada, que su secreto sueño de grandeza es una mera ilusión [...] No se le dice esto con palabras. Se le comunica de una manera mucho más convincente. Se le muestra, en todo lo que le sucede, que nadie

cree que pudiera hacer una contribución personal y significativa.

Se le dan asignaturas, inmediatamente se le somete a prueba y se le dan exámenes. Ahora, os pregunto, si esto es una preparación para la vida, ¿decidme dónde, en qué lugar del mundo, en qué lugar de la relación con nuestros colegas, en qué lugar del mundo de la industria, en qué lugar jamás, en qué momento de su vida, se le da a una persona esta curiosa secuencia de conferencias preparadas y preguntas preparadas, preguntas de las que se saben las respuestas? ¿Dónde se le vuelve a marcar de esta manera? [...] ¿Cuándo o dónde se le volverá a traer al día del juicio final cada semana?

Si nos preguntamos por qué tan pocos alumnos sobreviven al sistema universitario [...] —sospecho que todo empieza aquí» *E. Land* [5]

[Consejos a sus estudiantes] «Muestra iniciativa, pues la fortuna favorece a los audaces. No esperéis que los profesores os digan lo que tenéis que hacer; si fuéramos buenos gestores probablemente no estaríamos en la Universidad.» *D. Patterson* [9]

«Los procesos de pensamiento de los humanos no son como la lógica matemática de las máquinas. De hecho, si los procesos de pensamientos humanos fueran como la lógica, no hubiéramos tenido que inventarla para ayudarnos a pensar: la lógica es importante *porque* es diferente.

La lógica claramente no es un buen modelo del pensamiento humano. Los humanos tienen en cuenta tanto el contenido como el contexto de un problema, mientras que la fuerza de la lógica y su representación simbólica formal es que el contenido y el contexto son irrelevantes. Introducir el contenido del problema significa interpretarlo en términos concretos y asociarlos otra vez al mundo conocido, con acciones e interacciones reales.» *D. Norman* [8]

«Uno de los obstáculos que tenemos en hacer [esta innovación] son los exámenes. Al final, si los examinamos haciendo que trabajen a mano, es difícil cambiar los estudios para que puedan utilizar los ordenadores durante el curso.» *C. Wolfram* [12]

«Una de las cosas más paralizantes que pueden suceder a un profesor inexperto y poco preparado es tener a un estudiante agresivo levantando su mano y preguntando “¿Para qué sirve todo esto?” Y una de las cosas más irresponsables que un profesor puede

hacer es responder “No lo sé. Eso no es mi problema”. Si no tienes una respuesta para la pregunta de este estudiante, no estas haciendo bien tu trabajo» *S. Krantz* [4]

«¿Cómo puede la informática ser “irrelevante” cuando permea tanto nuestro mundo? Quizá el problema es que nuestro curso tenía muy poca conexión con la informática que tiene lugar en el mundo de nuestros estudiantes. Mientras que los estudiantes se asombran ante la Web, las videoconsolas portátiles y los smartphones, la mayoría de los cursos introductorios presentan a los alumnos los conceptos que están detrás de estas maravillas con números de Fibonacci y las Torres de Hanoi. Lo que los estudiantes veían como informática estaba desconectado con lo que les enseñábamos en nuestra clase de informática.» *M. Guzdial* [3]

«Las necesidades de los estudiantes de tener un contacto directo con los usuarios y experiencia real de implementación argumenta con fuerza hacia más cursos de proyectos y experiencias, incluso a costa de aprendizaje a partir de libros. Las técnicas de análisis y los métodos formales de síntesis son herramientas necesarias, pero los métodos avanzados pueden auto-aprenderse cuando sean necesarias. Los instintos son más difíciles de adquirir.» *F. Brooks* [2]

3. Labor del Profesor

Aprendizaje vs. enseñanza, este es uno de los principios impuestos por el EEES, pero si el profesor ha pasado de ser un transmisor a un facilitador ¿por qué de manera continua nos estamos planteando cuál es su papel? ¿cuál es su labor? Si no es el centro, quizás haya que considerar otros elementos en la valoración del aprendizaje de nuestros alumnos, como el propio plan de estudios o los conocimientos y actitudes iniciales que los alumnos poseen.

*Dejando a un lado la discusión de la importancia de la labor docente, otra pregunta que queda en el aire es **decidir** qué es lo que el profesor debería enseñar o transmitir. Actualmente, dentro de un plan de estudios el saber aparece compartimentado olvidándose de la conexión entre las ideas y las habilidades que permiten la puesta en práctica de los conocimientos y la capacidad de enfrentarse a un desarrollo profesional pleno. Un buen profesor, o*

aquel que desarrolla una buena labor docente, debe ser capaz de transmitir no sólo conocimiento sino las habilidades necesarias para utilizar ese conocimiento. Para ello debe prepararse, entrenarse. Ser profesor es y debe ser una profesión valorada, exige esfuerzo y dedicación y como tal debe estar reconocida y apoyada.

«En el núcleo de las ideas sobre la docencia de la mayoría de los profesores hay un interés en lo que el profesor hace y no en lo que se supone que los estudiantes deben aprender. Según esa concepción habitual, enseñar es algo que los profesores hacen a los estudiantes, normalmente proporcionando verdades de la disciplina. Es lo que algunos escritores llaman un “modelo de transmisión”» *Ken Bain* [1]

«[El movimiento Writing Across Curriculum] presupone que los profesores, y no los planes de estudio ni las acreditaciones, son el centro del proceso educativo y son la clave para la reforma educativa.» *A. Young* [13]

«Un Plan de Estudios sugiere que existen dominios diferenciados de conocimiento, comprensión y habilidades que crean un marco para la enseñanza y el aprendizaje. Las categorías en las que dividimos nuestras experiencias nos son muy importantes. Mas allá de la educación, más allá de las instituciones, la gente trabaja muy cómodamente sin este tipo de categorías. Todos creamos categorías para propósitos distintos, pero la educación divide al mundo en bloques de conocimiento para que podamos enseñar a los niños el material, las ideas, el conocimiento y las habilidades que nosotros consideramos importantes.» *K. Robinson* [10]

«Internet probablemente puede repartir toda la información que se enseña en la universidad, como también puede hacerlo una buena enciclopedia. Por lo tanto, ¿para qué ir a la universidad? Porque hechos aislados no constituyen una educación. El significado no viene sólo de datos. La resolución creativa de problemas depende del contexto, interrelaciones y experiencia. La matriz que la rodea puede ser más importante que los trozos individuales de información. Y sólo un ser humano puede enseñar a alguien la conexión entre las cosas.» *C. Stoll* [11]

«Los profesores excepcionales tratan sus clases, sus discusiones programadas, sus sesiones de resolución

de problemas y demás elementos de su enseñanza como esfuerzos intelectuales formales que son intelectualmente exigentes y tan importantes como su investigación y su trabajo académico. Esta actitud es quizás más patente en las respuestas que nuestros sujetos proporcionan a una cuestión sencilla: «¿Qué te preguntas cuando te dispones a preparar tu docencia?». [...] [Los profesores excepcionales] comienzan con cuestiones sobre los objetivos de aprendizaje para los estudiantes, en lugar de con aquellas que plantean qué debe hacer el profesor.» *Ken Bain* [1]

«Ese [profesor] novato debe dirigirse a algún lado para aprender a enseñar. No puedes aprender a tocar el piano o a esquiar implemente mirando cómo lo hace otro. Y el hecho de haber estado sentado en un aula la mayor parte de su vida no significa que sepa cómo enseñar.» *S. Krantz* [4]

4. El aprendizaje

Queremos pasar del “modelo de transmisión” en el que el profesor proporciona al alumno las “verdades” de la disciplina a uno de aprendizaje, en el que el alumno recoge —absorbe— el conocimiento y lo convierte en suyo. Para poder pasar a este Nuevo Modelo debemos entender mejor que es eso del aprendizaje.

El modelo actual, con clases de teoría y ejemplos prácticos no recoge ni cómo se aprende ni cómo se ha generado el conocimiento. ¿Cómo aprendemos? Esto es una cuestión importante sobre el que quizá pasamos por encima demasiado a la ligera. Es un punto que merece una seria reflexión.

El aprendizaje es un proceso que es a la vez mucho más complejo que lo que pensamos y por otro lado mucho más natural que en lo que lo convertimos en nuestras clases. Para conseguir más, quizá debemos hacer menos y dar más protagonismo a nuestros alumnos.

«Nos engañamos a nosotros mismos si creemos que los comportamientos hábiles son fáciles, que se pueden conseguir sin esfuerzo. Olvidamos los años de puesta a punto, de aprendizaje y práctica que se necesitan para ser hábil incluso en las actividades humanas más fundamentales: comer, caminar, hablar, leer y escribir. Es tentador querer una gratificación instantánea —rendimiento experto inmediato y placer

experimental— pero la verdad es que esto sólo ocurre tras un crecimiento y una afinación considerables.» *D. Norman* [8]

«Tendemos a perder la paciencia y a olvidar las luchas de los poco experimentados. Un profesor que ha estado trabajando, y enseñando, unas ideas durante 20 años no puede entender cómo los estudiantes no recuerdan lo que ya han visto una vez. ¡Una vez! La clave del éxito aquí consiste en desarrollar (o recordar) un poco de sensibilidad hacia el punto de vista del estudiante.» *S. Krantz* [4]

«Siempre se ha hablado de la “Teoría” en contraposición a la “Práctica”, pero tengo mis dudas de que se haya tenido clara la distinción entre una y otra. Yo, que he sido «cocinero antes que fraile», me declaro culpable de haber dedicado clases, que debían haber sido de “prácticas”, a la realización de ejercicios que se limitaban a encasillar un caso particular dentro de un esquema general previamente conocido. Hoy pienso que algunos casos prácticos tendrían que proporcionar la ocasión de desarrollar en el alumno su capacidad de ver el mundo como lo que es, no como nuestras gafas llamadas “Teoría” nos invitan a verlo.» *J. Miró Nicolau* [6]

«[En contraposición al “modelo de transmisión”] los mejores educadores piensan en la docencia como cualquier cosa capaz de ayudar y animar a los estudiantes a aprender. Enseñar es atraer a los estudiantes, diseñando cuidadosamente un entorno en el que ellos aprendan.» *Ken Bain* [1]

«[Arthur C. Clarke] dijo dos cosas interesantes: “Un maestro que puede ser reemplazado por una máquina debería serlo”. La segunda cosa que dijo fue que “Si un niño tiene interés, entonces la educación ocurre”.» *Sugata Mitra* [7]

5. Conclusión

El EEES nos ha llevado a reflexionar sobre cómo innovar en el aula para conseguir que nuestros estudiantes adquieran las competencias apropiadas que les permitan realizarse en el ámbito profesional. En los últimos años la innovación docente ha despertado un enorme interés, pero antes de ese gran despertar muchos docentes habían puesto su mirada en esta parte tan importante del trabajo de un profesor universitario. E incluso antes, mucho antes, in-

telectuales y científicos de renombre reflexionaron sobre cómo debe llevarse a cabo la transferencia del conocimiento entre las viejas y las nuevas generaciones. Posiblemente, como dijo Salomón: “No hay nada nuevo bajo el sol”, pero quedan muchas cosas ocultas aún en la sombra que podemos descubrir.

Referencias

- [1] Bain, Ken: *Lo Que Hacen los Mejores Profesores Universitarios*. Publicaciones de la Universitat de Valencia, 2006.
- [2] Brooks, Frederick P.: *The design of design*. Addison-Wesley, 2010.
- [3] Guzdial, Mark: *Teaching Computing to Everyone*. Communications of the ACM, 52(5), pp 30–33, 2009.
- [4] Krantz, Steven G.: *How to Teach Mathematics. A personal perspective*. American Mathematical Society, 1994.
- [5] Land, Edwin: *The generation of greatness*. Conferencia dada en MIT el 22 de mayo de 1957. Disponible en <http://groups.csail.mit.edu/mac/users/hal/misc/generation-of-greatness.html>
- [6] Miró Nicolau, José: *El caso del heredero reacio IX*. ReVisión, vol. 1, núm. 2, 2008. Disponible en <http://www.aenui.net/ReVision/>
- [7] Mitra, Sugata: *The child-driven education*. Conferencia en TED. Disponible en http://www.ted.com/talks/sugata_mitra_the_child_driven_education.html.
- [8] Norman, Donald.: *Things That Make Us Smart*. Perseus Books, 1993.
- [9] Paterson, David A.: *Your Students Are Your Legacy*. Communications of the ACM, 52(3), pp 31–33, 2009.
- [10] Robinson, Sir Ken: *Out of our minds. Learning to be creative*. Capstones, 2001.
- [11] Stoll, Clifford: *Silicon Snake Oil*. Anchor Books, 995.
- [12] Wolfram, Conrad: *Teaching kids real math with computers*. Conferencia en TED. Disponible en http://www.ted.com/talks/conrad_wolfram_teaching_kids_real_math_with_computers.html
- [13] Young, Art.: *Teaching Writing Across Curriculum*, 3ª Ed. Prentice Hall Resources for Teaching Writing. 2002.

Un sistema de ayuda a la escritura de textos estructurados en LaTeX para estudiantes con déficit visual.

Esther Durá

Instituto de Robótica, Universitat de Valencia
Avda. Catedrático José Beltrán, 2. 46980-PATERNA
Ester.Dura@uv.es

Juan Domingo

Instituto de Robótica, Universitat de Valencia
Juan.Domingo@uv.es

21 de junio de 2011

Resumen

La redacción y presentación de documentos estructurados grandes, especialmente si requieren notación matemática, puede resultar difícil para los alumnos ciegos o con déficit visual usando editores clásicos. El proyecto que se propone, aún en fase inicial, consiste en un sistema de ayuda a la escritura de textos en \LaTeX que reconozca como órdenes simples de voz las principales marcas de este lenguaje y emita también por voz indicaciones útiles como errores, mensajes de estado, etc.

Se discuten las ventajas del uso de lenguajes de marcado para personas ciegas, entre otras la posibilidad de conversión automática a cualquier otro formato (HTML, XML, ...), la posibilidad de inclusión de tablas, de fórmulas (con transformación a MathML o similares), la posible generalización a lenguajes de programación y la facilidad de trabajo colaborativo con otros compañeros, para facilitar la inclusión del alumno con necesidades educativas especiales en el proceso de aprendizaje.

Summary

The writing and presentation of large structured documents, particularly if they require mathematical notation, may be a difficult task for blind or visually impaired alumni using standard editors. The proposed project, still in its initial steps, is aiming to help in the task of writing \LaTeX texts. It should recognize by means of simple voice commands the main marks of this language and should emit, also by voice, useful indications like errors, state messages, etc.

The advantages of using markup languages for visually impaired people are discussed, among others the possibility of automatic conversion to any other format (HTML, XML, ...), the possibility of including tables, equations (with transformation to MathML or similar), the possible generalization to

general purpose programming languages and the ability to do collaborative work with other classmates. In this way the inclusion of the alumni with special educational needs becomes easier.

Palabras clave

Necesidades educativas especiales (NEE), interfaces, déficit visual, lenguajes estructurados, \LaTeX

1. Introducción

La universidad ha sido hasta ahora una institución poco abierta a las personas con discapacidad. Todos los profesionales que trabajamos en ella tenemos la obligación de adaptar nuestros métodos de enseñanza para garantizar la igualdad de oportunidades a todos nuestros estudiantes independientemente de su condición de discapacidad. (Adicional vigésimo cuarta LOU 2002, [3]). Por tanto, debemos dotarnos de los sistemas adecuados para apoyar el proceso de enseñanza/aprendizaje en las mejores condiciones posibles.

Debido a la importancia que las nuevas Tecnologías de la información (TICs) han experimentado recientemente en el proceso de enseñanza/aprendizaje el acceso al medio electrónico es muy importante, especialmente para la inclusión de alumnos con déficit visual. Sin embargo tal integración no siempre es posible debido a una combinación de barreras en el diseño y contenido de los sitios Web y en las aplicaciones de usuario, que trata de paliarse con navegadores adaptados, dispositivos multimedia, lectores de pantalla, reconocedores de voz, etc.

La inclusión de alumnos con déficit visual en el aula requiere entre otras cosas la escritura y la edición de textos en el ordenador. La escritura de fórmulas matemáticas, tablas, inclusión de imágenes, etc. así como corrección de éstos en un editor puede resultar complejo si el programa que se utiliza no

es el apropiado. Igualmente, si el alumno no puede compartir la información escrita con sus compañeros de clase se obstaculiza su plena integración en el aula, especialmente en las tareas grupales.

El objetivo de este trabajo, aún en fase inicial, es desarrollar un herramienta software de ayuda a la escritura de textos en \LaTeX que reconozca como órdenes simples de voz las principales marcas de este lenguaje y emita también por voz indicaciones útiles como errores, mensajes de estado, etc. Actualmente existen pocas herramientas software que provean un entorno amigable y utilicen un lenguaje de edición apropiado, como veremos en la sección 2. Argüiremos en la sección 3 que el lenguaje de marcas \LaTeX resulta especialmente adecuado.

2. Soluciones disponibles

La mayoría de las ayudas a la interacción con el ordenador de personas con déficits visuales están orientadas a la lectura, bien sea por medio de interfaces Braille, bien mediante la síntesis de voz. Existen múltiples soluciones de software libre, como Orca¹ o el proyecto de accesibilidad de KDE² entre otras, centradas cada una en un gestor de ventanas diferente. Un punto de acceso interesante a esta información de modo organizado puede encontrarse en las webs de The Linux Foundation³ o en The Blind Readers Page⁴. Existen también otros sistemas de ayuda, pero la mayoría de ellos comerciales, y disponibles únicamente para Windows, como SuperNova Access Suite⁵ o GWMicro⁶, aunque existe alguno libre y de fuente abierta, como NVDA⁷. Algunos se limitan a ser lectores directos de pantalla (es decir, no tienen en cuenta la estructura del documento, si la hay), mientras que otros la usan para permitir la accesibilidad, como SimplyWeb o el lector web incluido en la distribución LinuxSpeaks⁸.

Las ayudas a la escritura son menores, quizá porque se asume que la interacción a través de un teclado Braille resuelve el problema, sin reparar en el

¹<http://wiki.tiflolinix.org/wiki/Orca>

²<http://accessibility.kde.org/>

³<http://www.linuxfoundation.org/collaborate/workgroups/accessibility>

⁴<http://www.blindreaders.info/>

⁵<http://www.yourdolphins.com/productdetail.asp?id=1>

⁶<http://www.gwmicro.com/Window-Eyes/>

⁷<http://www.nvda-project.org/>

⁸http://www.joekamphaus.net/Web_browser.html

alto grado de realimentación visual que los videntes recibimos al escribir, no tanto respecto a la última porción de texto escrito, sino respecto a la estructura general del documento. Existen poquísimos editores que tengan esto en cuenta, por ejemplo, NoteTab⁹.

Respecto a proyectos de naturaleza académica, es interesante citar el trabajo de Dorbisek et al. sobre un navegador dirigido por voz [1], así como la tesis de máster de O'Donovan [4].

3. Lenguajes de marcas para NEE: déficit visual

Un lenguaje de marcas es una forma de codificar un texto que, junto con el mismo, incorpora etiquetas (marcas) que contienen información acerca de su estructura. La inclusión de tales marcas presenta una serie de ventajas para la escritura, lectura y edición de documentos por alumnos con déficit visual. Entre estas ventajas cabe destacar:

- **Estructura clara:** las etiquetas de marcado en \LaTeX tienen un significado semántico. Esto presenta una gran ventaja ya que permite estructurar de modo semiautomático el documento, dividiéndolo en capítulos, secciones, subsecciones, etc., controlando en todo momento la numeración y las referencias cruzadas, mediante el uso de etiquetas. Estas características hacen que un documento en \LaTeX pueda ser aprehendido fácilmente en su totalidad, lo que resulta muy atractivo para estudiantes con deficiencia visual ya que les permite "imaginar" claramente la estructura completa.
- **Facilidad de construcción de macros y órdenes:** el usuario puede definir órdenes para que éstas se ajusten a sus preferencias personales. Por ejemplo, es posible que una determinada expresión aparezca repetida varias veces en el texto. Nada más fácil que definir una orden que reemplace a todo un bloque. O bien, es posible que no guste la forma en que \LaTeX numera las páginas. Una redefinición al principio del documento permite cambiar esto. La gran ventaja de esto para estudiantes con déficit visual es la posibilidad de crear un lenguaje de marcas nuevo sobre \LaTeX con marcas de longitud menor que

⁹<http://www.notetab.com>

la original, y efectuar su asignación perpetua a órdenes de voz de corta duración.

- **Presentación automática:** \LaTeX tiene una serie de marcas que facilitan el formateado del documento para su presentación con un formato de salida de calidad profesional que lo hace indistinguible del texto escrito por un vidente.
- **Facilidad para incluir:**
 - Ecuaciones matemáticas: éstas, a diferencia de las palabras de una frase normal, no son una sucesión de caracteres sino que usan símbolos cuyo significado depende de su posición relativa en dos dimensiones (arriba, abajo, anexo, etc). \LaTeX permite la inclusión de marcas que definen exactamente la posición de los símbolos en una ecuación; esto proporciona una estructura que puede ser identificada mentalmente por el estudiante no vidente.
 - Tablas: de la misma forma que ocurre con las ecuaciones, las tablas pueden ser representadas en \LaTeX mediante una serie de marcas que definen las dimensiones y posición de cada elemento. Como en el caso anterior, estamos haciendo una transposición de la geometría en una gramática formal.
- **Conversión a formatos diferentes:** Otras de las ventajas que presenta la producción de un documento en \LaTeX es que existe software para su conversión a otros formatos como HTML o XML, o en general, a cualquier otro lenguaje estructurado. También permite la inclusión de paquetes especiales, por ejemplo `braille.sty`, para crear documentos imprimibles en Braille.
- **Trabajos en grupo:** para tener plena integración en una clase y participar en las tareas grupales, muy importantes en los nuevos grados, las personas con déficit visual necesitan poder tener acceso al trabajo de sus compañeros videntes, así como compartir con ellos los documentos escritos. \LaTeX proporciona características que permiten tanto la compartición como la fácil división de un documento entre alumnos videntes y con déficit visual, para su posterior composición.

Estas razones han motivado la elección de \LaTeX como base para la creación de una herramienta soft-

ware que permita la edición, escritura y lectura de documentos estructurados.

4. Desarrollo de un prototipo

Para cumplir los objetivos señalados el sistema de software que se implemente deberá verificar unos requisitos concretos, y fijados éstos deberán escogerse las herramientas y la metodología de desarrollo. Veamos ambas cosas.

4.1. Requisitos

Algunos de los requisitos naturales de un sistema como el propuesto ya se cumplirán de modo natural por el hecho de haber elegido \LaTeX , entre otros la presentación final profesional, la conversión inmediata a otros formatos, etc. Aparte de ellos, se identifican también éstos como necesarios para la consecución de un sistema útil:

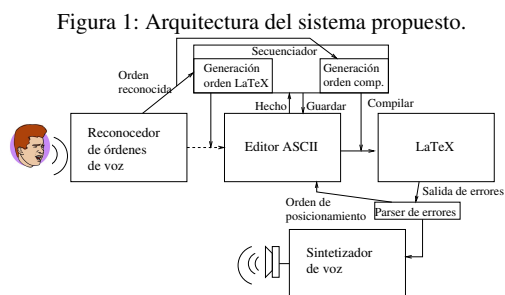
- Debe ser capaz de crear la estructura vacía de un documento complejo y navegar por ella usando tanto órdenes simples de voz como teclas especiales (asumiendo la entrada por un teclado Braille), así como emitir en forma acústica (voz) dicha estructura para su verificación. Esto incluiría la lectura del índice compuesto por \LaTeX hasta ese momento, la lista de secciones y subsecciones, la lista de tablas, etc. y la navegación entre ellas.
- Debe ser de aprendizaje sencillo, aun cuando se asuma que el usuario conozca la estructura y funcionamiento de \LaTeX , esté acostumbrado a la notación matemática y sea capaz de estructurar con orden su propio pensamiento. La interfaz debería contener sólo unas pocas órdenes simples, para minimizar la confusión entre ellas, y responder con rapidez.
- Debe emitir sus mensajes, bien sean errores u otros por medio de voz sintetizada electrónicamente y debe proveer de mecanismos que permitan la corrección de tales errores en respuesta a órdenes de voz. Más aún: debido a la dificultad de esto, debería ser capaz de detectar muchos de los errores en el mismo momento en que se produzcan (llaves no pareadas, entornos no cerrados, etc.)
- Debería realizarse íntegramente con herramien-

tas de software libre, a ser posible de fuente abierta, para evitar costos para el usuario y facilitar la mejora por parte de la comunidad de programadores.

5. Implementación

La arquitectura general del sistema involucra varios bloques, que por facilidad se conectarán entre sí usando las herramientas usuales para comunicación entre procesos de Unix (*pipes*, *fifo*s, etc.) lo que sugiere la implementación patrón en Linux. Esto facilita la interconexión y el cambio futuro de alguna parte del sistema, y no es, pese a lo que pueda parecer, un problema de portabilidad ni a OSX (que es un Unix) ni a Windows, gracias a la existencia de herramientas como Cygwin que proveen una capa POSIX sobre el núcleo de Windows.

Los módulos esenciales serían un reconocedor de voz restringido, entrenado sólo para un número reducido de órdenes; un editor de texto ASCII, para mostrar en pantalla lo hecho con destino a usuarios videntes o con déficit visual, y a grabar el documento; un sintetizador de voz para emitir los mensajes; una distribución de \LaTeX debidamente instalada para compilar el documento, un analizador de errores de compilación y un módulo de conexión entre todos ellos que secuencie las acciones de cada uno. Este último puede contener también el analizador de errores en línea. El esquema general se muestra en la figura 1.



Para el reconocedor de voz la opción más prometedora parece ser la basada en CMU Sphinx¹⁰. Para el editor, la sencillez y la posibilidad de admitir entrada a través de *pipes* sugieren el uso de vi o

¹⁰<http://cmusphinx.sourceforge.net/>

emacs. Respecto al analizador, un programa simple en Perl es suficiente, y en cuanto al sintetizador de voz, la mejor opción parece ser Festival¹¹. El módulo de coordinación sería un programa en C o C++ que gestione hilos, descriptores de archivos y de procesos.

6. Conclusiones

Este trabajo ha presentado las ideas iniciales para la implementación de una herramienta software para la ayuda a la preparación de textos estructurados por estudiantes con déficit visual. La principal conclusión es que el uso de lenguajes estructurados de marcas, y en particular de \LaTeX , representa una ayuda si cabe mayor en el caso de estos estudiantes debido a que refleja la estructura de su intención sobre el documento en el propio documento, y permite la modificación y navegación sencilla por él. La importancia de una herramienta de este tipo para la inclusión de estos alumnos en las actividades de grupo creemos que responde a la filosofía de los nuevos grados bajo el Espacio Europeo de Educación Superior.

Agradecimientos

Este trabajo ha sido parcialmente subvencionado con cargo al proyecto DPI-2008-06691 del Ministerio de Ciencia e Innovación.

Referencias

- [1] Dobrisek, S., Gros, J., Vesnicer, B. et al. *A Voice-Driven Web Browser for Blind People*, Proceedings of the 5th International Conference on Text, Speech and Dialogue, September 2002, ISBN 3-540-44129-8, pp. 453–460.
- [2] Lammport, L., *\LaTeX , a document preparation system*, Addison-Wesley, 1994.
- [3] Ley Orgánica de Universidades, BOE de 24/12/2001, modificada por BOE 13/04/2007.
- [4] O'Donovan, L. *Browser Extension to Improve Web Usability for Blind Computer Users*, MSc thesis, Dept. of Computer Science, University of Cork (Ireland), October 2008.

¹¹<http://www.cstr.ed.ac.uk/projects/festival/>

Aplicación de nuevas técnicas docentes en la asignatura *Sistemas Cliente/Servidor*

E. Jimenez-Ruiz
Computer Science Department
Oxford University
Wolfson Building, Parks Road
OX1 3QD, Oxford, UK
ernesto@comlab.ox.ac.uk

D. Llido, R. Berlanga, O. Rajadell
Departamento de Lenguajes y Sistemas Informáticos
Universitat Jaume I
Campus de Riu Sec
E-12071 Castelló de la Plana, España
dllido,berlanga,orajadel@uji.es

Resumen

En este trabajo mostramos nuestras experiencias en la aplicación de metodologías de aprendizaje cooperativo y basado en proyectos en la asignatura *Sistemas Cliente/Servidor* en los cursos académicos 2008/2009 y 2009/2010.

Summary

In this work we present our teaching experience in the application of cooperative and project-based learning methodologies within the subject *Client/Server Systems* in the academic years 2008/2009 and 2009/2010.

Palabras clave

Aprendizaje cooperativo, aprendizaje basado en proyectos, resultados del aprendizaje, Sistemas Cliente/Servidor, programación Web.

1. Introducción

La asignatura *Sistemas Cliente/Servidor* se imparte durante el segundo cuatrimestre del tercer curso de las titulaciones Ingeniería Informática, Ingeniería Técnica en Informática de Gestión (en modo presencial y semipresencial) e Ingeniería Técnica en Informática de Sistemas. La asignatura es de carácter optativo y tiene 6 créditos (4,8 créditos ECTS), repartidos en 3 teóricos y 3 de laboratorio.

Los profesores intentan dar a la asignatura un enfoque principalmente práctico, aplicando tecnologías cliente/servidor actuales al desarrollo de aplicaciones web, en las que conviven lenguajes para dar formato al contenido de la página web (ej. HTML, CSS) y lenguajes que proporcionan dinamismo y funcionalidad (ej. PHP, JavaScript, AJAX).

Creemos que, en parte por la temática de la asignatura, y en parte por el enfoque práctico que se da a la misma, en los últimos cursos se ha alcanzado un número de alumnos matriculados muy alto (92 alumnos en el curso 2008/2009 y 70 alumnos en el curso 2009/2010) en relación al resto de optativas de las titulaciones de informática.

Durante los cursos anteriores los alumnos han mostrado diferentes niveles de conocimiento de programación en general y de lenguajes concretos como PHP. Al comienzo de la asignatura se pasó una encuesta para evaluar y tener una guía del conocimiento inicial de los estudiantes en los bloques más importantes de la asignatura. Como era de esperar, las frecuencias más altas se centran en los conocimientos *medios-bajos* (véase la figura 1). Dado que la asignatura es una optativa de tercer curso, la gran mayoría de ellos tiene cierta experiencia en creación de páginas web estáticas. Sin embargo, en la asignatura pretendemos proporcionar las bases necesarias para que los alumnos den un aspecto adecuado a la página web así como cierta operabilidad y funcionalidad (ej. gestión de sesiones, restricción del acceso, consultas a base de datos, etc.).

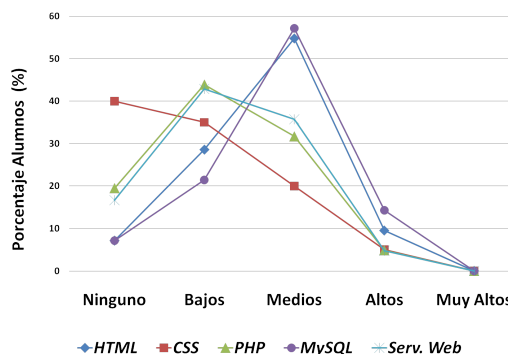


Figura 1: Nivel de los estudiantes al empezar la asignatura

2. Metodología didáctica y evaluación

La asignatura ha seguido de cerca las pautas recogidas en el EEES [7] y ha tenido como principal objetivo el formar futuros profesionales, no sólo en los aspectos más técnicos sino también en competencias como el trabajo en grupo y la cooperación. Para ello proponemos una metodología de aprendizaje cooperativo (AC) [1] y aprendizaje basado en proyectos (ABP) [2, 3] en torno al diseño y elaboración de un portal Web motivado por un caso de uso real. En el curso 2008/2009 el portal web simuló una tienda on-line de material deportivo; mientras que en 2009/2010, el propósito fue simular una web para gestionar la agenda de la policía de barrio¹.

La aplicación de aprendizaje cooperativo también vino motivada por el gran número de alumnos matriculados, y la procedencia de tres titulaciones diferentes, donde las diferencias entre los conocimientos de los alumnos se ha enfatizado. El aprendizaje cooperativo ha ayudado a relajar estas diferencias y ha facilitado el aprendizaje de la mayor parte de alumnos. Además, también ha ayudado a distribuir de forma más eficiente el esfuerzo de los profesores a la hora de atender a los alumnos.

El aprendizaje cooperativo se ha aplicado tanto en la parte práctica como en la teórica de la asignatura:

1. Se motivó la realización del proyecto durante las sesiones prácticas de la asignatura y en grupos de tres o en pareja (también se aceptaron trabajos individuales en casos justificados).
2. Los alumnos crearon de forma colaborativa un *glosario de términos* de teoría en el aula virtual² de la asignatura. La inserción de entradas fue libre, y se dejó a los propios alumnos la corrección y puntuación (grado de confianza) de las definiciones dadas.

2.1. Itinerarios de evaluación

La evaluación de la asignatura considera tanto el aprendizaje cooperativo como el aprendizaje autónomo. Tras la experiencia de años anteriores y teniendo en cuenta las diversas situaciones de los estudiantes, en el curso 2009/2010 se ofrecieron 3 itinerarios de evaluación de la parte práctica:

- **Itinerario A.** Desarrollo del proyecto de forma individual, o en grupos (máximo de 3 alumnos), el cual puntúa un 70% de la nota final. El proyecto tiene unos hitos que alcanzar y se entrega en unas fechas determinadas.
- **Itinerario B.** Examen escrito de prácticas, hasta un 40% de la nota final. Los ejercicios a resolver están basados en los problemas básicos del proyecto.
- **Itinerario C.** Es una alternativa al itinerario A, cuando los alumnos no pueden realizar las entregas en los plazos exigidos. El proyecto completo se entrega en junio y su calificación es hasta un 40% de la nota total.

La parte teórica de la asignatura (aprendizaje autónomo) se supera mediante la realización de dos parciales, o mediante un examen final en junio (hasta un 30% en itinerario A y hasta un 40% en itinerarios B y C). Nótese que el 20% restante en itinerarios B y C se puede obtener mediante ejercicios adicionales, participación en clase y/o colaboración en la creación del glosario.

2.2. Proyecto por equipos

Uno de los primeros hitos de la asignatura es la formación y registro de los grupos. El objetivo de registrar los equipos de trabajo es doble: por un lado fomentar el compromiso de los distintos integrantes del equipo en el trabajo que deciden abordar, y por otro incentivar al alumno a comenzar a desarrollar el trabajo desde el inicio del curso.

Los profesores dan libertad a los alumnos en la formación, ya que la cohesión del grupo está muy ligada a la satisfacción de sus miembros. La única restricción por parte de los profesores es un límite de tres miembros por grupo. Se tomó esta decisión en base a nuestra experiencia en otras asignaturas y también a nuestra experiencia pasada como alumnos. Además, los alumnos no suelen tener suficiente experiencia en el desarrollo de proyectos de forma cooperativa [5] y es aconsejable empezar con grupos pequeños [4]. De esta forma también se incentiva una mayor participación y aumento del diálogo por parte de los integrantes del grupo.

Durante el desarrollo del proyecto se proporcionan a los estudiantes las guías básicas con soluciones parciales en alguno de los problemas a resolver,

¹ Este proyecto se inspira en un proyecto real del Ayuntamiento de Castellón, en el que participaron los autores

² Aula virtual: <https://aulavirtual.uji.es//>

además también se marcan los plazos de entrega (itinerario A) para cada sesión. La autonomía [2] de los alumnos fue en cierta medida limitada en las primeras fases de desarrollo, pero también se proponen ejercicios complejos³ en los que los alumnos son libres de utilizar diferentes soluciones y tecnologías. En el itinerario C la autonomía es algo mayor ya que no existen plazos de entrega intermedios, sin embargo, como se discute en la sección 2.3, el éxito en este itinerario es reducido.

2.3. Resultados de la evaluación

La tabla 1 resume los grupos que se registraron en el itinerario A y los que finalizaron con éxito. En el curso 2008/2009, 70 alumnos de los 92 matriculados se organizaron en 36 grupos, de los cuales 22 llegaron al final superando todos ellos la parte práctica. En el curso 2009/2010 se formaron 32 grupos, comprendiendo 56 de los 72 alumnos matriculados, de los cuales 16 defendieron y aprobaron el proyecto.

La tabla 1 también recoge el éxito de los diferentes tipos de equipo. Los equipos individuales fueron los más sensibles al fracaso, ya que un 80 % en 2008/2009 y un 71 % en 2009/2010 abandonaron el itinerario A. Esto demuestra los beneficios del trabajo cooperativo frente al trabajo individual para el desarrollo del proyecto de la asignatura.

La comparativa entre los resultados de las parejas y los grupos de 3 también confirma nuestras intuiciones presentadas en la sección 2.2. Un grupo mayor no garantiza un mejor resultado, sino todo lo contrario. Las notas medias de las parejas fueron superiores en ambos cursos académicos, incluso en 2009/2010 el índice de abandono/disgregación en los grupos de 3 llegó al 66 %.

Cabe destacar que en el curso académico 2009/2010 se añadió el itinerario C, dando más flexibilidad en la entrega del proyecto. Esto permitió *rescatar* a 5 grupos (1 grupo de 3, 2 parejas y 2 individuales) que no pudieron entregar en los plazos marcados pero continuaron con el desarrollo del proyecto. Originalmente este itinerario pretendía funcionar en paralelo con el itinerario A, pero la mayor parte de grupos que abandonaron el itinerario A o bien dejaron la asignatura o pasaron al itinerario B.

Respecto al itinerario B, en el curso 2008/2009 se

presentaron al examen práctico 13 alumnos, de los cuales 2 aprobó en junio y 2 en septiembre; mientras que en 2009/2010, se presentaron 8 alumnos y aprobaron 1 en junio y 4 en septiembre. En ambos cursos académicos esperábamos un mayor número de estudiantes en cada convocatoria, sin embargo el abandono del proyecto para la mayor parte de los casos significó el abandono de la asignatura.

3. Discusión y conclusiones

Hemos introducido, con un éxito parcial, el aprendizaje cooperativo basado en proyectos como metodología didáctica. Como expone Traver [5, 6], tanto los profesores como los alumnos necesitan más experiencia para poder explotar el potencial de estas metodologías. Para ello, el aprendizaje cooperativo debe convertirse en algo natural de la docencia universitaria poniéndose en práctica en gran parte de las asignaturas.

En los resultados presentados en la Sección 2.3 hemos comprobado un salto significativo en el índice de éxito entre los equipos individuales y las parejas. Sin embargo no se ha obtenido el mismo resultado al incrementar el número de participantes a tres, sino todo lo contrario, ya que la nota media descendió e incluso en el curso 2009/2010 dos grupos abandonaron y otros tantos se disgregaron.

Destacamos dos posibles razones para explicar el mayor éxito de las parejas frente a los grupos de 3:

1. Los estudiantes carecen de la experiencia necesaria para realizar un reparto eficaz de las tareas y para aprender los unos de los otros.
2. Los profesores no definen las tareas del proyecto con suficientes dimensiones y pasos que faciliten el reparto y la cooperación.

Del Canto y otros autores [8] también discuten diversos problemas derivados del trabajo en grupo tales como: dificultades para alcanzar acuerdos, diferentes niveles de ambición entre los miembros del grupo, o miembros que no cumplen con su parte de trabajo. No obstante, enfocan los conflictos como una parte más del proceso de aprendizaje en la que los alumnos deben ser capaces de resolverlos.

En un futuro inmediato, pretendemos dedicar más tiempo y perfeccionar la aplicación de metodologías cooperativas de forma que los estudiantes *aprendan* a aplicarlas de forma satisfactoria [6, 8].

³Enunciado de las prácticas (2009/2010): <http://krono.act.uji.es/people/Ernesto/Practicas2010-SCS.zip>

Cuadro 1: Resumen de los equipos en los curso 2008/2009 y 2009/2010

Tipo de Equipo	Equipos 2008/2009			Equipos 2009/2010		
	Registrados	Finalizan	Nota media	Registrados	Finalizan	Nota media
Individual	10	2	8,1	14	4	8,3
Pareja	18	13	7,7	12	9	9,1
Grupo de 3	8	7	7,2	6	2	8,2

También queremos mantener el proyecto de la asignatura actualizado principalmente respecto a estas dos dimensiones: *necesidades profesionales del momento y restricciones temporales de los alumnos*. Con este objetivo, al finalizar cada curso académico, pasamos una encuesta a los alumnos en la que preguntamos por el grado de satisfacción de la asignatura y los temas que les hubiese gustado tratar⁴. En el curso 2009/2010 destacamos las siguientes sugerencias:

- *Realización de un estimación más precisa de la carga de trabajo*. Para tratar con éxito esta sugerencia, como se ha comentado, será necesaria una participación tanto de los profesores como de los estudiantes. Los profesores deberán realizar un proyecto más ajustado y enseñar a trabajar de forma cooperativa, mientras que los estudiantes deben incrementar sus habilidades para trabajar cooperativamente.
- *Definir un proyecto con tareas más guiadas*. En la línea de la sugerencia anterior, el proyecto tendrá como objetivo fomentar el aprendizaje cooperativo, pero para ello también será necesario dejar cierta autonomía a los grupos para que discutan y resuelvan los problemas.
- *Mayor organización de los contenidos teóricos y una correspondencia más clara con las tareas de la parte práctica*. La asignatura debe fomentar el aprendizaje cooperativo pero sin olvidar los conocimientos técnicos, impartidos en las lecciones teóricas, que se deben adquirir.
- *Introducir tareas relacionadas con la seguridad web*. En la medida de lo posible la asignatura y el correspondiente proyecto irán introduciendo nuevos temas y problemas para resolver.

⁴Cabe destacar que esta pregunta también se formuló al inicio del curso, pero muy pocos alumnos propusieron nuevos temas.

Agradecimientos

Los autores agradecen la financiación obtenida mediante los proyectos de mejora e innovación educativa de la Universitat Jaume I, aplicados en la asignatura Sistemas Cliente/Servidor: *‘Soporte a las necesidades de programación web’* y *‘Aplicando aprendizaje cooperativo basado en proyectos’*.

Referencias

- [1] Ovejero, A., *El aprendizaje cooperativo. una alternativa a la enseñanza tradicional*, 336 páginas, Editorial PPU España, 1991.
- [2] ITESM, *Método de proyectos*, Sistema Tecnológico de Monterrey, Mexico, 2002
- [3] Solomon, G., “Project-Based Learning: a Primer”. *Technology and Learning*, vol. 23, 2003
- [4] Rico, I. y Fontecha, C., *El aprendizaje cooperativo en educación física*. Instituto Nacional de Educación Física. Universidad Politécnica de Madrid, 2008.
- [5] Traver, V.J. y Traver J.A., “¿Por qué no enseñamos a aprender cooperativamente?”. *Actas de las X Jornadas de Enseñanza Universitaria de Informática (JENUI)*, pp. 297-304, 2004.
- [6] Traver, V.J. y Traver J.A., “Obstáculos al aprendizaje cooperativo universitario: una mirada a los estudios de informática y a la Universidad Jaume I”. *Actas del I Simposio Nacional de Docencia en la Informática*, pp. 53-60, 2005.
- [7] Suárez, B., “Las titulaciones técnicas y el Espacio Europeo de Educación Superior (EEES)”. *Jornada sobre convergencia en el EEES*, 2002.
- [8] del Canto, P., Gallego, I., López J.M., Mora, J., Reyes A., Rodríguez, E., Sanjeevan, K., Santamaría, E. y Valero, M., “Conflictos en el trabajo en grupo: Dos casos representativos”. *Actas de las XV Jornadas de Enseñanza Universitaria de Informática (JENUI)*, pp. 379-386, 2009.

Utilización de tinta digital para favorecer la implantación de una asignatura del grado en Ingeniería Informática

Jorge Más-Estellés
Dpto. Física Aplicada
Universidad Politécnica de Valencia
Camino de Vera s/n
46022 Valencia
jmas@fis.upv.es

José Vicente Benlloch-Dualde
Dpto. Informática de Sistemas y Computadores
Universidad Politécnica de Valencia
Camino de Vera s/n
46022 Valencia
jbenlloc@disca.upv.es

Resumen

La experiencia ha consistido en la utilización de Tablets PC en una clase de ejercicios de teoría de circuitos, y en el uso de tinta digital para la revisión, tanto de ejercicios propuestos a los alumnos, como de memorias de prácticas de laboratorio.

La experiencia ha sido llevada a cabo en un grupo de alto rendimiento y docencia en inglés, en el primer año de impartición del Grado en Ingeniería Informática.

El trabajo se centra en la descripción de cómo se ha desarrollado la experiencia, y en las ventajas y problemas encontrados.

Summary

Experience has involved using Tablet PC on a problem-solving class about electric circuits, and using digital ink to review proposed exercises and laboratory reports.

Experience has been carried out on a high performance group and English teaching, on the first year of a Computer Science degree.

This paper is focused on how the experience has been developed, and on lessons learnt, found advantages and disadvantages.

Palabras clave

Tinta digital, grado, Tablet PC, corrección.

1. Motivación

En la actualidad, numerosas escuelas de Ingeniería españolas se encuentran inmersas en la puesta en marcha de los primeros cursos de las titulaciones de grado. Uno de los pilares de este proceso es potenciar la actividad del alumno dentro y fuera del aula y, relacionado con la consecución de los

logros de aprendizaje, incorporar un seguimiento continuo de dicha actividad. Esta situación está suponiendo una sobrecarga evidente para los docentes que se enfrentan a la revisión de todo el material generado por la actividad de sus alumnos que, en no pocas ocasiones, se distribuyen en grupos numerosos.

Desde hace tiempo, numerosos autores señalan el potencial de los ordenadores personales para facilitar este tipo de enfoques [9]. Además, en disciplinas científicas o tecnológicas, tan habituales en titulaciones de Informática, las posibilidades educativas de los equipos se incrementan cuando incorporamos como elemento de la interfaz de usuario, un lápiz especial que produce tinta digital [7]. De entre los distintos dispositivos pertenecientes a las conocidas como *pen-based technologies*, destaca por sus prestaciones el Tablet PC, que se podría definir como un ordenador portátil que utiliza ese lápiz para introducir libremente información sobre una pantalla digitalizadora, de forma similar a como lo haríamos sobre un papel.

En un estudio reciente [11], fueron revisados 144 trabajos relacionados con el uso de Tablet PC en entornos educativos. En el mismo concluyeron que, en el ámbito universitario, alrededor del 45% de las experiencias estaban relacionadas con estudios de Informática. A pesar de que numerosos trabajos señalan un impacto positivo de estas tecnologías en el aprendizaje [10, 4], hasta el momento se han descrito pocas experiencias de uso de las tecnologías de tinta digital en nuestro país [5]. La mayoría de las referencias apuntan a instituciones norteamericanas o australianas.

Este trabajo explora las posibilidades de las tecnologías de tinta digital en la implantación de una asignatura de primer curso del nuevo grado de Ingeniería Informática en la Escuela Técnica

Superior de Ingeniería Informática, de la Universidad Politécnica de Valencia.

2. Desarrollo de la experiencia

La experiencia llevada a cabo se ha dividido en dos acciones bajo el común denominador del uso de la tinta digital. En ambos casos se ha desarrollado en un grupo de características singulares, ya que era un grupo del grado en Ingeniería Informática calificado como de alto rendimiento académico (grupo ARA), y además con docencia en inglés.

En la primera de ellas, se ha utilizado una red de Tablets PC, de la que ya disponía la Escuela gracias a un proyecto financiado por Hewlett Packard [6] en la convocatoria del año 2008. En particular, se ha utilizado para desarrollar una clase de resolución de circuitos eléctricos dentro de la asignatura Fundamentos Físicos de la Informática, con el fin de dinamizar la participación del estudiante y, al mismo tiempo, facilitar una realimentación inmediata por parte del profesor.

La segunda acción ha consistido en la utilización de tinta digital para la revisión y corrección de ejercicios y memorias de prácticas de laboratorio hechos por los alumnos en la misma asignatura. La recogida y devolución de estos ejercicios o memorias se ha efectuado mediante una herramienta desarrollada por nuestra universidad.

2.1. Utilización de Tablets PC

Nuestra Escuela dispone de un aula dotada con una red inalámbrica de 25 Tablet PC. El profesor puede disponer de un equipo análogo, o bien de un ordenador de sobremesa, en ambos casos conectado a un proyector que permite la visualización del material de la clase a los alumnos. Debido al reducido tamaño del grupo, cada Tablet PC era utilizado por uno o, a lo sumo dos alumnos. La comunicación entre los equipos a través de la red inalámbrica lo hemos hecho mediante la aplicación libre Classroom Presenter [2]. Mediante esta aplicación, el profesor puede proyectar una presentación al conjunto de la clase y, a su vez, cada Tablet recibe una instancia de dicha presentación. Sobre esa presentación de partida, el profesor puede realizar anotaciones, que serán difundidas a los equipos de los alumnos.

Por su parte, los alumnos pueden realizar sus anotaciones particulares en la copia que han recibido. Además, el profesor puede recibir contribuciones de sus alumnos, pudiendo decidir cuáles muestra al resto de alumnos, y cuáles no. El profesor puede visualizar las contribuciones de sus alumnos, pero sin saber quién se la manda, a no ser que las firmaran. Esta característica proporciona a los alumnos un anonimato que les hace desinhibirse y sentirse más libres a la hora de participar con sus soluciones. Al acceder a la aplicación, existen diferentes roles de profesor y alumno, con permisos diferentes.

En la experiencia llevada a cabo, el profesor propuso a los alumnos la resolución de un circuito eléctrico de corriente continua, mediante dos procedimientos diferentes: aplicando las leyes de Kirchoff y aplicando el principio de superposición; además, debían calcular el generador equivalente de Thèvenin entre dos puntos del circuito.

A partir del enunciado del ejercicio, cada alumno/grupo de alumnos, escribe las ecuaciones correspondientes en la pantalla de su Tablet, y las envía al profesor. Si este lo estima conveniente, puede proyectar la solución de un grupo en la pantalla general, tanto para mostrar la solución correcta, como para mostrar errores comunes. De esta manera, se va construyendo la solución del ejercicio con aportaciones de todos los grupos de alumnos, a la vez que el profesor podría revisar, aunque fuera una vez terminada la sesión, las distintas contribuciones, y así hacerse una idea de los logros reales de aprendizaje. Es muy importante que el profesor marque y controle muy bien los tiempos, ya que en caso contrario, puede dilatarse mucho y se le dedique más tiempo del previsto. Si bien la aplicación es sencilla, también es recomendable que el profesor haya practicado previamente con el sistema y lo maneje con soltura, ya que de lo contrario, se vería afectada la dinámica de la clase.

2.2. Corrección de ejercicios y memorias

La segunda acción en la que se ha utilizado tinta digital, ha sido en la revisión y corrección de ejercicios propuestos a los alumnos y memorias del trabajo práctico en el Laboratorio. Tanto unos como otras eran entregados por los alumnos en formato digital, sin necesidad de imprimirlos en papel. Ello lo podían hacer mediante una

plataforma específica desarrollada por nuestra universidad (*PoliformaT*), basada en Sakai, [12] y que, entre otras cosas, permite tanto la propuesta de tareas y ejercicios a los alumnos, como la recepción de estos trabajos y su posterior devolución después de corregirlos. El único inconveniente detectado en esta plataforma, es que no permite la corrección on line del ejercicio; es necesario guardar el ejercicio localmente, corregirlo, y después reenviarlo al alumno. Además, esta plataforma permite almacenar los ejercicios enviados y sus calificaciones. El ejercicio propuesto estaba en formato MS Word, y se le daba al alumno la posibilidad de devolverlo en el mismo formato o bien en pdf, ya que la herramienta de corrección de la que disponemos, se puede utilizar con ambos formatos.

Para llevar a cabo la corrección, es posible la utilización de un Tablet PC, ya que en el software que incorporan “de serie”, se incluyen herramientas para este menester. En particular, todos los equipos que incorporan MS Office, incluyen una opción de “Revisión” en la barra de herramientas, que permite hacer anotaciones manuales sobre el documento. El problema que hemos detectado, sobre todo aquellos profesores de mayor edad, es que la reducida dimensión de la pantalla de los Tablet dificulta su utilización en la corrección. Por ello, hemos preferido llevar a cabo las correcciones en equipos de sobremesa, con pantallas de mayor tamaño, conectando a estos equipos una tableta digitalizadora de bajo coste [3], pero que necesita la instalación de un software [8] que incorpora la opción de revisión de documentos. Es muy importante que el software a instalar sea totalmente compatible con la versión de MS Word utilizada pues, en caso contrario, pueden surgir innumerables problemas, como de hecho nos ocurrió. Este software permite la selección de lápices de distintos grosores y colores, así como goma digital para borrar anotaciones.

Para la corrección de los documentos en formato pdf, utilizamos la misma tableta digitalizadora con el software Adobe Acrobat 9.0 Pro [1], que directamente incorpora en sus menús la opción de revisión de documentos. En la figura se representa una página de uno de los ejercicios revisados en formato MS Word.

1.- A straight and infinite carrying-current wire is flowed by a constant current I_0 . At a distance a from wire there is a squared loop with side a and resistance R . On a time $(t=0)$, lower side of the loop started moving to up with constant speed v_0 . On a time t , compute:

a) Magnetic flux is passing through the loop.
 b) Induced electromotive force on loop.
 c) Induced current is flowing along the loop. Give its sense.
 d) Mutual inductance between wire and loop.
 e) Force acting on moving side of the loop.

Solution:

a) Magnetic field on a point inside surface of the loop is $B = \frac{\mu_0 I_0}{2\pi x}$ *B depends on x*

On a time t after side of the loop started moving, magnetic flux through the loop is:
 $\phi = \int_a^{a+vt} \frac{\mu_0 I_0}{2\pi x} dx = \frac{\mu_0 I_0}{2\pi} \ln \frac{a+vt}{a} = \frac{\mu_0 I_0}{2\pi} \ln(1 + \frac{vt}{a})$ *2a*
 $\phi = \frac{\mu_0 I_0}{2\pi} \ln(1 + \frac{vt}{a}) \cdot a^2 = \frac{\mu_0 I_0 a^2}{2\pi} \ln(1 + \frac{vt}{a})$ *Wb*

b) $\mathcal{E} = -\frac{d\phi}{dt} = -\frac{\mu_0 I_0 a^2}{2\pi} \cdot \frac{1}{1 + \frac{vt}{a}} \cdot \frac{v}{a} = -\frac{\mu_0 I_0 a v}{2\pi(1 + \frac{vt}{a})}$ *you must review theory*

c) $i = \frac{\mathcal{E}}{R} = \dots$

d) Mutual inductance depends only on geometry, being it value $M = \frac{\phi}{I_0} = \frac{\mu_0 a^2}{2\pi} \ln(1 + \frac{vt}{a})$

e) On moving side of loop, magnetic field is uniform, and as induced current is zero, $F = iB = 0$

Figura 1. Corrección sobre un ejercicio de un alumno en formato MS Word.

3. Discusión y conclusiones

La valoración subjetiva, tanto de los profesores como de los alumnos, es que ambas acciones influyen positivamente en el proceso de aprendizaje. En el caso de la utilización de tinta digital, es evidente que simplifica y acelera el proceso de revisión de documentos, evitando la utilización de papel y haciendo posible el envío inmediato de documentos; en la utilización de los Tablet, aunque la experiencia podría haber sido evaluada de una forma cuantitativa, la falta de un grupo de control de características semejantes al utilizado en la experiencia nos lo desaconsejó..

Como ventajas de la utilización de Tablets PC podemos citar:

- Elimina la utilización de papel por parte de los alumnos en la resolución del ejercicio.
- Permite que el ejercicio sea resuelto por el conjunto de alumnos y que el profesor reciba todas las soluciones, en formato electrónico.
- El anonimato del alumno le hace intentar la resolución del ejercicio de forma desinhibida.
- Permite mostrar los errores más comunes de manera instantánea.

- La novedad de la metodología supone una motivación extra para el alumno.

Como desventajas:

- Necesita de la instalación y mantenimiento de una infraestructura compleja y cara, si se utilizan Tablet PC. Si se opta por el uso de tabletas, resulta más asequible.
- Necesita de un adiestramiento previo por parte del profesor y de una cuidadosa preparación de material y planificación de la clase.

En cuanto a las ventajas de la utilización de tinta digital para la revisión de documentos:

- El intercambio de documentos en formato digital permite la utilización de plataformas de comunicación e intercambio, que además posibilitan almacenamiento de documentos y calificaciones.
- Se elimina completamente la utilización tanto de papel como de impresoras y tinta, con el consiguiente ahorro medioambiental y de tiempo.
- El alumno puede recibir su documento corregido de forma inmediata tras la corrección.
- Permite la utilización de archivos en diferentes formatos, algunos de libre uso.

Y como desventajas:

- Es necesario disponer, por parte del corrector, de una tableta digitalizadora (de bajo coste) y de software de no libre distribución.

Agradecimientos

Este trabajo ha sido financiado por la Universidad Politécnica de Valencia, en el marco del Proyecto de Innovación y Mejora Educativa “Uso de tecnologías de tinta digital como facilitadoras de las metodologías activas y la evaluación formativa” (A015/10).

A la E.T.S. de Ingeniería Informática de la UPV por permitirnos el uso del aula con Tablets y por su financiación, y a los técnicos informáticos por el mantenimiento del aula y equipos.

Referencias

- [1] Adobe Acrobat 9 Pro. <http://www.adobe.com>
- [2] Anderson, R. et al. *Classroom Presenter: Enhancing Interactive Education with Digital Ink*. Computer, 40(9), 56-61, 2007.
- [3] *Bamboo A6*. <http://www.wacomonline.com>
- [4] Bilén, S.G., et al. *Tablet PC Use and Impact on Learning in Technology and Engineering Classrooms: A Preliminary Study*, Workshop on the Impact of Pen-Based Technology on Education”, 11-19, 2008.
- [5] Bonastre, O.M. et al. *Pedagogical Use of Tablet PC for Active and Collaborative Learning*, International Professional Communication Conference, IEEE , 214-218, 23-25, 2006.
- [6] *Higher Education HP Technology for Teaching Grant Initiative Recipients (2004-2008)*. http://www.hp.com/hpinfo/grants/us/programs/tech_teaching/hied_global_emea.html
- [7] Jeffrey J. McConnell. *Active learning and its use in computer science*. Proceedings of ITiCSE '96, 52-54, 1996. <http://doi.acm.org/10.1145/237466.237526>
- [8] *JustWriteOffice*. <http://www.wacomonline.com>
- [9] Kinnaman, D. E., & Dyrli, O. E. *Teaching Effectively with Technology: What Every Teacher Needs to Know About Technology*. Technology & Learning Magazine, 1995.
- [10] Koile, K. and Singer, D. *Improving learning in CSI via tablet-PC-based in-class assessment*. Proceedings of (ICER '06), 119-126, 2006. <http://doi.acm.org/10.1145/1151588.1151607>
- [11] Mckenzie, W. & Franke, K. 2009. *Active, Constructive, Interactive: How are Tablet PCs Transforming the Learning Experience in Higher Education?*. 2009 Australasian Tablets in Education Conference, Monash University, Melbourne, 3-4, 2009. <http://www.monash.edu/eeducation/assets/documents/atiec/2009atiec-wendymckenzie.pdf>
- [12] Sakai. *A community of educators collaborating to create open source software that advances teaching, learning and research* <http://sakaiproject.org/>

Plan de acogida para los alumnos de nuevo ingreso en el grado de Ingeniería Informática en la ESII (UCLM)

Diego Cazorla* Pedro Cuenca* Mere Macià** J. Pascual Molina* J. Miguel Puerta*

*Dep. Sist. Informáticos ** Dep. Matemáticas

Universidad de Castilla-La Mancha

Escuela Superior de Ingeniería Informática

Campus Universitario s/n. 02006 Albacete.

{Diego.Cazorla, Pedro.Cuenca, Hermenegilda.Macia, JosePascual.Molina, Jose.Puerta}@uclm.es

Resumen

Con el propósito de facilitar la adaptación de los nuevos estudiantes a la Universidad, se ha diseñado en la Escuela Superior de Ingeniería Informática de Albacete (ESII) de la Universidad de Castilla-La Mancha (UCLM) un plan de acogida. Este plan consiste en un conjunto de actividades concebidas para asentar las bases del aprendizaje en la Educación Superior, en especial en los estudios de Informática.

Durante la primera semana del curso se realizan unas jornadas de acogida, con charlas de presentación de todos los servicios que ofrecen tanto el centro como la Universidad. También se imparten diversos talleres y seminarios. Además de estas jornadas, durante todo el curso se ofrecen actividades de apoyo y refuerzo, así como un plan de tutorías personalizadas que les acompañará durante sus estudios universitarios en este centro.

Summary

In order to facilitate the adaptation of new students to the University, a welcome plan is designed by our institution. This plan consists of a set of activities designed to lay the foundations for learning in Higher Education, specifically in Computer Science. During the first week of the first semester, introductory talks on the services of the School and the University are given, and several workshops and seminars are held too. In addition to these sessions, and throughout the two semesters, some support and reinforcement activities are also offered to students, and a personalized tutoring plan that will accompany them during their studies at this center.

Palabras clave

Evaluación y calidad de la docencia. Organización curricular y planes de estudios. Acogida de alumnos de nuevo ingreso.

1. Introducción

Algunas de las dificultades por las que atraviesa un estudiante al incorporarse al grado de Ingeniería Informática es tener que enfrentarse, entre otras situaciones, a nuevas materias, nuevos enfoques, nuevos métodos docentes y nuevos compañeros. De esta forma, generalmente es necesario cambiar su visión de la profesión de estudiante universitario, para que así puedan superar todos estos retos con garantías.

Nuestro primer objetivo, es hacerles ver qué se espera de un estudiante universitario [1]. El segundo, es convencerles de que lo pueden conseguir. El tercero, es proporcionarles las herramientas necesarias para su consecución. Para ello, se les guía desde el primer día con una serie de actividades, recogidas en este plan de acogida de la ESII, donde se les ofrece, principalmente, información, técnicas, medios y apoyo.

Existen varias iniciativas similares en la Universidad española, [3] y [5] podrían valer de ejemplo. En nuestra propuesta, se recoge el testigo de acción tutorial planteado en esos trabajos, adaptándolo a nuestras peculiaridades, y haciendo especial hincapié en las actividades dirigidas a los alumnos de nuevo ingreso. De esta forma, el plan presentado en este artículo se basa y planifica en tres pilares: las jornadas de acogida (durante la primera semana del curso), actividades de apoyo (principalmente en primer y segundo curso) y las tutorías personalizadas (durante su etapa como estudiante de la ESII).

2. Jornadas de acogida

Los primeros días de un estudiante en la Universidad son vitales para asentar las bases de su aprendizaje en la Educación Superior, en lo que va a ser su ocupación y preocupación en sus próximos años. En estos primeros días, además de familiarizarlos con

su nuevo entorno físico, aunque algunos estudiantes ya conocían la ESII, bien a través de su página web [9], bien a través de las actividades de promoción de la Escuela ofertadas a través de los centros de enseñanza secundaria, o bien por sus visitas en eventos concretos como son la *Semana de la ciencia* o las *Jornadas de puertas abiertas*, es importante que se fijan las bases de su aprendizaje. Así, se les indican los conocimientos, habilidades, destrezas y valores que se busca que vayan adquiriendo en su vida como estudiantes para después pasar a ser los profesionales que la sociedad necesita. Este es el mejor momento para que tomen conciencia de la profesión que acaban de iniciar de estudiante universitario del grado en Ingeniería Informática. Profesión, que a muchos les parecerá dura, y que seguro requiere de grandes sacrificios y esfuerzos. También se les muestra el plan de trabajo, cuáles son sus obligaciones y derechos y se les tiende la mano para llevar con éxito esta tarea.

Las diversas actividades propuestas en las Jornadas de acogida están basadas sobre la guía elaborada por la UIE (Unidad de Innovación Educativa) de la UCLM [8]. En primer lugar, se pretende dar a conocer a quienes dirigen el centro en estos momentos, a los profesores de este primer cuatrimestre y a los compañeros. Se les informa sobre diferentes normativas, especialmente, la de permanencia de la UCLM [10]. Y, por supuesto, que también conozcan el campus, dónde están y cómo son los servicios básicos que se ofrecen: biblioteca, despachos de profesores, instalaciones administrativas y deportivas, cafetería, etc. En segundo lugar, se ofrece una serie de talleres y seminarios destinados a despertar su interés por su propio proceso de aprendizaje del que ellos son los principales responsables.

Durante esta primera intensa semana, además de las actividades de acogida, se imparten también 2h de clases regladas de las asignaturas correspondientes, donde van conociendo a los profesores de este primer cuatrimestre y empiezan a tomar contacto con las materias que les van a ocupar en este primer curso. A continuación pasamos a exponer brevemente las actividades de esta primera semana, con una breve descripción de la temática de cada una.

- **Charlas en las jornadas de acogida (2-3h)**
 - *Charla de bienvenida del director del centro:* Bienvenida, presentación de la ESII y

de los estudios de grado de Ingeniería Informática (plan de estudios, metodología docente, tutorías personalizadas, normativa de permanencia y acreditación de idioma nivel B1 de la UCLM), salidas profesionales, etc.

- *Charla de la unidad de extensión universitaria y servicio de deporte.*
- *Charla de la delegación de alumnos de la ESII.*

- **Taller de presentación del servicio de biblioteca y consultas básicas en el catálogo (2h)**

Su finalidad es que adquieran y desarrollen las habilidades y destrezas básicas que necesitan para buscar y encontrar la información más adecuada, de manera presencial o telemática.

- **Seminario de Introducción a los Recursos Informáticos de la UCLM (2h)**

Su objetivo es mostrar algunos recursos de las TIC's que ofrece la Universidad, así como guías prácticas de uso y configuración. También se introducen otros servicios informáticos no institucionales, pero que les pueden ser de gran utilidad.

- **Taller de orientación al estudiante (9h)**

El fin de esta actividad es favorecer la adaptación a las nuevas metodologías de enseñanza y aprendizaje basadas en el esfuerzo y autonomía del estudiante. En particular se abordan las siguientes temáticas:

1. Introducción al Aprendizaje Cooperativo [4] (3h)
2. Aprendiendo a aprender: Estrategias y técnicas para lograr eficacia y satisfacción en las tareas de estudios. Introducción a la inteligencia emocional [2] (3h)
3. La organización del tiempo como factor de éxito académico (1,5h)
4. Orientaciones prácticas para la consecución de las competencias para la correcta comunicación oral y escrita [6] (1,5h)

3. Actividades de apoyo

Además de las jornadas de acogida, se ofrecen también actividades de apoyo, esta vez con carácter optativo. Algunas de estas ofertas están destinadas a homogenizar el nivel en algunas asignaturas tradicionalmente *más complicadas* para los nuevos alumnos, pretendiendo con ello obtener un mayor éxito en la superación de las mismas. Para ello se proponen diferentes modalidades, según sus necesidades.

- **Seminarios: cursos cero de Cálculo y Álgebra** (20h/seminario)

A una gran cantidad de alumnos de nuevo ingreso en nuestro centro les resultan difíciles las Matemáticas y, sobre todo, les cuesta adaptarse a la nueva forma de ver esta materia. En particular, a los provenientes de ciclos formativos y a los que no han cursado esta materia en el segundo curso de Bachillerato, les resultan especialmente difíciles. Es por ello se proponen estos dos seminarios de cursos cero. Tienen lugar en el mes de septiembre, previo al inicio del curso y están dirigidos a aquellos alumnos que hayan hecho la pre-inscripción en junio/julio.

- **Seminarios: Refuerzo de Cálculo, Álgebra y Física** (20h/seminario)

Para aquellos alumnos que no han podido beneficiarse de los cursos cero de Cálculo y Álgebra, se les ofrece la oportunidad de realizar los seminarios de refuerzo de dichas materias. Además, también se oferta otro seminario de refuerzo de Física. Cada uno de ellos se imparte simultánea y coordinadamente con la asignatura que refuerza (Cálculo y Física en el primer cuatrimestre, Álgebra en el segundo).

Por otro lado, desde la ESII se quiere fomentar el uso del software libre, y con este fin se propone un taller que intenta cambiar la visión que algunos de nuestros alumnos de nuevo ingreso tienen de él.

- **Taller de introducción a GNU/Linux-MOLINUX** (20h)

Esta distribución Linux es una iniciativa de la Junta de Comunidades de Castilla-La Mancha por potenciar el software libre, apostando por un sistema operativo seguro, fácil de usar y que permite reducir los costes del software.

Debido a que las nuevas metodologías docentes se centran en el estudiante, es necesario una constante interacción que permita una evaluación continua con garantías. Nuestro plan de estudios está diseñado con 5 asignaturas de 6 créditos ECTS por cuatrimestre [9], de este modo, y durante los dos primeros cursos y para aquellos alumnos que no hayan superado durante el cuatrimestre anterior las asignaturas propias de ese cuatrimestre, desde el centro se oferta la posibilidad de volver a cursarlas en el siguiente cuatrimestre, y así poder optar a una evaluación continua. De esta manera se facilitan, de forma natural, el cumplimiento de las normas de permanencia en la UCLM [10].

4. Plan de tutorías personalizadas

Además del asesoramiento y la ayuda recibida en los primeros cursos a través de las jornadas de acogida y las actividades de apoyo, creemos necesario un trato más personal, que permita dar respuesta al alumno a sus preguntas, dudas o inquietudes, orientándolo tanto en el momento de su ingreso como a lo largo de su permanencia en el centro. Para ello, se oferta el plan de tutorías personalizadas, que consiste en la asignación de un profesor, que hará el papel de tutor académico, a cada alumno de nuevo ingreso, permitiendo así una atención personalizada a lo largo de su vida como estudiante en el centro.

Este plan tiene por objetivos, además del de orientar y ayudar a los alumnos que así lo soliciten, intentar favorecer la integración y la participación de los estudiantes en el centro, dando especial apoyo a alumnos con necesidades especiales (de origen extracomunitario, discapacitado, etc.), mediar en los eventuales conflictos y canalizar a través del tutor las posibles dificultades con las que se encuentra el alumno para poder paliar las deficiencias desde el centro.

Además del tutor personal, el alumno también dispone de un tutor para las prácticas en empresas o para los programas de movilidad, si así se requiere.

5. Resultados

A continuación exponemos, a modo de ejemplo, los resultados obtenidos en el taller de orientación al estudiante.

- Alumnos asistentes: Variable, dependiendo de los días y los horarios (mañana-tarde).
- Máxima asistencia: 50 alumnos.
- Asistencia el día de la evaluación: 23 alumnos.
- Alumnos que entregan cuestionario: 22 alumnos.

Respecto a la valoración general, alrededor del 95% de los estudiantes que han respondido al cuestionario de evaluación consideraron que los contenidos del taller de orientación al estudiante le podían ayudar bastante o mucho a ser más eficaces en sus estudios.

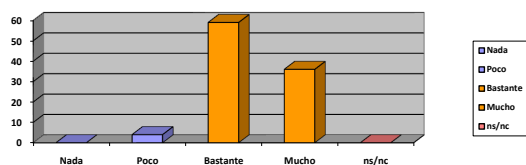


Figura 1: Resultados de la valoración general

Estos resultados, en cuanto a satisfacción general, se mantienen en los distintos talleres/seminarios. Aunque, desafortunadamente, no se ven reflejados en grandes resultados académicos. Por ejemplo, de los 30 alumnos que realizaron el curso cero de Cálculo o el seminario de refuerzo de Cálculo, el 50% superaron la asignatura de Cálculo y Métodos Numéricos en la convocatoria ordinaria (el nivel de aprobados está sobre el 30%).

6. Conclusiones

En este artículo se expone el plan de acogida de los alumnos de nuevo ingreso en la ESII. El grado de satisfacción de los estudiantes que han participado es elevado, así como de toda la comunidad que ha colaborado en el mismo. El punto más débil a destacar es el número de participantes: de 110 alumnos de nuevo ingreso en primer curso, en el mejor de los casos, han asistido 50 alumnos a las diversas sesiones de las actividades. Eso nos lleva a plantearnos un mejor marketing del mismo para cursos venideros y un mejor horario. También, como trabajo futuro, tenemos la intención de incorporar nuevos seminarios de refuerzo para las asignaturas de programación de primer curso, así como incorporar un sistema de tutorías basado en la mentoría entre iguales [7].

Agradecimientos

Este trabajo ha sido posible gracias al Proyecto de Innovación Educativa financiado por UIE de la UCLM.

Referencias

- [1] Alonso, J.R. *Una Universidad nueva*, El País, 12 Enero 2009.
http://www.elpais.com/articulo/educacion/Universidad/nueva/elpepuedu/20090112/elpepuedu_3/Tes
- [2] Goleman, D. *Inteligencia Emocional*, Ed. Kairós, 1996.
- [3] Gómez, A.; González, J. y Ortiz, C. *EMPATÍA: Implantación de un plan de acción tutorial para enseñanzas técnicas*. XIV Jornadas de Enseñanza Universitaria de la Informática, 2008.
- [4] Johnson D.; Johnson R. y Smith, K. *Cooperative learning: Increasing College Faculty Instructional Productivity*. ASHE-ERIC Higher Education Report No: 4, George Washington University, 1991.
- [5] León, C. *Actividades de orientación y tutoría en el grado de Informática*. XVI Jornadas de Enseñanza Universitaria de la Informática, 2010.
- [6] Poblete, M. y Villa, A. *Aprendizaje Basado en Competencias. Una propuesta para la evaluación de competencias genéricas*. Bilbao. Universidad de Deusto. Mensajero, 2007.
- [7] Sánchez, C.; Almendra, A.; Jiménez, F.J.; Melcón, M.J. y Macías, J. *Proyecto Mentor en la Universidad Politécnica de Madrid: un sistema de mentoría para la acogida y orientación de alumnos de nuevo ingreso*. Revista Iberoamericana de Sistemas, Cibernética e Informática, Vol. 6, N.2, 2009.
- [8] Unidad de Innovación Educativa de la UCLM.
http://www.uclm.es/organos/vic_ordenacionacademica/uie/
- [9] web Escuela Superior de Ingeniería Informática.
<http://www.esiiab.uclm.es>
- [10] web UCLM. Normativa de permanencia.
<http://www.uclm.es/normativa/pdf/estudiantes/alumnos11.pdf>

Uso de redes sociales para mejorar el rendimiento de los alumnos con diferentes estilos de aprendizaje

Sonia Vázquez, Irene Garrigós, Jose-Norberto Mazón, Estela Saquete y Rubén Izquierdo

Departamento de Lenguajes y Sistemas Informáticos
Universidad de Alicante

{svazquez,igarrigos,jnmazon,stela,ruben}@dlsi.ua.es

Resumen

Debido a los cambios que el Espacio Europeo de Educación Superior introduce al potenciar las horas de trabajo no presencial, se hacen necesarios nuevos mecanismos para posibilitar una mejor comunicación y cooperación en el proceso de aprendizaje. Las redes sociales, como *Facebook*, pueden suministrar estos mecanismos, pero su uso satisfactorio para la docencia puede verse afectado en gran medida por el estilo de aprendizaje de los alumnos. Este artículo plantea la necesidad de estudiar la influencia de los diferentes estilos de aprendizaje en la docencia no presencial mediante el uso de redes sociales con el fin de incrementar el rendimiento de los alumnos. Cabe destacar que este artículo describe el proyecto “Las redes sociales y su relación con los estilos de aprendizaje” a realizar dentro del programa de Redes de Investigación en Docencia Universitaria del Instituto de Ciencias de la Educación de la Universidad de Alicante.

Summary

Due to the changes introduced in the European Higher Education Area for increasing the long-distance learning, it is required new mechanisms for better communication and cooperation. Social networks, such as *Facebook* can provide these mechanisms, but their successful can be affected by the learning style of the students. This paper poses how learning styles can influence the performance of the students when social networks are used in teaching. It is worth noting that this paper describe the project “Las redes sociales y su relación con los estilos de aprendizaje” to be developed within the program “Redes de Investigación en Docencia Universitaria” of the “Instituto de Ciencias de la Educación” at the University of Alicante.

Palabras clave

redes sociales, estilos de aprendizaje, aprendizaje no presencial, aprendizaje colaborativo, EEES..

1. Motivación

El Espacio Europeo de Educación Superior (EEES) crea un nuevo modelo educativo basado en el Sistema Europeo de Transferencia y Acumulación de Créditos (ECTS) con el fin de comparar la evolución y el rendimiento de los estudiantes universitarios de diferentes países de la Unión Europea. Una característica fundamental del modelo EEES es que se asigna a cada trabajo académico, independientemente de su naturaleza presencial o no presencial, su correspondiente tiempo en créditos.

Por lo tanto, es necesario el uso de nuevos mecanismos para facilitar el desarrollo de trabajos académicos de índole no presencial como, por ejemplo, la plataforma Moodle¹ que da soporte a los profesores para la creación de comunidades educativas a distancia. Sin embargo, el uso de este tipo de plataformas (llamadas LMS - Learning Management System) puede dificultar la formación de redes de aprendizaje donde la comunicación y cooperación sean fluidas, debido a que no son herramientas que incluyan elementos creativos o lúdicos [3,4]. Contrariamente a este hecho, los estudiantes son propensos a una comunicación fluida a través de herramientas colaborativas orientadas a consolidar y ampliar sus relaciones sociales, como *Facebook*² [5]. Por

¹ <http://moodle.org>

² <http://www.facebook.com>

consiguiente, se considera que *Facebook* es un buen ejemplo de entorno orientado al ocio que puede usarse con éxito para mejorar la participación activa de los estudiantes en el proceso de aprendizaje [6].

Llegados a este punto, conviene destacar que se debe tener en cuenta el estilo de aprendizaje del alumno para diseñar el proceso de aprendizaje más adecuado. Además, en un entorno colaborativo es preciso determinar cómo influye el estilo de aprendizaje de cada alumno individualmente en el rendimiento global del grupo. Por tanto, la hipótesis de partida de este trabajo es que el uso de una red social como *Facebook* por parte de un grupo de estudiantes con diferentes estilos de aprendizaje puede hacer que el rendimiento global mejore. Con la finalidad de corroborar esta hipótesis se han propuesto una serie de experimentos que se están llevando actualmente a cabo bajo un programa de Redes de Investigación en Docencia Universitaria del Instituto de Ciencias de la Educación de la Universidad de Alicante.

En estos experimentos, se parte del hecho de que cada alumno posee un estilo de aprendizaje determinado, con unas características concretas que debe ser analizado individualmente. Este análisis lo realizaremos mediante el test de Felder [1,2]. Los resultados que se pretenden obtener con los experimentos nos permitirán analizar cómo influye el estilo de aprendizaje de los alumnos, de manera individual, y si esta influencia es más acusada en algún estilo de aprendizaje concreto. Además, también se podrá observar si la colaboración entre alumnos con distintos estilos de aprendizaje, cuando se usan redes sociales como *Facebook* para fines docentes, hace que el rendimiento global aumente. Este experimento se llevará a cabo realizando diversas actividades a lo largo del período docente de la asignatura Diseño y Programación Avanzada de Aplicaciones de Ingeniería en Informática en la Universidad de Alicante.

Este artículo está estructurado del siguiente modo. En la siguiente sección veremos algunos trabajos relacionados. En la sección 3 presentamos la metodología y la experimentación que desarrollaremos. Finalmente, en la sección 4 mostramos las conclusiones.

2. Trabajo relacionado

Existen varios trabajos que resaltan la importancia de determinar el estilo de aprendizaje de los estudiantes con el fin de mejorar su rendimiento al usar aplicaciones Web educativas, ya sea a nivel individual o para trabajo colaborativo. Tradicionalmente los estilos de aprendizaje se han tenido en cuenta para el desarrollo de aplicaciones Web educativas de manera individual [7,8]. Sin embargo, resulta de suma importancia poder utilizar el conocimiento derivado del estilo de aprendizaje para mejorar el trabajo colaborativo de los estudiantes tal y como se expone en [9]. En concreto, nuestro punto de vista es que el uso de las redes sociales como herramienta de aprendizaje colaborativo para el uso docente puede mejorar el rendimiento académico de las clases no presenciales siempre y cuando se tenga en cuenta el estilo de aprendizaje de los estudiantes.

No obstante, los trabajos que consideran estilos de aprendizaje en entornos colaborativos de las redes sociales se centran en analizar y descubrir el estilo de aprendizaje a partir del uso que se les da a dichas redes. Por ejemplo, en [11] se trata de determinar un conjunto de reglas que se apliquen a determinados atributos que describan la interacción del usuario en *Facebook* para predecir con cierta precisión su estilo de aprendizaje, sin necesidad de hacer ningún estudio previo. Otro trabajo relevante [10] hace un estudio del impacto de los estilos de aprendizaje en la formación de grupos para el desarrollo de un mejor aprendizaje colaborativo. En este trabajo se demuestra con una serie de experimentos que el rendimiento del estudiante en el trabajo colaborativo se ve afectado por los estilos de aprendizaje, resultando beneficiosa la interacción de estudiantes de diversos estilos para una mejor experiencia educativa.

En otro tipo de trabajos, por ejemplo en [12] se implementa una red social propia especialmente diseñada para tareas docentes y se detecta el tipo de aprendizaje del estudiante mediante la aplicación de técnicas de inteligencia artificial con la finalidad de adaptar la red social a las necesidades de los estudiantes según su estilo de aprendizaje.

Por lo tanto, los trabajos existentes no tratan de mejorar el rendimiento de los estudiantes usando redes sociales. Considerando la necesidad

de mejorar el rendimiento de los alumnos, sobre todo en el aprendizaje no presencial, nuestro trabajo se centra concretamente en el uso de una red social como *Facebook* para tratar de mejorar su rendimiento académico.

3. Mejora del rendimiento según el estilo de aprendizaje mediante el uso de redes sociales

Con la finalidad de validar si el uso de redes sociales como *Facebook*, en este caso, ayuda y refuerza el aprendizaje no presencial y mejora el rendimiento de los alumnos, se propone la realización del siguiente estudio. En primer lugar, en una primera fase de toma de contacto, se ha de determinar el estilo de aprendizaje de los alumnos. Esta fase previa es necesaria para recabar toda la información posible sobre la forma en que los alumnos procesan y retienen el conocimiento. Posteriormente, en la siguiente fase se les planteará cuatro tareas, dos de ellas de manera individual, y otras dos tareas similares a las primeras pero utilizando como plataforma de aprendizaje colaborativo *Facebook*. Estas cuatro tareas irán ligadas a temas relacionados con la asignatura que los alumnos cursan de forma presencial.

A continuación se detallan estas dos fases del estudio propuesto en este artículo.

3.1. Recopilación de información

La primera fase requiere la realización de un test por parte de los alumnos para determinar el estilo de aprendizaje de cada uno de ellos. Mediante los resultados obtenidos se podrá obtener una visión más precisa de la forma en que los alumnos aprenden y retienen la información, y de esta manera, intentar potenciar aquello que más les beneficia en el proceso de aprendizaje.

Para el desarrollo de esta fase, se ha adaptado el test de Felder [1,2], que consta de 44 preguntas y que clasifica a los estudiantes en 5 dimensiones diferentes:

- *Sensitivos-intuitivos*: En esta dimensión los estudiantes perciben mejor dos tipos de información: información externa o sensitiva a la vista, al oído o a las sensaciones físicas e información interna o intuitiva a través de memorias, ideas, lecturas, etc.
- *Visuales-verbales*: En esta dimensión, la información externa se canaliza mejor a

partir de formatos visuales como cuadros, diagramas, gráficos, demostraciones, etc. o a partir de formatos verbales mediante sonidos, expresión oral y escrita, fórmulas, símbolos, etc.

- *Inductivos-deductivos*: Los estudiantes clasificados en esta dimensión se sienten a gusto y entienden mejor la información si está organizada inductivamente, dando los hechos y las observaciones e infiriendo los principios o deductivamente, donde los principios se revelan y las consecuencias y aplicaciones se deducen.
- *Secuenciales-globales*: El progreso en el aprendizaje de los estudiantes clasificados en esta dimensión implica un procedimiento secuencial que necesita progresión lógica de pasos incrementales pequeños o un entendimiento global que requiere de una visión integral.
- *Activos-reflexivos*: La información se puede procesar mediante tareas activas a través de compromisos en actividades físicas o discusiones o a través de la reflexión o introspección.

La adaptación del test ha consistido en la traducción al español de las 44 preguntas y su posterior publicación en un servidor propio, al que los alumnos pueden acceder identificándose con su DNI³.

3.2. Metodología

Dado que el uso de *Facebook* como plataforma de aprendizaje colaborativo potencia ciertas dimensiones respecto a otras, centraremos nuestro estudio en la dimensión visual-verbal.

En primer lugar, la parte “visual” se refiere a la utilización de elementos de aprendizaje tales como vídeos, diagramas, esquemas, etc. En nuestro caso, se ha optado por la realización de dos tareas (individual sin el uso de *Facebook* y colaborativa a través de *Facebook*), en las que se utilizarán vídeos demostrativos acerca de ciertos conceptos teóricos. En segundo lugar, la parte “verbal” requiere de la utilización de elementos tales como textos, fórmulas o símbolos. Para potenciar el aprendizaje de esta dimensión se realizará otras dos tareas donde se utilizarán

³ <http://gplsi.dlsi.ua.es/testFelder>

básicamente textos explicativos sobre diferentes conceptos teóricos.

Cada una de estas tareas se realizará en el aula de manera individual y también se utilizará la potencia de *Facebook* como red social colaborativa, mediante el intercambio de información a través del muro en forma de comentarios y discusiones.

Finalmente para completar el estudio y evaluar el resultado de cada tarea, se realizarán tests a cada uno de los alumnos para cada una de las tareas (tanto las individuales como las realizadas de manera colaborativa) para determinar qué elemento de aprendizaje les ha proporcionado mejores resultados. De esta manera tendremos una evaluación de los conocimientos adquiridos de manera individual asociados a su estilo de aprendizaje y también si estos resultados mejoran con el uso *Facebook* en tareas similares, a partir de la colaboración de compañeros con diferentes estilos de aprendizaje. Además, mediante los resultados obtenidos, podremos valorar si realmente el uso de *Facebook* como plataforma de aprendizaje colaborativo, ha potenciado las diferentes dimensiones del estilo de aprendizaje que tenía cada alumno inicialmente.

4. Conclusiones

En este artículo se plantean una serie de experimentos que nos permitirán conocer si el uso de redes sociales (en concreto *Facebook*) permite que el rendimiento de los alumnos mejore cuando realizan trabajos colaborativos según sus diferentes estilos de aprendizaje. Los experimentos planteados en este trabajo se desarrollarán dentro del programa de Redes de Investigación en Docencia Universitaria del Instituto de Ciencias de la Educación de la Universidad de Alicante.

Referencias

- [1] Felder, R. M., y Soloman, V. (1984): Index of Learning Styles. <http://www.ncsu.edu/felder-public/ILSpage.html> [Consulta: mayo 2011].
- [2] Felder, R. M. y Silverman, L. K. (1988): "Learning and Teaching Styles in Engineering Education Application". *Engr. Education*, vol. 78 (7), pp. 674-681.
- [3] DeSchryver, M., Mishra, P., Koehler, M., & Francis, A. (2009). Moodle vs. Facebook: Does using Facebook for discussions in an online course enhance perceived social presence and student interaction? *Society for Information Technology and Teacher Education International Conference 2009* (pp. 329-336).
- [4] Ezeiza, A. (2009). Facebook como apoyo a la docencia presencial: ¿son nuestros alumnos "amigos"? <http://www.ainhoaezeiza.net>
- [5] Barkhuus L., Tashiro J.: Student Socialization in the Age of Facebook. *Conference on Human Factors in Computing Systems (CHI) 2010* (pp. 133-142).
- [6] Garrigós, I., Mazón, J.N., Saquete, E., Puchol, M., Moreda, P. La influencia de las redes sociales en el aprendizaje colaborativo. *JENUI 2010*.
- [7] Wolf, C.: iWeaver: towards an interactive web-based adaptive learning environment to address individual learning styles. *Euro. J. Open Distance E-learn. (EURODL 2002)*.
- [8] Brown, E.J., Brailsford, T.: Integration of learning style theory in an adaptive educational hypermedia (AEH) system. *ALT-C Conference 2004*.
- [9] Sánchez Hórreo, V., Carro, R.M.: Studying the Impact of Personality and Group Formation on Learner Performance. *CRIWG 2007: 287-294*
- [10] Alfonseca, E., Carro, R.M., Paredes, M., Ortigosa, A., Martín, E.: The impact of learning styles on student grouping for collaborative learning: a case study. *User Modeling and User-Adapted Interaction. Communities and Collaboration*, vol. 16(3-4), pp. 377-401. Springer (2006)
- [11] Quiroga, J.I. Modelado de usuario y adaptabilidad en redes sociales. http://www.eps.uam.es/esp/alumnos/trabajos_fin_master/Quiroga_Filgueira_Jose_Ignacio.pdf
- [12] Zatarain-Cabada, R., Barrón-Estrada, M.L., Ponce Angulo, V., García, A.J., Reyes García, C.A.: Identification of Felder-Silverman Learning Styles with a Supervised Neural Network. *ICIC (2) 2010: 479-486*

Práctica de Laboratorio sobre implementación Joystick HID-USB de interfaz con una emisora RC

Rafael Paz Vicente, Elena Cerezuela Escudero, Ángel Jiménez Fernández, Manuel Domínguez Morales, Gabriel Jiménez

José Ignacio Villar de Ossorno

Dpto. Arquitectura y Tecnología de Computadores
Universidad de Sevilla

Dpto. Tecnología Electrónica

E.T.S. Ingeniería Informática. Avda. Reina Mercedes s/n
41012 Sevilla
rpaz@atc.us.es

Resumen

Actualmente, la interconexión de cualquier tipo de periférico con un ordenador se realiza utilizando el bus USB. Dentro de este tipo de dispositivos, podemos resaltar un tipo especial denominado HID (Human Interface Device) destinada a la conexión de periféricos de interfaz humana, como pueden ser teclados, ratones, joysticks.

Este tipo de dispositivos se caracterizan por ser reconocidos por el sistema operativo sin necesidad de utilizar ningún driver. En este artículo se describe la implementación de un dispositivo de este tipo por parte de los alumnos.

Al no necesitar el desarrollo de un driver simplifica el número de elementos que el alumno deberá desarrollar para poder completar satisfactoriamente la práctica.

La siguiente práctica permite al alumno profundizar en los mecanismos por el cual dichos dispositivos se identifican y enumeran sus características y posibilidades, resultando atractivo al poder programar un microcontrolador que se comporte como un joystick.

Summary

Currently, the interconnection of all types of peripherals to a computer is performed using the USB bus. Within these devices, we highlight a special type called HID (Human Interface Device) used for connecting peripherals human interface such as keyboards, mice, joysticks. Such devices are characterized by being recognized by the operating system without using any driver. This article describes the implementation of a device of this kind from the students.

By not requiring the development of a driver simplifies the number of items that the student must develop in order to successfully complete the practice.

This lab session allows the student to gain insight into the mechanisms by which these devices are identified and listed its features and capabilities, resulting in attractiveness the ability to program a microcontroller to behave as a joystick.

Palabras clave

Aprendizaje basado en problemas, trabajo en equipo, aprendizaje colaborativo, redes de sensores inalámbricos, programación orientada a objetos.

1. Motivación

La adaptación del Espacio Europeo de Educación Superior y, en particular, la adopción del sistema de crédito europeo (ECTS) están determinando un cambio de modelo educativo. Evolucionamos desde un modelo centrado en la “enseñanza del profesor” a uno basado fundamentalmente en el “aprendizaje del alumno” [1]. Esto ha determinado que prácticamente todas las universidades hayan promovido planes de innovación docentes que faciliten la implantación de nuevas metodologías por parte del profesorado..

En la Universidad de Sevilla, en el curso 2007, se puso en marcha el Plan de Renovación de las Metodologías Docentes 2007 [2]. Este plan estaba dividido en varias líneas promoviendo, entre otras cuestiones, la formación del profesorado en nuevas las metodologías docentes,

implantación de dichas metodologías en las asignaturas, dotación de espacios docentes, etc.

Con la implantación de los nuevos grados, los profesores están ultimando los contenidos, metodologías y planificaciones de las nuevas asignaturas. En muchos casos nos hemos encontrado con la novedad de la carga docente no presencial por parte del alumno, esto nos ha supuesto plantearnos nuevas cuestiones acerca de cómo implementar dicha carga. Estas cuestiones giran en torno a cómo elaborar actividades y tareas que ocupen esta carga por parte del alumno, así como su seguimiento y evaluación por parte del profesor. En definitiva, las universidades han invertido una elevada cantidad de recursos en formar al profesorado en nuevas metodologías didácticas y de aprendizaje, sin embargo, poco se invertido en la formación en nuevos contenidos.

Esta deficiencia se agudiza en el ámbito de la ingeniería informática, y en particular en el ámbito de la arquitectura de computadores, sistemas empujados, dispositivos móviles, etc., ya que esta es una tecnología en continuo cambio y evolución, y además resulta crucial para el desarrollo tecnológico que los ingenieros informáticos posean conocimientos de la tecnología más actual posible, estando esta responsabilidad casi exclusivamente sobre el profesor.

En el ámbito de las ingenierías, las cuales son ciencias aplicadas, opinamos que el uso de las metodologías del Aprendizaje Basado en Proyectos (ABP) resulta muy adecuado para su uso en las partes no presenciales de las asignaturas. Está tratado ampliamente en numerosos trabajos [3,4,5] y a pesar de que es una metodología relativamente nueva, ha demostrado ser eficaz en la enseñanza de las más diversas ramas de conocimiento [5]. Sin embargo, puede que no todo sea ventajoso, el uso del ABP en la carga no presencial de las asignaturas demanda un elevado nivel de motivación por parte del alumno, ya que es el propio alumno el que tiene que "obligarse" a participar en el ABP.

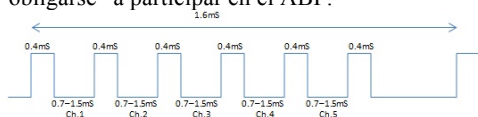


Figura 1: Fotografía trama PPM

2. Material proporcionado para la elaboración de la práctica

2.1. Placa Silabs C8051F342 con USB

Para desarrollar la práctica de la asignatura, que ocupará un total de dos sesiones prácticas, se proporcionará a los alumnos una placa de circuito impreso basada en el microcontrolador de Silabs C8051F342. Dicho microcontrolador se basa en el popular 8051 de Intel, uno de los microcontroladores más significativos de 8 bits (junto al Motorola 68HC11 y posteriormente los microcontroladores PIC de Microchip). El núcleo de dicho procesador se halla licenciado en la actualidad por un gran número de fabricantes de microcontroladores, y existen en el mercado una amplia oferta de utilidades de desarrollo, compiladores, etc.

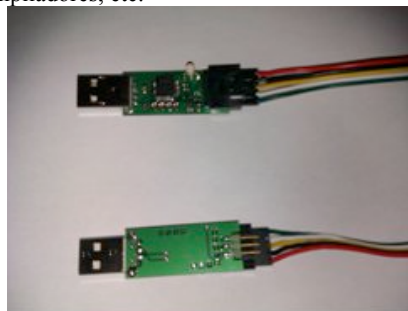


Figura 2: Fotografía de placa USB.

La placa que se proporciona a los alumnos se basa fundamentalmente en el microcontrolador antes mencionado. Dicho microcontrolador integra un controlador USB, que permitirá su interconexión con el ordenador. Adicionalmente, cuenta con un divisor de tensión que permitirá obtener el nivel de tensión de una batería externa (por ejemplo, la de una emisora RC), que permitirá estudiar el muestreo de señales analógicas utilizando el ADC que incorpora el microcontrolador. Mediante el empleo del Timer del microcontrolador, se pretenderá emitir patrones intermitentes de luz utilizando un led bicolor (Rojo y Verde, o amarillo mediante la mezcla de los dos otros dos colores).

Dos conectores situados en el otro extremo del circuito, permitirán la interconexión a un dispositivo externo. En nuestro caso, la señal PPM proveniente de la emisora RC.

2.2. Compilador de C para 8051

Tradicionalmente los microcontroladores se han programado siempre utilizando el lenguaje ensamblador específico de cada microcontrolador debido a las restricciones de memoria y a la baja velocidad de los microcontroladores, que hacían necesario que el desarrollador realizaría la programación del microcontrolador a bajo nivel.

En la actualidad, el aumento de las capacidades y de las prestaciones de los microcontroladores ha permitido el empleo de lenguajes de alto nivel en su programación. Al utilizar un lenguaje de alto nivel como el lenguaje C, se ha simplificado significativamente la tarea de programación y depuración de los programas que se desarrollan.

El alumno no deberá aprender un nuevo lenguaje de programación al abordar la programación de una nueva familia de microcontroladores. Así mismo, la utilización de un lenguaje de alto nivel permite programar utilizando un mayor nivel de abstracción, obviándose muchos detalles específicos de cada familia.

En nuestro caso, se podrá utilizar varios compiladores de C disponibles para microcontroladores 8051, como son el Keil y el SDCC (gratis GPL).

2.3. Programa de ejemplo para el manejo del bus USB desde el microcontrolador.

Para facilitar el comienzo del desarrollo de la práctica por parte del alumno, y a fin de reducir el tiempo que se pierde inevitablemente al comenzar a trabajar con un nuevo entorno de desarrollo, hasta que uno se familiariza con el entorno y sus peculiaridades, se le ofrece al alumno un proyecto de ejemplo completamente funcional que podrán compilar y descargar en la placa de desarrollo para comprobar fácil y rápidamente que la instalación del entorno se ha hecho de manera satisfactoria, y de que se comprende adecuadamente la mecánica de trabajo en el nuevo entorno.

2.4. Software y documentación para creación de descriptores HID.

El interés de desarrollar un dispositivo HID es doble. Por un lado se enseñará al alumno el funcionamiento del bus USB, y los mecanismos

utilizados por el controlador USB y por el sistema operativo para enumerar los diferentes dispositivos USB que se pueden conectar a los distintos puertos USB, directamente o mediante el uso de concentradores o HUBs.

Por otro lado, dentro de los dispositivos USB, los dispositivos HID se caracterizan por no necesitar el desarrollo de un driver para el sistema operativo que se utilice. En el proceso de enumeración del dispositivo, un dispositivo HID se identifica como tal, y le describe sus características al sistema operativo mediante unas tablas o descriptores. Dentro de los dispositivos HID posibles, como son teclados, ratones, trackballs, joysticks, tarjetas gráficas, etc. Nos vamos a centrar en el Joystick, que puede resultar atractivo a los alumnos, al permitirles a partir de esta práctica, realizar distintas modificaciones para fabricar sus propios joysticks o gamepads.

Se le proporcionará al alumno la documentación necesaria para la creación de los descriptores de dispositivo.

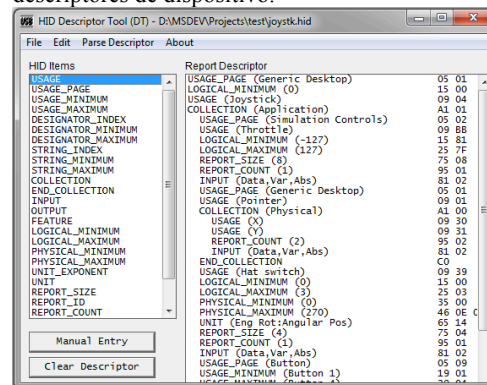


Figura 3: Fotografía aplicación descriptores HID

2.5. Documentación de la trama PPM

Para realizar la captura de la señal PPM de la emisora RC, se repasará a los alumnos el uso del PCA ("Programable Counter Array") del microcontrolador para la captura de señales digitales, permitiendo medir con precisión la anchura de cada uno de los pulsos enviados desde la emisora.

Se les dará a los alumnos una introducción a la señal PPM, sus características, y cómo se codifica en un único canal para ser transmitido por

radio el valor de los diferentes sticks e interruptores de la emisora.

3. Metodología de elaboración y seguimiento

La asignatura se divide en varias prácticas, donde el alumno se familiarizará progresivamente con el entorno de desarrollo. Para esta práctica, estructurada en dos sesiones diferentes, se supone que el alumno se ha familiarizado en prácticas anteriores con el entorno de desarrollo, así como con la documentación del microcontrolador proporcionada por su fabricante, permitiendo que esta práctica se centre en el desarrollo de un dispositivo USB, y en la profundización del manejo de otros componentes del microcontrolador, como son los timers, ADC, etc.

La primera de las sesiones se centrará en explicar las nociones necesarias sobre el USB y el HID, y la estructura y funcionamiento del código de ejemplo facilitado, así como las modificaciones que deberán realizar. Los alumnos tendrán el resto de la sesión de práctica, para realizar estas modificaciones. Para que puedan completar dicha tarea antes de la siguiente sesión de prácticas, se les cederá a modo de préstamo el material que utilizan en la práctica para que puedan completar los ejercicios en casa, permitiéndoseles también el acceso a un laboratorio.

En la segunda sesión, se trabajará con el PCA para capturar la señal PPM proveniente de la emisora. Con ayuda del ADC se podrá monitorizar el nivel de batería de la emisora. Utilizando el timer, se emitirán distintos patrones de luz para indicar distintas condiciones (no hay señal PPM, batería baja, etc).

4. Conclusiones

En este trabajo se ha presentado el enfoque adoptado en dicha asignatura para realizar una aproximación amena y atractiva al alumno a la programación de microcontroladores. Se le enseña a manejar de forma ágil la documentación técnica proporcionada por el fabricante, así como abordar de manera progresiva la programación de un microcontrolador.

La programación del controlador USB suele ser una tarea laboriosa debido a que hay que manejar de forma correcta los diferentes mensajes

y descriptores intercambiados entre el ordenador personal y el microcontrolador. Gran parte de esta tarea se le facilita al alumno al aportarle un código de ejemplo básico que deberán modificar adecuadamente para alcanzar los objetivos planteados en la práctica.

Se les intenta motivar al pedirles que implementen un dispositivo Joystick, para que el objetivo les resulte atractivo. Aportándoles una ayuda adicional en los aspectos más arduos y complicados, garantizando así un máximo aprovechamiento de las horas de trabajo. Evitando así que se desanimen al encontrar dificultades que les puedan provocar la pérdida de mucho tiempo, sin sentir que van avanzando en el desarrollo de la práctica.

Agradecimientos

Este proyecto ha sido posible en parte gracias al proyecto VULCANO (TEC2009-10639-C04-02).

Referencias

- [1] M. Valero-García, J. Navarro, FAQ sobre la docencia centrada en el aprendizaje. <http://epsc.upc.edu/~miguel%20valero/contingut3.html>.
- [2] Universidad de Sevilla. Plan de Renovación de Metodologías Docentes 2007. http://www.institucional.us.es/plan_renovacion.
- [3] T. Markham Project Based Learning, a guide to Standard-focused Project based learning for middle and high school teachers. *Buck Institute for Education, (2003)*.
- [4] G. Solomon. Project Based Learning: a Primer, *Technology a Learning*, vol. 23, num. 6 (Jan 2003).
- [5] H. Dolmans, et al. Problem-based learning: future challenges for educational practice and research. Blackwell Publishing Ltd 2005. *Medical Education (2005)*; 39.
- [6] M. Hedley, S. Barrie. An undergraduate microcontroller systems laboratory., *IEEE Transactions on Education*, Volume: 41, Issue: 4, (Nov.1998).
- [7] M. Valero-García. Las dificultades que tienes cuando haces PBL. Cap. 8 La Educación Superior hacia la Convergencia Europea.
- [8] <http://www.usb.org/developers/hidpage>

Sistema de co-diseño hardware/software basado en FPGA para la captura de video analógico a través del bus serie USB

Angel Jimenez Fernandez, Rafael Paz Vicente, Manuel Dominguez Morales, Elena Cerezuela Escudero, Miguel Angel Rodriguez Jodar

Dpto. Arquitectura y Tecnología de Computadores
Universidad de Sevilla

E.T.S. Ingeniería Informática. Avda. Reina Mercedes s/n
41012 Sevilla
ajimenez@atc.us.es

José Ignacio Villar de Ossorno

Dpto. Tecnología Electrónica

Resumen

Con la adaptación del Espacio Europeo de Educación Superior (EEES) y la creación de los nuevos títulos de grado, se plantea una nueva visión metodológica en la enseñanza superior.

En el caso de titulaciones relativas a las ingenierías informáticas y electrónicas resulta muy interesante el uso de metodologías orientadas al Aprendizaje Basado en Proyectos (ABP).

En esta exposición se presenta un proyecto en el que se ilustra a los alumnos como diseñar sistemas hardware / software. En particular proponemos el codiseño de un sistema capaz de capturar video analógico mediante el uso de un convertidor analógico/digital (ADC) y una FPGA, y además envíe la información capturada a través del bus USB usando un microcontrolador.

De esta manera esperamos que los participantes de la experiencia ABP adquieran conocimientos relativos a las metodologías para codiseño hardware/software, uso de FPGA, microcontroladores y ADC. Así como entrenar habilidades relativas a la descripción de circuitos lógicos mediante el lenguaje VHDL.

Summary

By adapting the European Higher Education Area (EEES) and the creation of new graduate degrees, there is proposed a new methodological view in higher education. In the case of studies related to computer and electronic engineering result very interesting to use the methodologies for Project Based Learning (PBL). In this paper, we present an exposition that shows students how to design systems hardware / software. In particular we propose the co-design a system capable of capturing analog video using an analog / digital converter (ADC) and a FPGA, and

also send the information captured through the USB bus using a microcontroller. In this way we hope that the participants of the PBL experience acquired knowledge of the co-design methodologies for hardware / software, use of FPGA, microcontroller and ADC, and also training skills related to the description of logic circuits using VHDL.

Palabras clave

Aprendizaje basado en problemas, trabajo en equipo, aprendizaje colaborativo, programación de microcontroladores, lenguaje C, ARM.

1. Motivación

La adaptación del Espacio Europeo de Educación Superior y, en particular, la implantación de las nuevas titulaciones de grado, el modelo educativo está experimentando un gran cambio. Evolucionamos desde un modelo centrado en la "enseñanza del profesor" a uno basado fundamentalmente en el "aprendizaje del alumno" [1].

Debido a esta evolución, la mayoría de las universidades han promovido planes de innovación docente que faciliten la implantación de nuevas metodologías por parte del profesorado.

En la Universidad de Sevilla, en el curso 2007, se puso en marcha el Plan de Renovación de las Metodologías Docentes 2007 [2]. Este plan estaba dividido en varias líneas promoviendo, entre otras cuestiones, la formación del profesorado en nuevas las metodologías docentes, implantación de dichas metodologías en las asignaturas, dotación de espacios docentes, etc.

Con la implantación de los nuevos grados, en muchos casos nos hemos encontrado con la novedad de la carga docente no presencial por

parte del alumno. Esto nos ha supuesto el planteamiento de actividades y tareas que ocupen esta carga por parte del alumno, así como la elaboración de mecanismos para su seguimiento y evaluación por parte del profesor. Las universidades han invertido una elevada cantidad de recursos en formar al profesorado en nuevas metodologías didácticas y de aprendizaje, sin embargo, poco se ha invertido en la formación en nuevos contenidos. Siendo este un aspecto crucial en los estudios de base tecnológica, ya que la tecnología está en continuo cambio y evolución, resultando crucial para el desarrollo tecnológico que los futuros profesionales posean conocimientos de la tecnología más actual posible, estando esta responsabilidad casi exclusivamente delegada al profesor.

En particular, en el ámbito de las ingenierías, el uso de las metodologías del Aprendizaje Basado en Proyectos (ABP) resulta muy adecuado para su uso en las partes no presenciales de las asignaturas. Está tratado ampliamente en numerosos trabajos [3,4,5] y ha demostrado ser eficaz en la enseñanza de las más diversas ramas de conocimiento [5]. Sin embargo, puede que no todo sea ventajoso. El uso del ABP en la carga no presencial de las asignaturas demanda un elevado nivel de motivación por parte del alumno, ya que es el propio alumno el que tiene que “obligarse” a participar en el ABP [7]. Para tratar de conseguir esa motivación extra, pensamos que se puede conseguir proponiendo proyectos que les sean muy atractivos.

2. El problema del material y los laboratorios

Gracias al I Plan Propio de Docencia de la Universidad de Sevilla en el curso 2008-2009 (línea 13: infraestructuras y equipamiento de espacios docentes) se adaptó un laboratorio para el desarrollo hardware para su libre uso por parte del alumnado, creando así un espacio docente adaptado a las necesidades de la implantación de metodologías basadas en ABP en materias que versan acerca del desarrollo de sistemas hardware.

Este laboratorio se equipó, entre otras medidas de seguridad, con un control de acceso controlado por la tarjeta identificativa de los estudiantes, de manera que los alumnos pertenecientes a las asignaturas participantes de experiencias ABP

están autorizados a su uso, pudiendo sólo acceder los alumnos adecuados. Además este sistema nos permite hacer un seguimiento acerca del uso individual de cada alumno del laboratorio. A diferencia de las prácticas simuladas y eminentemente software, en las que todo el material necesario es un ordenador, para el desarrollo de prácticas hardware suele ser necesario el uso de un material mucho más específico y costoso, como son fuentes de alimentación, multímetros, osciloscopios, generadores de señales, analizadores lógicos o de espectro. El alumno puede encontrar disponible todo este equipamiento y puede hacer un uso libre del mismo, acercando y familiarizando al alumno al equipamiento de laboratorio necesario para el desarrollo hardware que se encontrará en su vida laboral. Es muy importante destacar que el alumno no sólo desarrolla la práctica propuesta en el laboratorio, si no que en su propio desarrollo se instruye en el uso del equipamiento de test y medida existente. Ya que el desarrollo de sistemas hardware no sólo implica los conocimientos teóricos o fundamentales subyacentes, sino que implica un fuerte conocimiento de los instrumentos necesarios para la fabricación y desarrollo, no siendo esta implicación tan fuerte en los desarrollos software. Los alumnos están comprometidos a hacer un uso responsable del laboratorio, siendo advertidos, que un mal uso del equipamiento repercutirá en una calificación, ya que demuestran así su falta de conocimiento en la materia.

3. Material proporcionado para la elaboración de la práctica

En esta sección pasamos a exponer el material que será proporcionado para la elaboración de la práctica.

3.1. La plataforma USB-AER

La tarjeta USB-AER es una tarjeta basada en una FPGA y diseñada para por el grupo de investigación de Robótica y Tecnología de Computadores (RTS) de la Universidad de Sevilla para permitir la comunicación entre diferentes tipos de dispositivos basado en la comunicación AER. Nos hemos decantado por el uso de esta plataforma porque es muy adecuada para el desarrollo de la práctica, y además los profesores

cuentan con una elevada experiencia en su uso y manejo, así como diversos desarrollos software relativos a la comunicación USB. La plataforma USB-AER dispone de 2MB de memoria RAM estática, con un tiempo de acceso de de 12 nseg, y estructurada en palabras de 32 bits. Además posee una interfaz para tarjetas SD/MMC, en las se puede almacenar el firmware que se cargara en la FPGA, y el contenido inicial de la memoria RAM, permitiendo así que el dispositivo pueda configurarse de forma autónoma sin necesidad de estar conectado a un PC. Adicionalmente, por medio de un puerto USB puede reconfigurarse la tarjeta desde un PC, con fines de depuración, e intercambiar datos con el PC. Usando esta funcionalidad puede usarse la tarjeta USB-AER como puente entre cualquier información que fluya por la FPGA y el bus USB.

La versión de la interfaz USB tiene un ancho de banda de 11 Mbps, aunque el valor exacto depende del tipo de transferencia que se use, la sobrecarga del protocolo, etc. Siendo el ancho de banda efecto suficiente para la mayoría de las aplicaciones. Dado que la plataforma USB-AER se basa en una FPGA, el sistema es completamente reconfigurable, cargando diferentes firmwares para la FPGA, los cuales permiten diferentes usos para el mismo hardware.

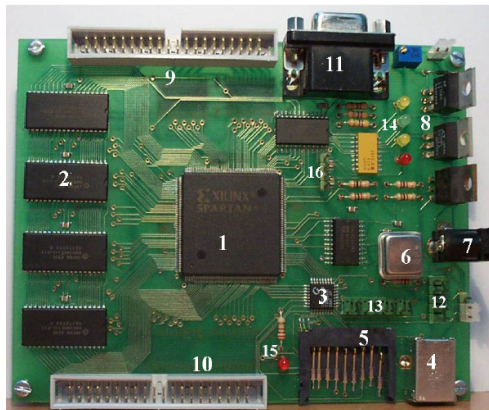


Figura 1: Fotografía de la plataforma USB-AER.

En la figura 1 mostramos una fotografía de la plataforma USB-AER, en ella aparecen identificados los diferentes componentes que la componen:

1. FPGA Spartan II 200.

2. Memoria SRAM. 512Kx32bits.
3. Microcontrolador Cygnal C8051F320.
4. Conector USB.
5. Conector para tarjeta de memoria MMC/SD.
6. Reloj de cuarzo de 50 MHz
7. Conector de alimentación (6V-12V).
8. Fuente de alimentación (2.5V, 3.3V y 5V).
9. Puerto AER de entrada (No usado en esta práctica).
10. Puerto AER de salida (No usado en esta práctica).
11. Puerto VGA de salida (sin DAC) (No usado en esta práctica).
12. Puerto de programación del Cygnal.
13. Conector JTAG de la FPGA.
14. Diodos LED de FPGA.

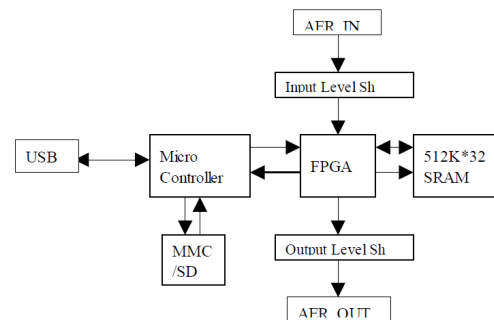


Figura 2: Diagrama de bloques de la plataforma USB-AER.

En la figura 2 mostramos el diagrama de bloques de la plataforma USB-AER. La tarjeta USB-AER se basa fundamentalmente en una FPGA Xilinx Spartan 2, conectada a 4 chips de memoria RAM conformando un total de 512Kx32 bits, con 4 señales de escritura independiente de forma que pueden escribirse palabras de 32 bits, o bytes independiente. Con un reloj de 50Mhz, es posible realizar un acceso cada ciclo a estas memorias (tiempo de acceso de 12 nseg). Por otra parte, un microcontrolador se encarga de cargar el contenido del firmware de la FPGA simulando una memoria EEPROM serie conectada a la FPGA. Aunque como hemos comentado con anterioridad, el firmware podrá cargarse utilizando un fichero almacenado en la tarjeta MMC o bien por el puerto USB. El microcontrolador, Cygnal C8051F320, está

basado en un núcleo 8051 con un interfaz USB 1.1 Full Speed, se encarga del interfaz con la MMC (protocolo SPI) y con un PC a través del puerto USB. Adicionalmente, un bus asíncrono compuesto por 8 líneas de datos, además de algunas señales adicionales de control, interconectan la FPGA con el microcontrolador para el intercambio de datos. Por medio de este bus, el microcontrolador, una vez cargado el firmware en la FPGA, cargará el contenido de la RAM, desde la MMC/SD o el USB. El firmware cargado irá recibiendo los datos desde el microcontrolador, y almacenarlo en la RAM externa de la placa, en la interna de la FPGA o realizar la operación que se necesite. Las señales por las cuales se carga el firmware en la FPGA son distintas a las que se utilizan posteriormente para transferir datos. Un reloj de cuarzo de 50 MHz genera una señal de reloj estable para la FPGA, mientras que el microcontrolador utiliza un oscilador interno de 12 MHz, que mediante un PLL interno se multiplica por 4 para conseguir 48MHz para el USB, y 24 MHz para el procesador 8051. La interconexión entre el microcontrolador y la FPGA se sincroniza por medio de dos señales: strobe y busy. El microcontrolador activa strobe para enviar o recibir datos, mientras que la FPGA puede activar la señal busy para congelar el envío de datos, aunque dado la mayor velocidad de la FPGA sobre el microprocesador esto raramente es necesario.

3.2. Comunicación a través del bus USB

Cuando la tarjeta USB-AER se encuentra conectada a un PC por medio del USB, desde el PC se podrá modificar el firmware contenido en la FPGA, además de enviar o recibir datos desde esta. Puesto que las aplicaciones que pueden cargarse en la FPGA son muy diversas, el protocolo debe ser lo bastante genérico para no tener que ser modificado al desarrollar nuevos diseños. El protocolo está orientado en forma de paquetes o comandos. Antes de cualquier transferencia hay que enviar desde el PC un paquete de control de 64 bytes al microcontrolador (escribir en el dispositivo). El formato de dicho paquete es el siguiente:

Offset	Bytes	Campo	Descripción
00h	3	"ATC"	"Magic Number" para identificar un paquete de control.
03h	1	Command	Comando.
04h	4	Longitud	Numero de bytes que se van a transferir
08h	16	Parámetros	16 bytes de parámetros que se envían a la FPGA.

Figura 3: Formato de las tramas USB.

Actualmente hay definidos 3 comandos para el control de la FPGA. Los comandos implementados son los siguientes:

Código	Comando	Descripción
00h	Update Firmware	Se utiliza para subir un nuevo firmware a la FPGA
01h	Write FPGA	Se utiliza para enviar datos a la FPGA.
02h	Read FPGA	Se utiliza para recibir datos de la FPGA.
03h	SetDevName	Se utiliza para darle un nombre al dispositivo.

Figura 4: Comandos implementados para la comunicación USB.

El campo longitud indica el tamaño de los datos que se van a transferir. En el caso del comando '00' indica el tamaño del firmware que se va a transferir. En los comandos '01' y '02' el tamaño de los datos que se van a transferir. A continuación se transfieren 16 bytes que serán parámetros del comando que se le enviarán a la FPGA. Hay una señal de control entre el microcontrolador y la FPGA que se activará para distinguir si los datos que se transfieren son parámetros o datos. El significado es dependiente del diseño cargado en la FPGA. Si no se necesitan parámetros pueden ser ignorados por la FPGA, aunque el microcontrolador siempre los reenviará a la FPGA. Estos parámetros siempre son enviados por el PC a la FPGA, independientemente de si se trata de un comando de lectura o de escritura. Una vez transferido este paquete de control, dependiendo de la dirección del comando especificado, se transferirán desde el PC a la FPGA o viceversa tantos datos como se indicaron en el campo longitud. Estos datos serán troceados y enviados por el USB en fragmentos de 64 bytes.

3.3. El conversor Analógico Digital: ADC08100

Para el desarrollo de esta práctica se necesita una fuente de video compuesto, ya sea una cámara de video CCD o la salida de video un PC. Para digitalizar la señal de video compuesto se los alumnos conectan la salida de video compuesto producido por una cámara al ADC. Una máquina de estados en el interior de la FPGA se encargará de detectar los sincronismos horizontales y

verticales (sincronismo de frame), para ir tomando las muestras en los instantes adecuados.

Describiendo a grandes rasgos las características de una señal de video compuesto, consiste en transmitir una imagen como niveles de tensión. La imagen se escanea por filas, de manera que el nivel de tensión es proporcional al nivel de luminosidad. Una vez terminada una fila, se envía un pulso de sincronismo, indicado por un nivel de tensión negativo, y se procede con la siguiente fila. Al terminar toda la imagen compuesta por 625 líneas de las cuales solo 575 son visibles, se envía un pulso de sincronismo negativo de mayor duración que el que se empleaba al final de cada fila para indicar el final de un frame o imagen.

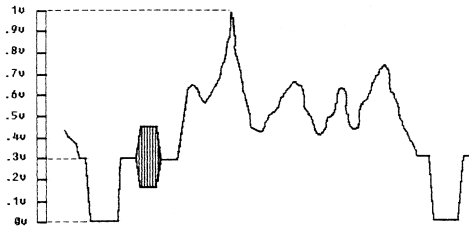


Figura 5: Señal de video compuesto.

El ADC08100 es un conversor analógico/digital que incorpora un circuito de track-and-hold, el cual le permite operara entre 20 y 100Mega muestras por segundo, ofreciendo una excelentes respuestas tanto en AC como en DC, y funcionando con una alimentación de simple rail de 3.3V. Este conversor ADC ha sido diseñado especialmente para aplicaciones de video, imagen y comunicaciones, siendo ideal para el desarrollo de la práctica.

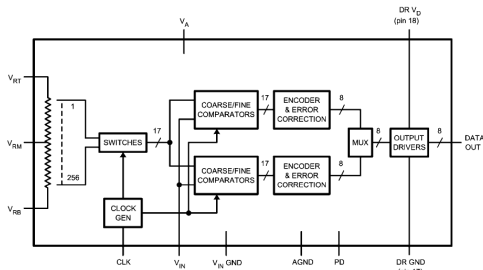


Figura 6: Diagrama de bloques del ADC08100.

El ADC08100 proporciona una salida digital compuesta por 8 bits en paralelo y una señal de reloj, actualizando el valor de cada muestra en el flanco de bajada de una señal de reloj.

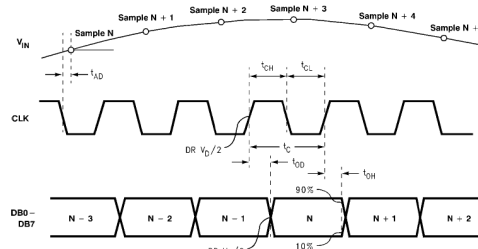


Figura 6: Señal de video compuesto.

En la figura 7 mostramos una fotografía del montaje final de la práctica. En ella puede observarse como se ha conectado la plataforma USB-AER al ADC08100, esta conexión se realiza mediante un bus paralelo de 40 bits, aunque sólo 9 son utilizados. Finalmente como fuente de video los alumnos disponen una pequeña cámara CCD, mediante la cual podrán experimentar y desarrollar satisfactoriamente la práctica.

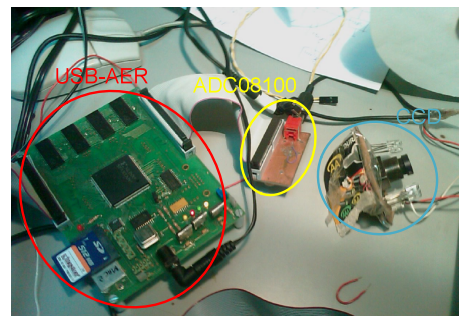


Figura 7: Fotografía del montaje final.

4. Metodología de elaboración y seguimiento

La asignatura se divide en sesiones teóricas y sesiones prácticas, primando fundamentalmente estas últimas. Durante las primeras sesiones (teóricas), se estudia los fundamentos del codiseño hardware-software, donde se separan las funcionalidades del problema que se quiere resolver, asignando responsabilidades a ambas partes. Se hace hincapié en las diferencias entre los lenguajes de programación, como una secuencia de instrucciones que son ejecutadas por un procesador, y los lenguajes de descripción de hardware (HDL), donde lo que se definen son

circuitos digitales, y que por lo tanto no describen un algoritmo, sino un circuito lógico.

En estas sesiones se explica las nociones del lenguaje VHDL, la diferencia entre qué partes del VHDL son sintetizables, y por tanto trasladables a un circuito, y cuáles no, sino que sirven únicamente a la hora de realizar testbench (que son las que se parecen más a un lenguaje de programación, precisamente).

Una vez se les ha enseñado la sintaxis y demás aspectos del VHDL, se pondrán a prueba dichos conocimientos en el laboratorio. Las primeras prácticas se centrarán en la realización de testbench (entidades de prueba utilizadas en la simulación), y al manejo del simulador de vhdl.

El simulador permite realizar varios tipos de simulaciones: behavioural (de comportamiento, sin tener en cuenta los retrasos de la circuitería), post-síntesis y post-placement, donde se tienen en cuenta los retrasos de la circuitería (en el primero), teniendo en cuenta el modelo concreto de FPGA que se va a utilizar, y el más completo de todos, el post-placement, donde se tendrá en cuenta también los retrasos introducidos por las conexiones dentro de la FPGA, y que por tanto dependerá de la colocación física dentro de la FPGA, y la separación entre los distintos elementos.

Una vez se han familiarizado con el simulador, y aprendido a identificar y corregir fallos, basándose en los cronogramas obtenidos con ayuda del simulador, se pasará a realizar las primeras síntesis y a probar los diseños sobre una FPGA real. Una vez se han completado satisfactoriamente estas primeras sesiones prácticas, se pasará a realizar un diseño completo, que es el que se describe en este artículo.

El objetivo de la práctica es desarrollar con ayuda de una FPGA y un convertidor analógico digital externo (capaz de muestrear a una velocidad de hasta 100Mhz), un diseño capaz de capturar la señal proveniente de una cámara de video y almacenarla dentro de la memoria existente en la FPGA. Para ello se les describe las características de la señal de video compuesto, y la funcionalidad de las señales de sincronismo, horizontal y vertical.

Mediante el diseño de una máquina de estados, deberán detectar dichos sincronismos, para realizar las capturas en unos instantes de tiempo determinados. Para simplificar el

contenido de la práctica, se supondrá que la señal de video es monocromática, obviándose la parte de la demodulación de la sub-portadora de color.

Para ayudarles en el desarrollo de esta parte, se les facilitará junto con la documentación de la práctica, un diagrama con el funcionamiento de la máquina de estados que deben implementar, así como unos esqueletos de algunas partes del diseño ya planteados, que ellos deberán completar. Para esta parte del ejercicio se cuenta con dos sesiones de prácticas, durante las cuales podrán consultar al profesor las dudas que les vayan surgiendo.

Durante la semana que transcurre entre las dos sesiones de prácticas, los alumnos podrán acceder a un laboratorio de prácticas habilitado para tal fin, donde podrán continuar el trabajo con el mismo material que se ha utilizado en las prácticas. A diferencia de otras asignaturas, donde al alumno se le ofrece a modo de préstamo el material necesario para que puedan trabajar en casa, debido al número de elementos necesarios para un puesto de trabajo (y por tanto, al elevado coste que representa), se ha estimado más conveniente en que los alumnos pudieran acceder a él en el laboratorio. Entre el material que necesitan podemos señalar la placa con la FPGA, la placa con el ADC externo, programador USB de Xilinx, fuente de alimentación, osciloscopio, etc.

En la siguiente sesión de prácticas, deberán interconectar la FPGA con el microcontrolador, utilizando un protocolo prefijado, que permitirá una vez terminada la captura, el ir leyendo secuencialmente el contenido capturado en la memoria e ir transmitiéndolo al ordenador utilizando el bus USB. El microcontrolador empleado, un Silabs C8051F320 (basado en el núcleo de Intel del 8051), tiene integrado un controlador USB. La parte del código del microcontrolador empleado que se encarga de la gestión de los mensajes del bus USB se le proporciona al alumno, como parte del material con el que contarán para realizar la práctica, permitiéndoles centrarse en el aspecto de interconexión entre el microcontrolador y la FPGA.

Para poder comprobar que el diseño que están haciendo se comporta satisfactoriamente, se les proporcionará un programa de ejemplo con el que podrán mostrar en la pantalla del ordenador el contenido descargado por el puerto USB. Aunque

en una práctica posterior deberán ellos mismos realizar un pequeño programa de ejemplo que realice estas mismas funciones, utilizando unas librerías que se les suministra. Se ha creído conveniente que contarán en un primer momento con un programa ya funcional para facilitar la detección de errores en la parte que están desarrollando, evitando la frustración de no conseguir los resultados esperados, sin saber en cuál de las componentes se encuentra localizado el fallo.

De esta forma, al ir desarrollando la práctica de manera progresiva, contando en cada paso con las herramientas necesarias para probar el diseño que están realizando de forma aislada, se facilita enormemente la localización de fallos, evitando que el alumno se sienta frustrado y pierda la motivación al emplear mucho tiempo buscando errores sin saber en qué parte de la cadena se encuentra. Esto permite que se pueda plantear este tipo de prácticas, con un nivel de dificultad que de otra forma, si tuvieran que empezar todo el diseño desde cero, no podrían plantearse, o que en caso de plantearse, requerirían un número de sesiones prácticas y un trabajo añadido del alumno fuera de las mismas en horas de laboratorio, demasiado alto. Además, el número de alumnos que perderían la motivación al no ir obteniendo resultados visibles de forma progresiva, sería también muy elevado, repercutiendo negativamente en su proceso de aprendizaje.

Esta práctica ha sido positivamente acogida por parte de los alumnos, en el momento que se encuentra estructurada como distintas prácticas con hitos intermedios. Así mismo, fomenta el intercambio de ideas, a la hora de encontrar soluciones a los problemas que se van encontrando.

Esta metodología de aprendizaje basado en problemas asegura además un mejor aprendizaje por parte del alumno, respecto al modelo tradicional de enseñanza donde las sesiones prácticas tenían una importancia superior a las sesiones prácticas.

5. Conclusiones

En este trabajo se ha presentado el enfoque adoptado para la impartición de la asignatura de co-diseño en VHDL, intentando realizar una aproximación progresiva y amena al contenido de

la práctica, para intentar despertar la curiosidad del alumno, a la vez que se profundiza en el estudio del co-diseño software-hardware. Parte de la práctica se desarrolla realizando el desarrollo de un circuito descrito en VHDL, sobre una FPGA, que se encargará de la captura de imágenes proveniente de una cámara de video. Una vez terminada la parte hardware, deberán interconectarlo con una aplicación en MATLAB, o cualquier lenguaje de alto nivel, utilizando unas librerías que manejan el bus USB.

Se pretende así que el alumno tenga una visión completa de lo que sería una cadena completa entre un circuito electrónico (FPGA), su interfaz con un microcontrolador, que hace de interfaz con el ordenador por medio del USB, y por último, del manejo de dicha información desde un lenguaje de alto nivel. Algunas capas, como el driver del dispositivo USB, no se han tratado durante esta práctica, debido a que se escapaba de los objetivos del curso. Sin embargo, no siempre es necesario el desarrollo de un nuevo driver. En este caso, se ha utilizado un driver proporcionado por el fabricante del microcontrolador.

También se ha conseguido que el alumno entienda y se familiarice con la señal de video compuesto y sus características, ayudándoles a entender cómo funcionan las señales de video analógicas, que describen una secuencia de imágenes mediante su recorrido por filas.

A pesar de la aparente complejidad de dicha práctica, se les ha dado en forma de una lista o guía, con diversos puntos que debían ir resolviendo paso a paso, hasta su completa resolución. Se ha hecho hincapié en los aspectos que podían tener mayor dificultad, aportándole pistas o información adicional para superar satisfactoriamente el ejercicio. Ha sido bien acogida por los alumnos y la mayoría ha resuelto sin demasiada dificultad la totalidad de la práctica.

Agradecimientos

Este proyecto ha sido posible en parte gracias al proyecto VULCANO (TEC2009-10639-C04-02).

Referencias

- [1] M. Valero-García, J. Navarro, FAQ sobre la docencia centrada en el aprendizaje.

- <http://epsc.upc.edu/~miguel%20valero/contingut3.html>.
- [2] Universidad de Sevilla. Plan de Renovación de Metodologías Docentes 2007. http://www.institucional.us.es/plan_renovación.
- [3] T. Markham Project Based Learning, a guide to Standard-focused Project based learning for middle and high school teachers. *Buck Institute for Education, (2003)*.
- [4] G. Solomon. Project Based Learning: a Primer, *Technology Learning*, vol. 23, num. 6 (Jan 2003).
- [5] H. Dolmans, et al. Problem-based learning: future challenges for educational practice and research. Blackwell Publishing Ltd 2005. *Medical Education* (2005); 39.
- [6] M. Hedley, S. Barrie. An undergraduate microcontroller systems laboratory. *IEEE Transactions on Education*, Volume: 41, Issue: 4, (Nov.1998).
- [7] M. Valero-García. Las dificultades que tienes cuando haces PBL. Cap. 8 La Educación Superior hacia la Convergencia Europea.

Entorno de soporte para el autoaprendizaje en el diseño de circuitos digitales

D. Bañeres, I. Bermejo, R. Clarisó, J. Jorba, M. Serra, F. Santanach, A. Rodríguez
Universitat Oberta de Catalunya

Rambla del Poblenou, 156, 08018, Barcelona

{dbaneres,rclariso,jjorba,mserravi,fsantanach,jrodriguezarr}@uoc.edu,ivan20@gmail.com

Resumen

El diseño de circuitos digitales forma parte de las competencias básicas de los nuevos Grados en Ingeniería Informática e Ingeniería de Telecomunicaciones. Un obstáculo importante para el aprendizaje de dichas competencias es que las herramientas académicas existentes para el diseño de circuitos no permiten validar si un diseño se ajusta a la especificación de partida. En este artículo, se describe un entorno de autoaprendizaje para que los estudiantes puedan realizar ejercicios de diseño de circuitos y recibir un feedback continuo.

Summary

The design of digital circuits is a basic competence of the new Degrees in Computer Science and Engineering of Telecommunications. An important hindrance in the learning process of these skills is that the existing academic tools for the design of circuits do not allow the student to validate if his design satisfies the specification. In this article, we describe an online environment where the students can verify their designs with an automatic feedback.

Palabras clave

Circuitos, verificación, autoevaluación, educación a distancia, Model Checking

1. Motivación

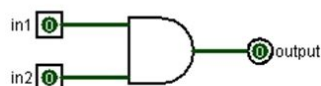
En los nuevos grados del espacio europeo de Ingeniería Informática e Ingeniería de Telecomunicaciones, los alumnos desarrollan las competencias relacionadas con los principios básicos del funcionamiento del mundo digital. Una competencia básica a adquirir es saber aplicar técnicas de análisis y síntesis de circuitos digitales sencillos. Es esencial que los alumnos sean capaces de diseñar circuitos digitales antes de ampliar sus conocimientos en las posteriores asignaturas de los respectivos grados.

El profesorado de esta área tiene muchas dificultades para diseñar el modelo de aprendizaje de las

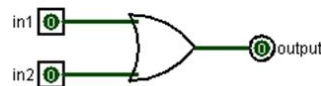
Enunciado

Diseña un circuito con dos entradas (**in1**, **in2**) cuya salida (**output**) tenga valor 1 si y sólo si ambas entradas tienen valor 1.

Solución correcta



Diseño estudiante



Feedback

INCORRECTO para la traza de entrada
 $in1 = (0)$, $in2 = (1)$

Figura 1: Ejemplo de uso de herramientas de verificación en un ejercicio de diseño de circuitos.

técnicas de diseño de circuitos. Normalmente, esta competencia está ligada a asignaturas obligatorias de primer curso (como por ejemplo *Fundamentos de Computadores*) con un alto número de estudiantes e históricos importantes sobre el nivel de abandono y de no superación de la asignatura. Un ejercicio "tipo" en este ámbito docente consiste en diseñar diversos circuitos a partir de una especificación de partida. Ésta se basa en un conjunto de entradas, de salidas y el comportamiento final deseado para el circuito en relación a estas señales. Al estudiante se le pide que describa los componentes del circuito que implementa este comportamiento. Las actividades relacionadas con el diseño de circuitos no son un proceso mecánico, sino más bien creativo y se puede obtener más de una solución válida. Por eso, es vital que los alumnos puedan desarrollar estas prácticas recibiendo un feedback constante que les permita evaluar el progreso de su aprendizaje. Habitualmente, para el profesorado es difícil dar un feedback inmediato, constante e individual a todos los estudiantes de una aula numerosa. Además

el problema es significativamente mayor cuando el sistema de aprendizaje es a distancia.

Esta inquietud surge también en la docencia de otros ámbitos temáticos dentro de la Informática. El trabajo en este campo ha dado lugar al desarrollo de diferentes herramientas para la autoevaluación de ejercicios, como por ejemplo LEARN-SQL [1] (bases de datos), ACME [11] (matemáticas, estadística, bases de datos y otros), Judge.org (programación) [10] o Pl-Man (lógica) [7], entre otros. La mayoría de estas herramientas se basan en la comprobación del resultado de *juegos de prueba*. Sin embargo, estas herramientas no resultan apropiadas para la autoevaluación de circuitos lógicos: el número de entradas posibles es muy elevado y no resulta factible definir el resultado esperado para cada una de ellas. Tampoco es posible identificar un subconjunto de casos "representativos".

Como ejemplo, consideremos un ejercicio de diseño de circuitos como el que se muestra en la Figura 1. El enunciado pide el diseño de una puerta AND. En la parte superior de la figura se muestra la solución correcta al ejercicio. Imaginemos que un estudiante propone la solución errónea que se muestra en la parte inferior de la figura, utilizando una puerta OR. Proponemos una herramienta que permita al estudiante comprobar por sí mismo si su solución es correcta y que, en caso de error, le proporcione como resultado un conjunto de valores para las entradas que ejemplifique el fallo y permita depurar el diseño. En este caso, la herramienta indicaría el error dando como ejemplo los valores de entrada $in1 = 1$ y $in2 = 0$, en los que el valor de la salida obtenida (1) difiere de la esperada en el enunciado (0). Este ejemplo es trivial y sólo se utiliza para ilustrar el proceso de resolución. Los ejercicios reales de diseño de circuitos tienen enunciados más complejos, que pueden llegar a tener decenas de puertas lógicas y otros elementos como registros y multiplexores. En este escenario, hay múltiples soluciones posibles y no es trivial identificar posibles errores, por lo que el feedback obtenido mediante una herramienta de autocorrección puede ser muy valioso.

Actualmente existen varias herramientas académicas mediante las cuales los alumnos pueden diseñar gráficamente circuitos lógicos [?, ?, ?]. Estas herramientas son muy intuitivas de utilizar pero tienen un inconveniente: "no se puede determinar si un diseño es válido a partir de una especificación dada".

Para realizar esta verificación existen herramientas industriales de diseño y verificación [14]. No obstante, su utilización presenta varios inconvenientes: su uso en asignaturas introductorias no es adecuado debido a su complejidad y alto coste económico. También existe otro tipo de herramientas que muestran de forma automática el diseño de circuitos a partir de un mapa de Karnaugh o grafo de estados [6] pero no permiten verificar si el diseño proporcionado por un alumno es correcto a partir de la especificación.

Precisamente para aportar solución a esta problemática, se propone un entorno virtual de autoaprendizaje que permita a los estudiantes una autoevaluación constante de sus diseños de circuitos.

El entorno tiene las siguientes características:

- El sistema de verificación es similar al utilizado en la industria de fabricación de microchips [14], aunque a menor escala. Se ha desarrollado una herramienta académica de verificación basada en analizar los circuitos mediante técnicas de Model Checking [8].
- La herramienta verifica si dos circuitos son *funcionalmente* iguales, es decir, se comprueba si tienen el mismo *comportamiento* (proporcionando un contraejemplo en caso contrario).
- El alumno utiliza un software de diseño y simulación de circuitos [2] que permite dibujar el circuito. Internamente, el proceso de verificación compara este diseño con la solución oficial mediante Model Checking.

Después de presentar los objetivos docentes (Sección 2) y la arquitectura del entorno (Sección 3), este artículo describe la experiencia de su aplicación en una asignatura obligatoria con el respectivo análisis de los resultados (Sección 4).

2. Objetivos docentes

El principal objetivo es proporcionar un entorno virtual para la autocorrección de ejercicios de diseño de circuitos con disponibilidad 24x7 (las 24 horas, los 7 días de la semana). Mediante este entorno, los estudiantes pueden comprobar de forma automática si su diseño es correcto. Además de facilitar el proceso de autoevaluación, la plataforma permite tener un control sobre la evolución de los alumnos en todo momento registrando su utilización y los patrones de uso. En universidades virtuales, como es el caso de la Universitat Oberta de Catalunya (UOC) [13], este

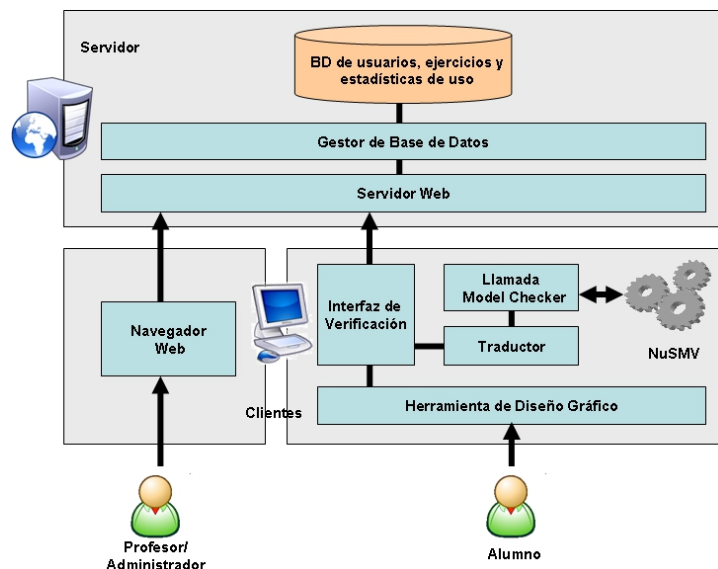


Figura 2: Arquitectura del entorno de verificación.

entorno permite incidir en el autoaprendizaje de los alumnos en competencias complejas.

En las asignaturas de primer curso donde se desarrolla la competencia de diseño de circuitos digitales (en nuestro caso, *Fundamentos de Computadores*), los estudiantes utilizan una herramienta de diseño gráfico de circuitos [2]. Esta herramienta les permite diseñar gráficamente los circuitos sin necesidad de tener conocimientos de lenguajes de descripción de hardware [9,15]. Además, les permite simular de forma manual su diseño para valores concretos en las entradas. En este proceso de aprendizaje creemos fundamental que los estudiantes puedan recibir un feedback inmediato en el momento de resolver un ejercicio de diseño. El diseño de circuitos no es una actividad repetitiva, sino que requiere un alto grado de creatividad que debe adquirirse con la práctica. El feedback permite identificar los errores para corregirlos.

Este entorno permite el autoaprendizaje y autoevaluación de los estudiantes de una forma planificada. También se puede controlar en todo momento los ejercicios disponibles. De esta forma se puede planificar el conjunto de ejercicios que los alumnos tienen disponibles, dependiendo del material docente asignado para cada una de las semanas. Mediante este control de la información disponible y junto con los datos estadísticos, se puede conocer en todo

momento la evolución individual de los alumnos y de la evolución colectiva de las aulas. Esta información permite al profesorado detectar las principales carencias de los estudiantes en los aspectos de diseño para poder incidir en los puntos débiles que se observen.

Creemos que este entorno puede potenciar la dedicación del estudiante al aprendizaje de las competencias del ámbito de tecnologías de computadores. Para potenciar y motivar aún más la introducción de esta plataforma en el aula, se propone asignar un peso en la evaluación de la asignatura a actividades prácticas ligadas a la utilización de esta plataforma.

En concreto, se pretende que este entorno facilite a los estudiantes el aprendizaje de la asignatura y que esto se vea reflejado en los resultados de la evaluación. En particular, se ha considerado tres métricas a evaluar: la *tasa de abandono* de la asignatura, la *tasa de aprobados* y la *nota media*.

3. Descripción de la plataforma

En esta sección, se describe el entorno de soporte al autoaprendizaje desarrollado. En primer lugar, se presenta la arquitectura de la plataforma. En segundo lugar, se explicita cómo se realiza el proceso de verificación.

3.1. Arquitectura

La plataforma utiliza un modelo cliente-servidor tal como se muestra en la Figura 2. En el servidor, el sistema está formado por una base de datos y un servidor Web y de aplicaciones (Apache con Tomcat) que hace de interfaz para comunicarse con los clientes. Internamente, la base de datos almacena la información sobre los usuarios y las colecciones de ejercicios existentes. Además, el sistema guarda los diseños enviados por los estudiantes como posible solución, para permitir al docente examinar el progreso de los estudiantes.

Los usuarios pueden acceder al entorno por dos vías diferentes: por navegador Web o mediante el programa de diseño gráfico Logisim [2], dependiendo de su perfil. Se han considerado dos perfiles de usuario diferentes en esta aplicación:

- Perfil de alumno: El alumno puede acceder a la plataforma mediante el programa de diseño de circuitos lógicos Logisim. El alumno puede ver en todo momento los ejercicios disponibles y su evolución en la resolución de los mismos.
- Perfil de profesor/administrador: El profesor accede a la plataforma mediante un interfaz Web. Mediante este interfaz se pueden añadir/modificar ejercicios, consultar los ejercicios resueltos por los estudiantes y ver estadísticas sobre la evolución de una aula o de un alumno en concreto. También se tienen permisos tanto para modificar los usuarios del sistema como para modificar las aulas disponibles.

Centrándonos en los estudiantes, el programa Logisim que utilizan ha sido adaptado con la introducción de un nuevo módulo (*VERILUOC*) que se puede ver en la Figura 5-(a). Este nuevo interfaz permite visualizar los ejercicios disponibles y permite verificar el circuito diseñado en ese momento mediante Logisim. En las siguientes secciones, describiremos el proceso de verificación junto con un ejemplo.

3.2. Proceso de verificación

Como se puede observar en la Figura 2, el proceso de verificación se produce en el ordenador del estudiante. Esta decisión se justifica para garantizar la escalabilidad de la plataforma: el proceso de verificación de un circuito requiere un cálculo intensivo, por lo que realizar este proceso en el servidor podría

dificultar el acceso de los estudiantes al entorno durante los picos de utilización (como las fechas de entrega de las prácticas). Realizando la verificación en el cliente se evitan posibles sobrecargas del servidor y se garantiza la prestación de servicio. Es necesario observar que se han implementado medidas especiales de seguridad para que la solución propuesta por el equipo docente no sea visible para el estudiante ni sea almacenada en su ordenador.

Para el proceso de verificación, se estudiaron dos posibles implementaciones alternativas. La primera era utilizar técnicas de simulación y la segunda utilizar técnicas basadas en Model Checking. En la primera opción se debería utilizar algún intérprete de lenguaje de descripción de Hardware¹ para simular la ejecución de los circuitos. En este caso, se construye un test que simula para los dos circuitos todos los posibles valores en las entradas. Para cada valor se comprueba si el valor de cada una de las salidas coincide. Este tipo de simulación es bastante rápido para circuitos pequeños, pero aumenta exponencialmente el tiempo y la complejidad para circuitos con muchas entradas. Además, el problema se agudiza en el momento de verificar circuitos secuenciales.

La segunda técnica, que hemos seleccionado para esta plataforma, se fundamenta en Model Checking. La ventaja respecto a la anterior, es su eficiencia en tiempo en el proceso de verificación tanto en circuitos combinatoriales como secuenciales. La técnica de Model Checking consiste en comprobar si un modelo cumple una condición dada. En nuestro caso (véase Figura 3), se traduce en construir a partir de los dos circuitos que se quiere comparar un nuevo circuito con las mismas entradas y con una única salida con la función $out_v = \bigwedge_{i=1}^n out1_i \oplus out2_i$ donde $out1_i$ y $out2_i$ son las salidas i del circuito 1 y circuito 2 respectivamente, n el número de salidas del circuito y \oplus la operación XNOR. Este circuito deberá cumplir que la salida out_v siempre obtenga el valor 1. Model Checking comprobará de forma eficiente si esta condición se cumple para todos los posibles valores de entrada y, en el caso de circuitos secuenciales, para todos los posibles estados.

El proceso de verificación se puede observar en el diagrama de flujo de la Figura 4. Inicialmente, la solución proporcionada por el equipo docente está en formato Verilog. Actualmente, Logisim guarda los

¹En el caso de utilizar como lenguaje de descripción Verilog, se puede utilizar algún software libre como Icarus [16].

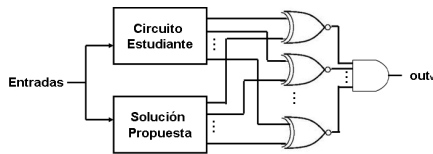


Figura 3: Modelo de verificación con Model Checking.

circuitos en un formato propio no compatible con ningún lenguaje de descripción de Hardware (formato CIRC). Con el propósito de desarrollar un verificador lo más independiente posible, existe el parser *CIRCtoVER* que traduce el formato de descripción de Logisim a Verilog (en la Figura 2 se define como *traductor*). En el momento en que los dos circuitos a verificar ya están en formato Verilog se puede iniciar el módulo de verificación *DiCiCheck* (**D**igital **C**ircuit **C**hecking) que en la Figura 2 se define como *Llamada Model Checker*. Inicialmente, este módulo construye el modelo explicado anteriormente utilizando el módulo *VERtoSMV*. Un model checker² verificará si la condición propuesta se cumple sobre el modelo. Posteriormente, el sistema analiza la salida del model checker que en caso de no ser funcionalmente equivalentes ya devuelve una traza indicando para que valores de las entradas se ha detectado el error. Esta traza se indica al estudiante en la pestaña de *Verificación* en el interfaz *VERILUOC*. Nótese que en caso de ser un circuito secuencial devuelve el conjunto de estados que ha verificado correctamente hasta encontrar el error. Mediante esta información, el alumno recibe el feedback y puede comprobar manualmente en Logisim los valores de la traza problemática y corregir su diseño.

3.3. Ejemplo de verificación

En la Figura 5 se presenta, a modo de ejemplo, la verificación del circuito mostrado en la Sección de Motivación. El objetivo es diseñar un circuito que produzca el producto lógico de dos entradas. Las entradas se identifican por *in1* y *in2* y la salida por *out put*. En la Figura 5-(a) podemos ver el interfaz *VERILUOC* con los posibles circuitos que se pueden verificar en la tabla superior. La tabla resume el número de intentos totales, fallidos y correctos realizados para cada ejercicio. La parte inferior se divide en dos pestañas: en la primera aparece una descripción textual del ejercicio, mientras que en la segunda

²Nuestro verificador utiliza el model checker NuSMV [4].

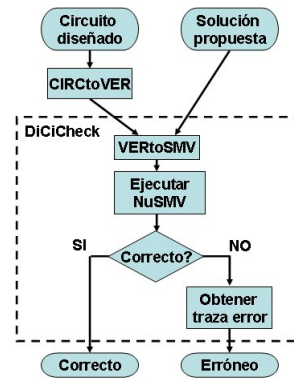


Figura 4: Proceso de verificación.

aparece el resultado de la verificación. Al seleccionar el circuito *EjemploAND2*, en la pestaña *Descripción* se muestra el enunciado del problema.

A partir de este enunciado, se diseña una posible solución (Véase Figura 5-(b)). En este caso, se ha implementado el circuito erróneamente a propósito, usando una puerta OR. Una vez terminado el diseño, se puede verificar mediante el botón *Verificar*. Antes de verificar el comportamiento del circuito, el verificador comprueba si las entradas y salidas se han definido acorde con el enunciado. En caso contrario, notifica este error al estudiante inmediatamente. En la Figura 5-(c) se puede observar el resultado en caso de error que aparece en la pestaña *Verificación*. El verificador devuelve un conjunto de valores para las entradas que producen un valor erróneo en las salidas. Con este feedback, el usuario puede simular con Logisim qué salida produce su diseño para esos valores de entrada. En este caso, el error es evidente. Si se modifica el diseño cambiando la puerta OR por una AND y se verifica de nuevo, el resultado correcto se puede observar en la Figura 5-(d).

4. Resultados

4.1. Prueba piloto

Para evaluar el impacto de la plataforma en el aprendizaje, se ha realizado una prueba piloto en una aula de la asignatura de *Fundamentos de Computadores* del Grado en Ingeniería Informática de la UOC [13]. La asignatura consta de 245 alumnos, los cuales se encuentran divididos en 4 aulas. El aula escogida para realizar la prueba piloto consta de 61 estudiantes.

La evaluación de esta asignatura se realiza me-

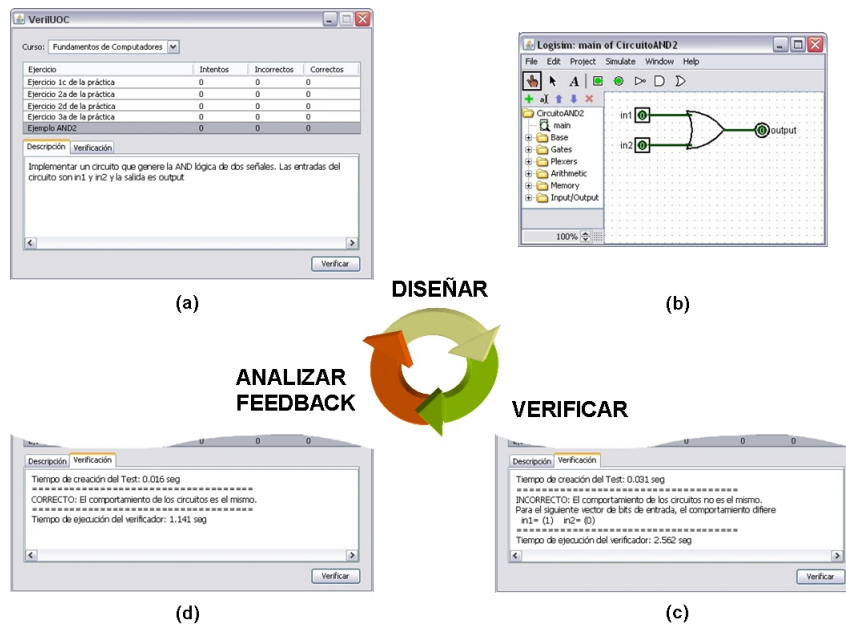


Figura 5: Ejemplo de verificación. (a) Interfaz VERILUOC de verificación. (b) Diseño erróneo de un producto lógico de dos entradas. (c) Resultado de la herramienta del diseño proporcionado. (d) Resultado de la herramienta al diseñar el circuito correcto (con una puerta lógica AND de 2 entradas).

diante una evaluación continua con 3 pruebas, una práctica y un examen final. La nota final se calcula de la siguiente forma $20\%EC + 40\%P + 40\%EX$, donde EC , P y EX son las notas de la evaluación continua, la práctica y el examen respectivamente. La práctica, con una duración aproximada de un mes, se realiza al final del semestre y permite relacionar todos los conceptos aprendidos en la asignatura. Concretamente, en ella se ejercitan los sistemas de numeración, circuitos combinatoriales y secuenciales. Como parte de esta práctica, el estudiante debe diseñar un conjunto de circuitos que al final se combinan en un sistema complejo.

Para realizar la práctica, el estudiante recibe el enunciado, el programa Logisim modificado y un manual de instalación y uso de Logisim y VERILUOC. Mediante la plataforma VERILUOC, los estudiantes tienen la posibilidad de verificar 4 ejercicios de la práctica con un peso del 65% (el 35% restante está asignado a ejercicios no relacionados con diseño de circuitos). El uso del entorno en esta prueba piloto no ha sido obligatoria, pero se ha explicado al alumnado que su utilización ayuda significativamente a la superación de la práctica.

4.2. Evaluación de la prueba

Después de realizar la prueba, hemos analizado el rendimiento de los estudiantes en la práctica. Nótese que, finalmente, sólo un número reducido de estudiantes han utilizado la plataforma: 1) El total de alumnos que ha presentado la práctica han sido 31 ya que el resto no habían superado la evaluación continua (en esta asignatura hay un gran índice de abandono de estudiantes cuando no superan la evaluación continua). 2) El total que alumnos que han utilizado la herramienta han sido 14. Consideramos que una de las razones ha sido la no obligatoriedad de su uso. Por lo tanto, los resultados que presentamos no son muy significativos pero muestran unos primeros indicios del potencial de la herramienta.

En el Cuadro 1 podemos ver una comparación de las notas de prácticas de los estudiantes que han utilizado la plataforma con respecto al total de estudiantes de la asignatura. Aunque el número de alumnos que la han utilizado es reducido (50% de los que han presentado la práctica), podemos ver que el porcentaje de estudiantes con notas de Excelente y Notable supera significativamente el global de la asignatura. La iteración en la resolución de los ejercicios a partir del feedback del entorno ayuda a obtener la

Calificación	% de estudiantes	
	VERILUOC	Global
Excelente	58 %	18 %
Notable	21 %	9 %
Aprobado	7 %	9 %
Suspense	7 %	6 %
Muy Deficiente	0 %	5 %
No Presentado	7 %	53 %

Cuadro 1: Calificaciones en la práctica

solución correcta y obtener la máxima puntuación en los ejercicios donde podía usarse el verificador. Analizando las estadísticas de uso del entorno, hemos observado que los alumnos han utilizado un promedio de entre 3 a 6 intentos (según el ejercicio) hasta obtener la solución correcta. Este dato nos confirma que el feedback en la mayoría de los casos es positivo y ayuda a resolver algún caso particular que los estudiantes no habían tenido en cuenta.

Finalmente, también hemos analizado el resultado de superación de la asignatura (véase Cuadro 2). Teniendo en cuenta que el uso de la plataforma no era obligatorio, se ha realizado la comparación del global de la asignatura con respecto del aula donde se ha hecho la prueba piloto. Podemos observar que el porcentaje de aprobados sobre presentados se ve incrementado significativamente. También existe un descenso en el porcentaje de abandono de la asignatura. Si calculamos la nota media, podemos observar un ligero incremento en el aula de la prueba piloto. Si calculamos la nota media únicamente de los alumnos que ha utilizado la plataforma, podemos ver que la nota media es de 8,27. Este resultado final, nos confirma que la utilización de este entorno puede ayudar a adquirir las competencias de esta asignatura y así incrementar el rendimiento de la misma.

4.3. Valoración de los estudiantes

Hemos realizado una encuesta a los estudiantes para recoger su opinión. Todos concluyen que les ha ayudado en la realización de la práctica, pero han tenido diferentes dificultades en su utilización.

- Dificultades en la instalación: El programa Logisim está implementado en JAVA, por lo tanto, es multiplataforma. El inconveniente que conlleva la utilización del entorno es que se debe instalar un programa complementario para la verificación (el model checker NuSMV [4]). El manual proporcionado a los estudiantes descri-

Métrica	Aula	Todas las
	VERILUOC	Aulas
Present. / Total	49 %	47 %
Aprobados / Present.	93 %	77 %
Nota Media (Present.)	6,97	6,56

Cuadro 2: Resultados finales asignatura

be paso a paso la instalación, pero los estudiantes se han encontrado con algunos problemas inesperados durante la instalación que dependen sobretodo del sistema operativo utilizado.

- Dificultades en la interpretación del feedback: En algunos casos los estudiantes no han sabido entender el error de su diseño a partir del feedback. Este problema aún es más evidente en el caso de los circuitos secuenciales, en los cuales se devuelve el conjunto de estados correctos hasta encontrar el erróneo.
- Dificultades en la utilización de Logisim: El programa es muy intuitivo de utilizar, pero hay algunos componentes que no están bien explicados en los manuales, por lo que los alumnos tienen dudas sobre su uso.

Para resolver estos problemas de forma global, proponemos mejorar el manual de instalación y de uso de entorno. Otra posibilidad es crear un Wiki para almacenar toda esta información electrónicamente junto con una nueva sección de "Preguntas más frecuentes" con los problemas más comunes.

4.4. Valoración del equipo docente

El equipo docente ha detectado algunos inconvenientes en la utilización del entorno:

- Casos de plagio: Este temor aparece en la mayoría de herramientas automáticas. El uso de las TIC en la enseñanza simplifica los métodos de copia de ejercicios.
- Mal uso del verificador: Ciertos estudiantes pueden hacer un mal uso del verificador usándolo como un mecanismo de "prueba y error" sin analizar realmente el feedback devuelto.

Para resolver estos problemas potenciales proponemos algunas soluciones. En el primer caso, se debería analizar, posteriormente a la entrega, el índice de similitud entre los diseños enviados por los estudiantes. Existen herramientas como JPlag [12] para realizar esta comprobación. Para controlar el mal

uso, proponemos limitar el número de verificaciones en un cierto periodo de tiempo.

5. Conclusiones y trabajo futuro

En este artículo hemos presentado un entorno para la verificación automática de circuitos lógicos y resultados preliminares con mejoras de los rendimientos en la asignatura relacionada.

Consideramos que hay dos aspectos destacables de esta aproximación. El primero es la integración de la plataforma con la herramienta de diseño de circuitos usada en la asignatura, de forma que el estudiante puede aprovechar la autocorrección sin tener que salir de la herramienta. El segundo aspecto es la utilización de Model Checking para la verificación de los circuitos. Esta técnica proporciona feedback al estudiante y, a diferencia de otros entornos para la corrección automática, no requiere hacer el esfuerzo de definir juegos de prueba ni se basa en la comparación directa de un circuito con otro (con lo que se permiten circuitos diferentes pero con el mismo comportamiento).

Este entorno aún tiene algunos inconvenientes en su utilización, los cuales han sido detectados y recogidos en la prueba piloto. Además, serán revisados en futuras versiones. Proponemos que en un futuro haya un repositorio de ejercicios de autoevaluación adicionales que ayuden a los alumnos a reforzar los conocimientos adquiridos.

Aunque nuestro primer objetivo ha sido utilizar el entorno en la asignatura de *Fundamentos de Computadores*, el verificador también es posible su utilización en otras asignaturas más avanzadas dentro del mismo ámbito, como *Electrónica Digital*. Una ventaja del analizador es que, al ser dependiente únicamente de Verilog, es posible utilizarlo en asignaturas donde se diseñen circuitos más complejos descritos con lenguajes de descripción de Hardware como Verilog [15], o bien VHDL [9].

Agradecimientos

Este trabajo ha sido financiado con la ayuda APLICIA 2010 de la Univ. Oberta de Catalunya.

Referencias

- [1] A. Abelló, M. E. Rodríguez, T. Urpí, X. Burgués, M. J. Casany, C. Martí, and C. Quer. LEARN-SQL: Automatic assessment of SQL based on IMS QTI specification. *ICALT*, pages 592–593, 2008.
- [2] C. Burch. Logisim, a graphical tool for designing and simulating logic circuits. <http://ozark.hendrix.edu/~burch/logisim/>, 2010.
- [3] CEDAR. CEDAR logic simulator. <http://cedarlogic.scienceontheweb.net>, 2010.
- [4] Cimatti, E. M. Clarke, E. Giunchiglia, F. Giunchiglia, M. Pistore, M. Roveri, R. Sebastiani, and A. Tacchella. NuSMV 2: An opensource tool for symbolic model checking. In *Computer Aided Verification*, volume 2404, pages 241–268. Springer Berlin, 2002.
- [5] Capilano Computing. *LogicWorks 5 Interactive Software*. Prentice Hall, 2004.
- [6] J. García, J. Sanz, and B. Sotomayor. A new approach to educational software for logic analysis and design. In *International Conference on Education (IADAT e2004)*, 2004.
- [7] M. J. Castel De Haro, F. Gallego, C. Pomares, P. Suau, C. J. Villagrà, and S. Cortés. e-VALUACIÓN en tiempo real. In *JENUI*, 2009.
- [8] K. L. McMillan. *Symbolic Model Checking*. Kluwer, 1993.
- [9] V. A. Pedroni. *Circuit Design with VHDL*. The MIT Press, 2004.
- [10] J. Petit and S. Roura. Programación-1: Una asignatura orientada a la resolución de problemas. In *JENUI 2009*, 2009.
- [11] F. Prados, I. Boada, J. Soler, and J. Poch. Automatic generation and correction of technical exercises. In *ICECE'05*, 2005.
- [12] L. Prechelt, G. Malpohl, and M. Philippsen. Finding plagiarisms among a set of programs with JPlag. *Journal of Universal Computer Science*, 8(11):1016–1038, 2002.
- [13] M. Serra, E. Santamaria, M.A. Rius, M. A. Huertas, and M. E. Rodríguez. Nuevos paradigmas de la educación: roles de acción docente en el entorno virtual de la UOC. In *CEDI'05*, 2005.
- [14] Synopsys. Synopsys EDA tools. <http://www.synopsys.com>, 2010.
- [15] F. Vahid. *Verilog for Digital Design*. Wiley, 2007.
- [16] S. Williams. Icarus: a free compiler for Verilog. <http://bleyer.org/icarus/>, 2010.

Animaciones interactivas para la enseñanza y aprendizaje de los protocolos de coherencia de cachés*

Alberto Alcón Laguéns, Sergio Barrachina Mir, Enrique S. Quintana Ortí
Dpto. de Ingeniería y Ciencias de los Computadores
Universidad Jaume I
albertoalcon@gmail.com, {barrachi,quintana}@icc.uji.es

Resumen

Entre los objetivos formativos de los cursos avanzados de arquitectura de computadores suele estar el de que los estudiantes sean capaces de describir y analizar el funcionamiento de los protocolos de coherencia de cachés. Aunque dichos protocolos son relativamente sencillos, es necesario analizar muchas situaciones diferentes para entender cómo abordan todos los detalles del problema que quieren resolver. Lo que hace que sean complejos de explicar y de comprender. Una herramienta que ilustrara gráficamente el funcionamiento de dichos protocolos facilitaría enormemente su enseñanza/aprendizaje.

Con objeto de mejorar la docencia de dicha materia, hemos desarrollado tres animaciones interactivas que muestran cómo funcionan tres de los protocolos de coherencia de caché más frecuentemente utilizados. Para cada protocolo, una serie de operaciones de lectura/escritura ilustran todas las posibles situaciones que pueden darse. Las animaciones permiten avanzar y retroceder para poder entender/estudiar mejor las acciones que tienen lugar en cada paso.

Summary

Among the educational objectives in advanced courses of computers architecture there is usually one that states that students should be able to describe and analyze how the cache coherence protocols work. Although these protocols are relatively simple, it is necessary to analyze many different situations to understand how they address all the details of the problem they solve. This makes them complex to be explained and to be understood. A tool that illustrates graphically the operation of these protocols should greatly facilitate the teaching/learning

*Este trabajo ha sido parcialmente financiado por el proyecto «Activitats formatives per a assignatures de la matèria Arquitectura de Computadores» de la Unitat de Suport Educatiu de la Universitat Jaume I (10G136-16).

of these protocols.

With the aim of improving the teaching on this subject, we have developed three interactive animations that show how some of the most frequently used cache coherence protocols work. For each protocol, a sequence of read and write operations illustrates all possible situations that can take place in each protocol. The tool is interactive in that the student can go forward and backward to understand/study the different actions that occur at each step.

Palabras clave

Arquitectura de computadores, multiprocesadores, coherencia de cachés, protocolos, animación interactiva

1. Introducción

Todos los multiprocesadores actuales contienen uno o más niveles de memoria caché para mitigar la diferencia entre las velocidades del procesador y de la memoria principal. En estos sistemas, los datos pueden almacenarse/replicarse en múltiples sitios. Esto introduce un problema serio ya que, si no se tiene cuidado, diferentes procesadores podrían acabar almacenando valores distintos para los mismos datos. Para evitar que esto ocurra, se necesita un esquema (protocolo o algoritmo) que garantice la coherencia del sistema de memoria.

En los cursos avanzados de arquitectura de computadores se enseña cómo se puede abordar el problema de la coherencia de cachés utilizando protocolos basados en espionaje y en directorio. Aunque los algoritmos son relativamente sencillos, hay que analizar muchas situaciones diferentes para entender cómo lidian estos protocolos con todos los detalles del problema. Creemos que una herramienta que ilustre gráficamente el modo en el que los distintos protocolos tratan las diferentes situaciones puede

facilitar el aprendizaje de dichos protocolos.

Para explicar los protocolos de coherencia de cachés en nuestras clases de arquitectura avanzada de computadores, buscamos herramientas de simulación o animaciones que pudieran ilustrar como funcionan dichos protocolos. Queríamos que nuestros estudiantes pudieran experimentar cómo cambian los estados en cada bloque de caché cuando los diferentes procesadores acceden a diferentes posiciones de memoria, cuáles son las transiciones entre estados, y cuáles son los mensajes que se generan en cada caso. Aunque dimos con algunos simuladores sofisticados, éstos eran demasiado complejos, principalmente porque no estaban diseñados para servir como herramientas de ayuda a la docencia. También encontramos varias animaciones sencillas que capturaban algunas situaciones de coherencia de cachés pero que, desafortunadamente, eran demasiado simplistas y, por otro lado, para nada exhaustivas en cuanto a las posibles situaciones que cubrían.

El simulador desarrollado por Jeremy Jones para el protocolo MECI de coherencia de cachés [4] ha resultado ser la herramienta que más se parecía a lo que estábamos buscando. Dicho simulador representa de forma gráfica un multiprocesador con un banco de memoria y tres procesadores con sus respectivas cachés. El usuario puede indicar de forma interactiva las operaciones de lectura y escritura que debe realizar cada procesador. Cuando se realiza una operación, se muestra de forma animada tanto el trasiego de los datos como la variación de los estados de las líneas de caché implicadas.

No obstante, dicho simulador presenta desde nuestro punto de vista, una serie de inconvenientes para el aprendizaje de los protocolos de coherencia. El principal inconveniente para utilizarlo como herramienta de apoyo a la docencia es que es necesario conocer de antemano cómo funciona el protocolo MECI de coherencia de cachés. Esto es así ya que para poder observar cómo funciona el protocolo en cada uno de los casos posibles, es necesario generar una secuencia de lecturas/escrituras que provoque el caso que se quiere observar. Otro de los inconvenientes que presenta actualmente es que no es multiplataforma. El simulador está desarrollado sobre Vivio [3], que por el momento solo puede ejecutarse en Windows (aunque está prevista una versión de Vivio basada en Qt, que presumiblemente será más portable). Por último, dicho simulador tan solo cubre uno

de los tres protocolos que queremos mostrar.

Debido a todo lo anterior, decidimos desarrollar un conjunto de animaciones interactivas que pudieran utilizarse en clase para mostrar a los estudiantes el funcionamiento de los protocolos de coherencia de cachés, y que les pudieran servir para ganar un mayor grado de comprensión de los mismos. También decidimos desarrollar las animaciones en Flash para que éstas tuvieran la mayor portabilidad posible.

Las animaciones interactivas que hemos desarrollado cubren los protocolos MCI y MECI basados en espionaje, así como el protocolo MECI basado en directorio. Los casos propuestos en estas animaciones son exhaustivos, cubriendo todos las posibles situaciones que se pueden dar en cada protocolo. Con cada ejemplo se muestra una descripción completa de qué está ocurriendo, y el estudiante puede avanzar y retroceder para examinar detenidamente cómo se resuelve cada una de los casos. Además, hemos intentado que las animaciones fueran intuitivas y para que no requirieran de preparación previa para comenzar a utilizarlas.

El resto del artículo está organizado como sigue. En el Apartado 2, ofrecemos un breve visión general del problema de la coherencia de cachés. El Apartado 3 describe las animaciones desarrolladas. Finalmente, en el Apartado 4, se resumen las conclusiones y el trabajo futuro que esperamos llevar a cabo.

2. El problema de la coherencia de cachés

Los multiprocesadores, especialmente los contruidos a partir de microprocesadores de bajo costo, ofrecen una solución efectiva en coste a la siempre creciente demanda de mayor potencia de computación [1]. Sin embargo, diseñar y programar correcta y eficientemente multiprocesadores, plantea una serie de problemas complejos, siendo uno de ellos el de mantener la coherencia del sistema de memoria.

El problema de la coherencia de cachés surge cuando diferentes procesadores guardan en sus respectivas cachés el valor de la misma posición de memoria y posteriormente lo modifican [2]. Una solución inteligente, basada en el uso de un bus como red de interconexión, aborda dicho problema en multiprocesadores a pequeña escala con memoria compartida. La idea básica es la de garantizar que, antes de que se escriba en una dirección de memoria, todas

las otras copias de dicha dirección, que podrían estar en otras cachés, sean invalidadas. De esta forma, el sistema permite que existan múltiples copias de lectura de una posición de memoria, pero solo permite una copia de escritura.

El componente clave para los protocolos habituales de coherencia basados en espionaje es el bus que interconecta los procesadores y los módulos de memoria distribuida. Cuando un procesador quiere escribir en un bloque de caché, que podría estar compartido, un protocolo basado en espionaje coloca la petición en el bus. De esta forma, todas las cachés pueden ver la petición y si tienen una copia de dicho bloque de caché, simplemente la invalidan.

En resumen, las operaciones de espionaje se implementan colocando la petición en el bus y haciendo que todas las cachés lean la dirección implicada; si dicha dirección coincide con una de las guardadas en la caché del procesador, o bien realizan una invalidación, o bien proporcionan los datos de su caché. Puesto que todas las peticiones tienen que colocarse en el bus, que solo puede servir una petición a la vez, el bus serializa las escrituras posiblemente concurrentes de dos o más procesadores. Esto impone un orden en todas las escrituras (incluyendo aquellas que van a la misma dirección), lo que es crítico para mantener la coherencia.

La reducción en la complejidad de la programación que se consigue al mantener la coherencia de cachés, junto con su relativamente barata implementación, llevó a su adopción en todos los multiprocesadores de pequeña escala basados en bus. En los últimos años, se han popularizado los multiprocesadores diseñados con un pequeño número de núcleos (dos a ocho) en el mismo chip que implementan protocolos de coherencia de cachés, reduciendo aún más el coste de los multiprocesadores a pequeña escala e incrementando su popularidad.

Desafortunadamente, los esquemas de espionaje utilizados en los multiprocesadores simétricos no escalan bien, puesto que el bus se convierte rápidamente en un cuello de botella cuando crece el número de procesadores conectados a él.

La solución a este problema consistió en desarrollar un mecanismo de coherencia que pudiera extenderse eficientemente a otros diseños arquitecturales o de interconexión. La respuesta fueron los esquemas basados en directorio. Estos esquemas confían en una estructura adicional, llamada directorio,

que realiza un seguimiento de qué procesadores han guardado en caché cualquier bloque de memoria principal. Puesto que el directorio mantiene información actualizada de qué cachés tienen copia de un bloque de memoria dado, un protocolo de coherencia puede utilizar dicha información para conseguir una visión consistente del sistema de memoria. Para garantizar la coherencia, el estado de cada bloque de caché es almacenado en la caché y se guarda información adicional para cada bloque en el directorio.

Los protocolos basados en directorio no fueron adoptados inicialmente debido a que, en un pequeño multiprocesador, un bus cubre ampliamente las necesidades de comunicación y, además, los esquemas basados en espionaje son fáciles y baratos de implementar [2]. También hay que tener en cuenta que, aunque la aproximación basada en directorios evita la utilización de difusiones para interrogar a todas las cachés, un directorio único y centralizado tampoco es la solución, ya que el cuello de botella se desplazaría del bus al directorio. Estos protocolos han empezado a utilizarse ampliamente con la llegada de multiprocesadores con un gran número de procesadores que utilizan una red de interconexión distinta a un bus y en los que el directorio se ha distribuido junto con los bancos de memoria.

Los protocolos de coherencia de cachés basados en espionaje y en directorio pueden utilizar tres o cuatro estados diferentes para marcar el estado de cada bloque de caché [5]. En el protocolo MCI de coherencia de caché, cada bloque de caché puede estar en uno de los siguientes tres estados: Modificado, Compartido o Inválido. Un bloque de caché se marca como modificado cuando su contenido ha sido modificado. Cuando un bloque de caché está en dicho estado, esa caché es la única que tiene una copia válida de dicho bloque. Un bloque de caché se marca como compartido cuando la información que contiene es fruto de una operación de lectura. Cuando un bloque está en dicho estado, hay otras copias válidas de los datos que contiene (como mínimo, en memoria principal). Finalmente, un bloque de caché se marca como inválido cuando otro procesador ha realizado una petición de escritura sobre el bloque de memoria al que hace referencia dicho bloque de caché.

Por otro lado, el protocolo MECI de coherencia de cachés incorpora un nuevo estado: Exclusivo. Este estado se utiliza para indicar que la caché tiene

una copia válida y no modificada de los datos y que no hay otras cachés con copia de dichos datos. Gracias a este estado, cuando un procesador realiza una operación de escritura sobre un bloque de caché que está únicamente en su caché (en estado exclusivo), no necesita poner en el bus una petición de escritura. Esto aligera la utilización del bus, especialmente cuando se ejecutan programas donde hay pocos datos que se compartan realmente entre los distintos procesos.

Las animaciones que presentamos en el siguiente apartado tienen por objeto facilitar la comprensión de cómo funcionan los protocolos MCI y MECI de coherencia de cachés basados en espionaje y el protocolo MECI basado en directorio.

3. Animaciones interactivas de protocolos de coherencia de cachés

Para desarrollar las animaciones hemos utilizado Adobe Flash. Esta herramienta proporciona dos formas de desarrollar aplicaciones. La primera de ellas consiste en definir una línea del tiempo y dibujar las imágenes que forman la animación. Afortunadamente, no es necesario dibujar manualmente todas las imágenes, ya que Adobe Flash es capaz de crear automáticamente las imágenes requeridas entre una imagen inicial y otra final. En particular, Adobe Flash hace esto siguiendo la pista a los objetos que cambian su tamaño o posición entre estas dos imágenes y generando las imágenes intermedias. Utilizando este método se pueden obtener fácilmente animaciones sencillas.

La otra forma de definir el comportamiento de una animación en Adobe Flash es utilizando ActionScript, un lenguaje de programación orientado a objetos, que se puede emplear para acceder a los métodos y propiedades de los objetos dibujados en el área de trabajo. Utilizando ActionScript se pueden conseguir efectos más complejos y, lo que es más importante, se pueden crear animaciones con las que el usuario pueda interactuar.

Para desarrollar las animaciones hemos utilizado ambos métodos: la definición de una línea de tiempo y la programación con ActionScript. El primero de ellos principalmente para definir todos los casos de estudio y el segundo para permitir al usuario la navegación por toda la animación.

Las animaciones están disponibles en tres ficheros Flash, uno para cada uno de los siguientes protocolos de coherencia de cachés: MCI basado en espionaje, MECI basado en espionaje y MECI basado en directorio (este último utiliza un directorio implementado como un vector de bits completo). Estas animaciones pueden descargarse desde: [«http://lorca.act.uji.es/projects/ccp/»](http://lorca.act.uji.es/projects/ccp/).

En la Figura 1 puede verse la interfaz gráfica de la animación correspondiente al segundo de los protocolos: el protocolo MECI basado en espionaje. La interfaz gráfica está dividida en las siguientes partes:

1. El título de la animación.
2. Una descripción detallada de qué ocurre en el paso actual.
3. Botones «anterior», «reproducir» y «siguiente». El usuario puede utilizarlos para ir al paso anterior, para reproducir (de nuevo) el paso actual y para ir al siguiente paso, respectivamente.
4. La representación del multiprocesador utilizado como ejemplo (esta misma representación se utiliza para el protocolo MCI basado en espionaje, mientras que para el protocolo basado en directorio se utiliza otra bastante diferente).
5. Botones para cada paso. El usuario puede utilizarlos para ir directamente a un paso determinado. Esta parte también muestra cuál es el paso actual y qué pasos se han completado ya.

El multiprocesador mostrado en las animaciones de los protocolos basados en espionaje consta de tres procesadores (P1, P2 y P3) y de dos bancos de memoria (MEM1 y MEM2), todos ellos interconectados mediante un bus. Cada procesador tiene los siguientes elementos (de arriba a abajo y de izquierda a derecha):

- Un área de texto que representa el núcleo del microprocesador y que se utiliza para indicar la petición de lectura o escritura que se está realizando.
- Una tabla que representa el contenido de la caché del procesador. La caché consta de dos bloques de 4 palabras cada uno.
- Una columna que muestra el estado actual de cada bloque de caché: modificado (M), exclusivo (E), compartido (C) o inválido (I).

Cada memoria consta de los siguientes elementos:

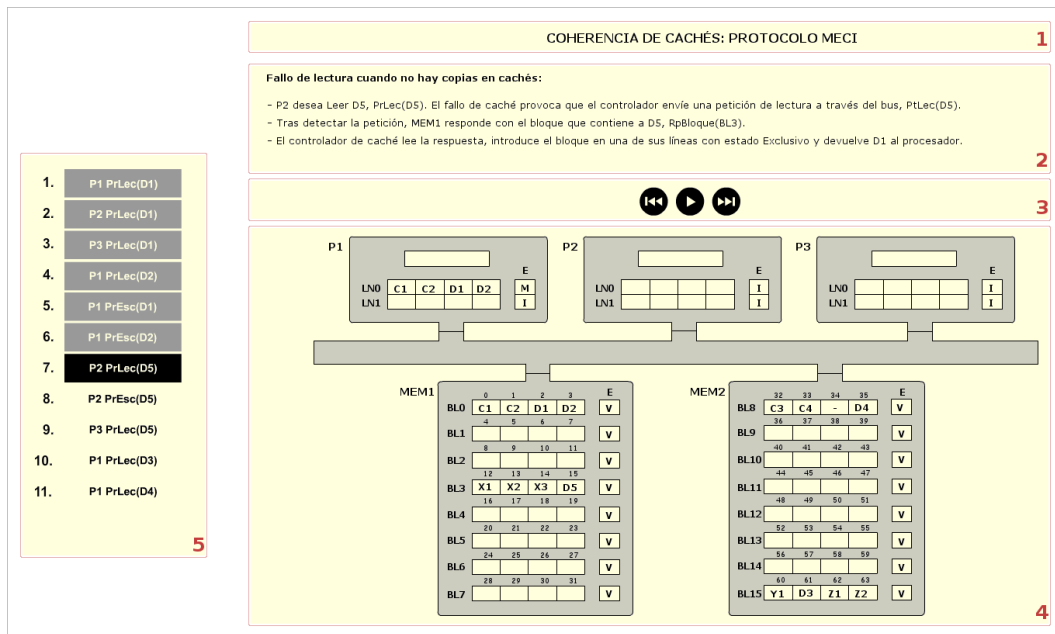


Figura 1: Interfaz gráfica de la animación correspondiente al protocolo de coherencia MECI basado en espionaje.

- Una tabla que representa el contenido de la memoria. La memoria contiene 32 palabras organizadas en 8 bloques de 4 palabras cada uno.
- Una columna que muestra el estado actual de cada bloque de memoria: válido (V) o inválido (I).

La animación del protocolo MECI basado en directorio utiliza un multiprocesador diferente (ver Figura 2). Esta arquitectura consta de cuatro nodos interconectados mediante una red. Cada nodo tiene un procesador, una caché y una memoria similares a los ya descritos. La principal diferencia entre este multiprocesador y los de las otras dos animaciones es que en éste, para cada bloque de memoria, hay un vector de bits donde se almacena la información del directorio.

El funcionamiento general de las animaciones se resume a continuación.

1. Al comenzar cada paso se muestra el estado del multiprocesador en dicho paso y una descripción detallada de qué es lo que va ocurrir. El usuario debe pulsar el botón «reproducir» para que comience la animación de dicho paso. Esto

se ha hecho así para permitir al estudiante leer qué es lo que va a ocurrir antes de que efectivamente ocurra.

2. En cuanto se pulsa el botón «reproducir», se muestra de forma animada cómo uno de los procesadores realiza una petición de lectura o de escritura y cómo esta petición provoca una consulta en su caché.
3. A continuación se puede observar cómo el controlador de caché responde al procesador con el dato solicitado. O alternativamente, cómo el controlador de caché coloca en el bus la petición requerida para conseguir dicho dato o para invalidar las otras copias de dicho dato.
4. En el caso de que se transmita la petición por el bus, se muestra cómo los controladores de caché afectados por dicha petición realizan las acciones indicadas por el protocolo de coherencia para dicha situación.
5. Finalmente, y si es el caso, se representa de forma animada cómo los datos viajan por el bus desde donde estén actualizados en ese momento hasta la caché, y de la caché al procesador que inició la operación.

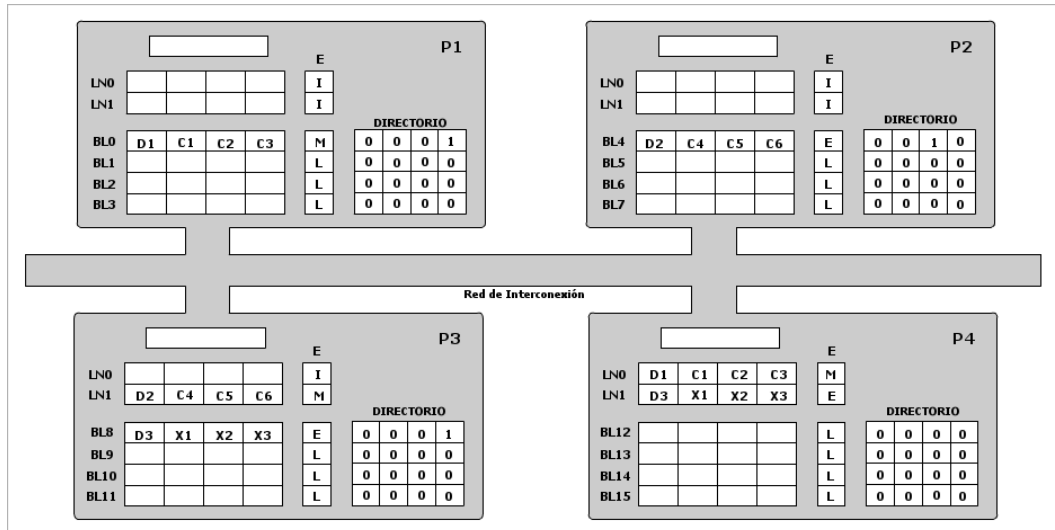


Figura 2: Multiprocesador utilizado en la animación del protocolo de coherencia MECI basado en directorio.

Añadir que mientras transcurre la animación, se resalta también de forma gráfica si los controladores de caché o los bancos de memoria actualizan el estado de alguno de sus bloques. De igual forma, en la animación del protocolo MECI basado en directorio, también se resalta gráficamente, cuando se da el caso, la modificación del directorio asociado a cada bloque de memoria.

El profesor puede explicar en clase el funcionamiento de los distintos protocolos utilizando estas animaciones como material de apoyo, en lugar de utilizar transparencias. (Las animaciones se han simplificado al máximo para que se visualicen correctamente cuando sean proyectadas en clase.) Otra posible forma de utilizarlas es que el profesor de un tiempo a los estudiantes para que estudien individualmente o en grupo el funcionamiento de los distintos protocolos. Para posteriormente hacer que sean los propios estudiantes los encargados de explicar en clase lo que esperan que ocurra en cada paso de la animación. Una vez que han descrito qué esperan que ocurra, se pone en marcha la animación para comprobar si estaban en lo cierto. En el supuesto de que los estudiantes hubieran fallado en su descripción, y puesto que la animación permite repetir cada paso, se puede incidir en el comportamiento que no había sido previsto correctamente. Ésta es la forma

en la que estamos utilizando actualmente las animaciones en el contexto del aula. Nuestra experiencia ha sido tremendamente positiva ya que ha conseguido involucrar a los estudiantes en el aprendizaje de los protocolos, a la vez que ha estimulado su participación en clase.

También se pueden proporcionar estas animaciones a los estudiantes. De esta forma, pueden utilizarlas por su cuenta para profundizar en su conocimiento sobre el comportamiento de los protocolos de coherencia de cachés. Utilizar estas animaciones de forma autónoma, preferiblemente después de haberlas visto en clase, creemos que les ha ayudado a alcanzar el objetivo formativo de ser capaces de describir y analizar el funcionamiento de dichos protocolos.

Lamentablemente, y puesto que solo tenemos un grupo de estudiantes y que los protocolos de coherencia representan una parte de la materia, no hemos sabido cómo medir numéricamente el impacto que la utilización de estas animaciones han tenido en el aprendizaje de nuestros estudiantes.

Por último, consideramos que las animaciones presentadas permiten adquirir una comprensión básica del funcionamiento de los protocolos de coherencia de cachés. Que es el objetivo en nuestro curso de arquitectura de computadores. No obstante, en el

caso de que los estudiantes debieran adquirir un conocimiento más profundo de dichos protocolos, sería conveniente complementar la utilización de estas animaciones con prácticas con un simulador como el de Jeremy Jones [4].

4. Conclusiones y trabajo futuro

Hemos desarrollado tres animaciones en Flash que abarcan todas las posibles situaciones que pueden darse en la práctica en los tres protocolos considerados de coherencia de cachés. Hemos comprobado en nuestra docencia que estas animaciones pueden ayudar a entender cómo funcionan estos protocolos de coherencia de cachés y que el profesor puede utilizarlas eficazmente en clase. Es más, consideramos que estas animaciones también pueden ser utilizadas por los estudiantes para analizar por su cuenta cómo funcionan los protocolos de coherencia de cachés.

Desarrollar las animaciones ha sido mucho más complejo de lo que esperábamos, especialmente cuando hemos necesitado introducir cambios sobrevenidos en sus líneas del tiempo. La experiencia nos ha enseñado que es sumamente conveniente planificar cuidadosamente todos los detalles antes de comenzar la codificación propiamente dicha de la animación. En particular, los cambios de última hora fueron muy costosos de implementar.

Finalmente, tenemos previsto seguir utilizando estas animaciones en nuestro curso avanzado de arquitectura de computadores y obtener realimentación de los estudiantes sobre qué mejoras piensan que podríamos realizar para conseguir una comprensión mejor de cómo funcionan los diferentes protocolos de coherencia de cachés.

Referencias

- [1] Culler, David E., Pal Singh, Jaswinder, Gupta, Anoop, *Parallel computer architecture: a hardware/software approach*, Morgan Kaufmann, 1998.
- [2] Hennessy, John L., Patterson, David A., *Computer Architecture: A Quantitative Approach*, Morgan Kaufmann, 2006.
- [3] Jones, Jeremy, *Vivio 5.1 - A Tool For Creating Interactive Reversible E-Learning Animations for the WWW*,
<http://www.scss.tcd.ie/~jones/vivio/home.htm>
- [4] Jones, Jeremy, *Vivio MESI cache coherency protocol animation*,
<http://www.scss.tcd.ie/~jones/vivio/caches/MESI.htm>
- [5] Ortega, Julio, Anguita, Mancia, Prieto, Alberto, *Arquitectura de Computadores*, Editorial Thomson, 2005.

SyCS: una base de conocimiento de soporte a la adquisición de la competencia “Sostenibilidad y Compromiso Social”

David López¹, Fermín Sánchez¹, Marc Alier²,
Jordi Garcia¹, Jose Cabré³

1.- Dpto. de Arquitectura de Computadores, 2.-Dpto. de Ingeniería de Servicios y Sistemas de Información, 3.- Dpto. de Organización de Empresas

Universitat Politècnica de Catalunya. UPC-BarcelonaTECH
Campus Nord. C/ Jordi Girona 1-3. 08034 Barcelona.

david@ac.upc.edu, fermin@ac.upc.edu, ludo@essi.upc.edu,
jordig@ac.upc.edu, jose.cabre@upc.edu

Resumen

La competencia transversal “Sostenibilidad y Compromiso Social” es comúnmente aceptada como imprescindible en el mundo actual, pero es una de las que resultan más complicadas de introducir en los planes de estudio, principalmente por el desconocimiento del profesorado. En este trabajo presentamos una base de conocimiento que reúne artículos científicos, libros, videos, compendios de datos, experiencias, etcétera, relacionadas con la sostenibilidad y las áreas de conocimiento asociadas a la Ingeniería Informática. Esta herramienta permite al profesor que debe trabajar esta competencia extraer información precisa y útil que relacione su asignatura con temas de sostenibilidad. La herramienta es de libre acceso, y está disponible en la dirección web <http://syics.fib.upc.edu>

Summary

The Skill "Sustainability and Social Commitment" is commonly accepted as essential in today's world. However it proves tricky to introduce into the curriculum, mainly because of lack of knowledgeable teachers. To address this issue we present a knowledge base that brings together scientific articles, books, videos, compilations of data, experiences, etc., related to sustainability and knowledge areas associated with computer science engineering. This is a valuable tool that should provide to the teacher accurate and useful information in the research task of finding links between his/her course and the "Sustainability and Social Commitment" skill. It is a free access tool that can be found at <http://syics.fib.upc.edu>

Palabras clave

Sostenibilidad. Compromiso social. Competencias transversales. Herramientas para el profesorado.

1. Motivación

Uno de los cambios que nos ha traído la adaptación al Espacio Europeo de Educación Superior (EEES) es la inclusión de las competencias profesionales o transversales en los planes de estudios de las universidades. Hay multitud de estudios sobre la necesidad de incorporar estas competencias [3, 5, 11].

Hay competencias cuya necesidad nadie discute, mientras que otras tienen menos consenso. También hay competencias con las que el profesorado se siente más cómodo, pues tiene alguna idea de cómo enseñarlas y evaluarlas.

La competencia transversal “Sostenibilidad y Compromiso Social” (de ahora en adelante, SyCS) es generalmente aceptada como imprescindible en el mundo actual (a pesar de tener detractores), pero despierta cierta preocupación debido al desconocimiento que de la misma tienen los docentes. Preguntas como ¿Qué es sostenibilidad?, ¿qué relación tiene con mi asignatura?, ¿qué deben aprender mis estudiantes?, o ¿cómo se evalúa la competencia? resultan comunes cuando se discute sobre esta competencia.

Existen diversos trabajos sobre de la relación general entre la informática y SyCS [4, 7], cómo introducir esta competencia en asignaturas [2, 6, 8, 10] y en un plan de estudios de ingeniería informática [9] o cómo evaluarla [2, 8, 11]. Sin

embargo, no solucionan un problema complejo: que cada profesor sepa cómo se relaciona la competencia SyCS con su asignatura.

Se puede encontrar un grupo de profesores que se interesen por el tema, que lo consideren motivador y que estén dispuestos a trabajarlo en su asignatura. Pero una vez están dispuestos, no se les puede dejar solos, sin apoyo, diciéndoles que busquen ellos mismos la relación de su área de conocimiento con SyCS y que ideen actividades partiendo de cero. Si no hay un cierto apoyo institucional, la iniciativa será un fracaso.

Por ello, en la *Universitat Politècnica de Catalunya* se ha desarrollado el proyecto STEP 2015. Este proyecto ofreció (entre otras cosas) soporte económico a un grupo de profesores de la *Facultat d'Informàtica de Barcelona* para poder desarrollar las herramientas, actividades y reflexiones necesarias para introducir la competencia SyCS en los estudios de Grado en Ingeniería Informática. Entre otras actividades, descritas en [9], se ha desarrollado una herramienta para dar apoyo a los profesores interesados: la Base de Conocimiento SyCS.

Esta herramienta reúne un grupo seleccionado de artículos de revistas científicas y de divulgación; bases de datos relacionadas con SyCS; páginas de responsabilidad social corporativa de empresas de informática; un compendio de leyes, directivas, recomendaciones y buenas prácticas; ejemplos y actividades para realizar en clase; etc., todo ello relacionando las áreas de conocimiento propias de la Ingeniería Informática con la competencia SyCS. La herramienta es de libre acceso y se puede encontrar en la dirección: <http://syics.fib.upc.edu>

Todas las entradas introducidas disponen de un pequeño comentario realizado por la persona que introdujo la información y una serie de etiquetas o palabras claves asociadas (*tags*), a cada entrada que permiten hacer búsquedas muy específicas.

Las ventajas de esta herramienta sobre un buscador genérico, como Google, son:

- La información está acotada. Con ciertas palabras clave, en Google pueden salir desde centenares a millones de respuestas, la mayoría de las cuales no son de interés para la persona que busca. En nuestro caso, todas las entradas han sido preseleccionadas, de manera que el número de resultados será mucho más

reducido, y la probabilidad de encontrar la información que se está buscando es mayor.

- La información introducida incluye comentarios. Estos pueden ser un par de líneas con un comentario básico, el resumen del artículo o informaciones más elaboradas. Los comentarios dan una primera visión del contenido de la entrada sin necesidad de acceder a la misma.
- La información está etiquetada, de manera que la persona que introdujo la información leyó (aunque fuera someramente) el contenido de la entrada, y decidió que la entrada debía indexarse bajo ciertas etiquetas, lo que permite hacer una búsqueda bastante eficiente y precisa (aunque el usuario también puede buscar por palabras fuera de la lista de etiquetas, que se buscan entre el título y el comentario de la entrada).

Con esta herramienta, el profesor puede buscar elementos que relacionen su asignatura con la competencia SyCS, de manera que puede adaptar actividades que encuentre, buscar ejemplos específicos o simplemente información real para diseñar sus propias actividades y ejercicios, todo ello de manera rápida y sencilla.

El resto del artículo se organiza de la siguiente manera: la Sección 2 explica el proceso de discusión que llevó a la organización de la base de conocimiento y el sistema de etiquetado. La Sección 3 detalla las especificaciones técnicas de la herramienta. La Sección 4 presenta un ejemplo de utilización. Finalmente, la Sección 5 presenta unas reflexiones finales y las conclusiones del trabajo.

2. El proceso de creación de la Base de Conocimiento.

Para crear la base de conocimiento fue preciso pensar inicialmente en la estructura y funcionalidades de la misma. Queríamos una base de conocimiento que permitiese localizar entradas a partir de palabras clave pero de forma muy libre (con casi cualquier palabra que se pudiese imaginar), y que por otra parte la asociación de palabras clave a las entradas fuese fácil y cómoda de realizar, además de fácil de actualizar. Necesitábamos, por lo tanto, que un conjunto reducido de palabras clave pudiese asociarse

como etiquetas a las entradas, pero que las búsquedas pudiesen realizarse a partir de un conjunto de palabras clave mucho mayor. También queríamos permitir la realización de búsquedas complejas, lo que implicaba disponer de una cierta estructura en las palabras clave.

Después de analizar diferentes posibilidades, decidimos dotar a la base de conocimiento de una estructura de tres dimensiones.

Por un lado tendríamos una estructura de palabras clave de dos dimensiones: un nivel principal y un nivel secundario. Establecimos un conjunto mínimo pero completo de palabras clave en el nivel principal que cubre todos los aspectos relacionados con SyCS. Estas palabras tienen un gran nivel de abstracción.

El nivel secundario está formado por un conjunto reducido pero completo de etiquetas, con conceptos mucho más específicos, de forma que se establece una relación entre las palabras claves del nivel principal y las del nivel secundario. Una misma palabra del nivel secundario puede estar relacionada con una o más palabras clave del nivel principal. Esta multisignación es la que define la posibilidad de realizar búsquedas complejas.

Las palabras clave que se definieron en el nivel principal fueron las siguientes:

1. Educación
2. Tecnociencia
3. Medioambiente
4. Modelos
5. Casos de estudio
6. Ciclo de Vida-Productos
7. Servicios
8. Valores
9. Aspectos Sociales
10. Aspectos Económicos
11. Estrategias
12. Tácticas

La Figura 1 presenta la relación de las palabras clave secundarias con las del nivel principal. La primera columna presenta el listado de las palabras clave del nivel secundario. El resto de columnas indican, mediante una X, con qué palabras del nivel principal está relacionada cada etiqueta del nivel secundario. Como puede comprobarse, el nivel principal está formado por palabras con un nivel de abstracción mayor que las del nivel secundario.

La selección de las palabras clave, tanto del nivel principal como del nivel secundario, nos llevó cuatro meses, durante los cuales nos

TAGS secundarios	1	2	3	4	5	6	7	8	9	10	11	12
Avances tecnológicos	X	X	X		X	X			X	X		
Formación	X	X		X	X				X	X		
Asignaturas	X	X	X	X	X				X			
Innovación	X	X			X			X	X	X		X
Método	X	X		X	X							X
Instituciones	X	X					X	X	X	X	X	
Cambio cultural	X			X	X				X	X	X	X
Atmósfera			X		X	X			X	X	X	X
Aqua			X		X	X			X	X	X	X
Aire			X		X	X			X	X	X	X
Biodiversidad			X		X	X			X	X	X	X
Combustibles			X		X	X			X	X	X	X
Energía	X	X	X	X	X	X			X	X	X	X
Tierra			X		X	X			X	X		
Minerales			X		X	X			X	X		
Cierre de ciclos	X	X			X	X	X		X	X	X	X
Reutilización	X		X		X	X	X		X	X	X	X
Consumo	X		X	X	X	X	X	X	X	X	X	X
Distribución	X	X		X	X	X	X		X	X	X	X
Producción industrial	X	X		X	X	X	X		X	X	X	X
Residuos	X	X	X	X	X	X	X		X	X	X	X
Cambio climático	X	X		X					X	X	X	X
Catástrofes naturales			X		X				X			X
Acidificación			X		X				X			X
Paisaje			X		X				X	X		X
Efecto invernadero	X	X		X					X	X		X
Sustancias químicas		X	X		X	X			X		X	X
Contaminación		X	X	X	X	X			X	X		X
Globalización	X	X	X	X	X	X			X	X	X	X
Calidad de vida	X			X					X	X	X	X
Recursos			X	X	X	X			X	X	X	X
Consumidor	X		X		X				X	X	X	X
Mercadotecnia	X		X		X				X	X	X	X
Empresa			X	X	X	X			X	X	X	X
Desarrollo sostenible	X	X	X	X	X	X	X		X	X	X	X
Retorno de inversión			X	X	X	X			X	X		X
Brecha digital	X	X			X				X	X	X	X
Justicia	X				X				X	X	X	X
Empleo					X				X	X	X	X
Ergonomía	X	X			X				X			X
Demografía				X	X				X	X	X	X
Salud	X	X	X	X	X				X	X	X	X
Urbanismo	X	X	X		X				X	X	X	X
Sociedad	X				X				X	X	X	X
Paz	X				X				X	X		X
Solidaridad	X				X				X	X		X
Desequilibrio	X	X			X				X	X		X
Política					X				X	X	X	X
Turismo	X		X		X				X	X	X	X
Derechos humanos	X		X		X				X	X	X	X

Figura 1. Relación entre las palabras clave del nivel principal y del secundario

Entre reunión y reunión clasificábamos webs, artículos y otras entradas a partir de las palabras clave que ya habíamos seleccionado, e incorporábamos provisionalmente palabras nuevas si lo considerábamos necesario. La incorporación definitiva se realizaba en nuestra reunión semanal

y siempre de común acuerdo. Una vez estuvieron escogidas todas las palabras clave, establecimos la relación entre el nivel principal y el secundario. Para ello, cada uno de los autores hizo una asignación que luego revisamos de forma conjunta. Para aquellas asignaciones en que no nos pusimos de acuerdo hicimos una nueva asignación de forma individual, después de discutir cada palabra clave, y marcamos solamente aquellas X en las que había acuerdo por mayoría.

Por otra parte, y en paralelo con el proceso anteriormente descrito, definimos la tercera dimensión de la tabla. Para permitir la búsqueda de entradas a partir de un gran número de palabras clave, establecimos para cada palabra clave del nivel secundario un conjunto de sinónimos o palabras relacionadas, por las cuales también era posible buscar en la base de conocimiento. Por ejemplo, para la palabra clave “biodiversidad” definimos como palabras relacionadas las siguientes: “Fauna, Flora, Especies, Biosfera, Naturaleza, Extinción, Animales, Plantas, Bosque, Árboles, Veda, Caza y Pesca”. De este modo, una entrada puede estar etiquetada como biodiversidad pero aparecerá también si se busca por cualquiera de las palabras relacionadas.

La existencia de las “palabras relacionadas” nos permite actualizar fácilmente la base de conocimiento y añadir nuevas palabras clave sin necesidad de revisar todas las entradas. Basta añadirlas como palabras relacionadas con una palabra clave ya existente. Esta tercera dimensión ha favorecido, además, la capacidad multilingüe de la base de conocimiento, ya que como palabras relacionadas se pueden poner también las traducciones a otros idiomas de todas las palabras de la lista. De forma ortogonal, se definió una lista de palabras relacionadas para cada palabra clave del nivel principal.

El proceso completo de creación de la estructura de palabras clave, desde las primeras discusiones sobre su estructura hasta la definición final de las palabras en cada nivel y de las palabras relacionadas, duró algo más de seis meses. En este proceso, además de los firmantes, intervinieron dos becarios que se encargaron fundamentalmente de mantener actualizada la lista y estructura de palabras clave y de catalogar las entradas a partir de las palabras clave ya definidas. Uno de los becarios tenía amplia formación en temas relacionados con SyCS (lo seleccionamos

justo después que leyese su tesis, relacionada con este tema), mientras que el otro era un estudiante de ingeniería informática. En la implementación de la base de conocimiento ha intervenido también un becario con la titulación de ingeniero informático.

3. Especificaciones de la Base de Conocimiento SyCS.

Para implementar la base de conocimiento SyCS definimos un conjunto de requisitos:

- Que fuera un sistema online basado en Web para maximizar la facilidad de acceso.
- Que implementara una base de datos documental con capacidad de enlazar a distintos tipos de contenidos.
- Que los objetos/documentos pudieran ser etiquetados (*tags*) según la taxonomía creada por el nuestro equipo de trabajo (descrito en la Sección 2).
- Que tuviera control de acceso con distintos niveles de concreción.
- Que guardara los registros de acceso para permitir realizar estadísticas de uso.
- Que fuera un entorno muy fácil de aprender y, a ser posible, familiar para el colectivo del profesorado de nuestro centro.

Pronto se descartó usar bases de datos relacionales o realizar un desarrollo a medida, dado que el tiempo, esfuerzo y coste de desarrollo y el riesgo asociado serían demasiado altos. Se barajó la opción de usar un sistema Wiki como el software Mediawiki (<http://mediawiki.org>) en el que se basa Wikipedia. Los Wikis son sistemas documentales muy potentes que permiten implementar casi cualquier tipo de estructura y navegación, pero requieren una atención continua o se corre el riesgo que la estructura de la información degenere. La complejidad del código fuente y la dificultad de adaptación de Mediawiki (y otros motores wiki considerados) para los propósitos del proyecto nos hicieron descartar esta opción.

Una alternativa que tomó mucha fuerza fueron los sistemas de Bookmarking (marcadores) Social como <http://Delicious.com> o <http://Diigo.com>. Estos sistemas permiten la creación de colecciones de referencias etiquetadas según taxonomías adaptables (se definen orgánicamente)

y realizar búsquedas complejas, lo cual cumplía con muchos de los requisitos que nos habíamos impuesto. Los sistemas de Bookmarking Social añadían una funcionalidad adicional: cada usuario puede crear su propia lista de referencias, etiquetar cada entrada según su propio criterio. Las referencias se pueden compartir con el resto de usuarios y ello aporta una cualidad adicional (un sabor 2.0, por así decirlo) a la base de conocimiento.

Como elemento negativo, los sistemas de Bookmarking Social basados en servicios 2.0 suponen el ceder el control de la base de conocimiento a una empresa externa, que proporciona un servicio gratuito pero que no se compromete a proporcionar una calidad de servicio ni se responsabiliza de la posible pérdida de información. Por tanto, buscamos un sistema similar que pudiéramos implementar nosotros y posteriormente alojar en un servidor de la *Facultat d'Informàtica de Barcelona*.

El año 2008 miembros de nuestro equipo desarrollaron un módulo de Moodle [1] que implementa las funcionalidades de un Bookmarking Social. Moodle (<http://Moodle.org>) es el software libre en el que se basa el sistema de gestión del aprendizaje de nuestra universidad y de la inmensa mayoría de las universidades españolas. Es un sistema bien conocido por los profesores que permite un acceso granular, contextualizado y registros de acceso.

Se adaptó el diseño de dicho módulo a requisitos específicos de la taxonomía creada por nuestro equipo. El sistema implementado se basa en Moodle 1.9.10, con una versión modificada del módulo <http://code.google.com/p/moodle-social-bookmarking/>

4. Ejemplo de utilización

La Base de Conocimiento descrita en este trabajo es accesible a todos los usuarios que lo deseen en la dirección <http://sysc.fib.upc.edu>, y su uso es muy sencillo. Tras una pantalla inicial de presentación de la Base de conocimiento, el usuario es direccionado directamente a la pantalla de búsqueda.

En la Figura 2 se puede ver una captura de la pantalla principal, la de búsqueda, con un ejemplo. Esta pantalla contiene, en el centro, el cuadro de texto del formulario para realizar la

búsqueda, y en la derecha dos nubes informativas. La nube superior indica las categorías de los elementos disponibles en la base de conocimiento: legislación, artículos, bases de datos, tesis doctorales, etcétera. La nube inferior indica las etiquetas predefinidas.

Una búsqueda se puede realizar de dos formas diferentes:

- Mediante una de las categorías de los elementos disponibles en la base de conocimiento o una de las etiquetas predefinidas de las nubes, con las que se han clasificado todas las entradas, o bien
- Mediante texto libre, que será buscado entre el título y la descripción de la entrada.

Para efectuar una búsqueda mediante una etiqueta predefinida, se puede seleccionar la etiqueta de una de las nubes de la derecha, o bien se puede escribir en el cuadro de texto el nombre de la etiqueta precedida de la marca "tag:". Por ejemplo, para realizar una búsqueda de las entradas catalogadas con la etiqueta *solidaridad*, se puede escribir directamente: "tag:solidaridad". En caso de querer realizar una búsqueda mediante texto libre, se puede escribir directamente el texto en el cuadro. Por ejemplo, escribiendo *ONU* se mostrarán todas las entradas que contengan esta palabra en el título o en la descripción.

En la Figura 2 se puede ver el resultado de la búsqueda "tag:sistemas operativos". Se puede observar un listado con las entradas seleccionadas que para cada entrada contiene el título, la descripción, y la lista de etiquetas que clasifican esa entrada. El título, a su vez, es un enlace hacia el recurso, sea de tipo página web, artículo, vídeo, etc. También se pueden realizar búsquedas más restrictivas combinando etiquetas y/o texto libre, de manera que la búsqueda se realizará entre las entradas que cumplan todos los requisitos. Para hacer este tipo de búsqueda se debe separar cada parámetro por una coma. Por ejemplo, la búsqueda: "tag:solidaridad,ONU" (o también "ONU,tag:solidaridad") mostrará las entradas que estén etiquetadas con *solidaridad* y que contienen la palabra *ONU* en el título o en la descripción. Otro ejemplo: la búsqueda "tag:solidaridad,tag:innovación" mostrará las entradas que estén etiquetadas con *solidaridad* y con *innovación* simultáneamente. De esta manera, se puede refinar la búsqueda tanto como se desee.

SyCS Database DB Search Base de Datos Jump to...

Busca primero la información más genérica en la nube de tags de la derecha. Luego puedes refinar la búsqueda en la línea de búsqueda. Clica el interrogante al lado del botón "Search" para ayuda a la hora de hacer búsquedas.

My Items Featured Items Search
 Search

Searching for: tag sistemas operativos

Amazon Elastic Compute Cloud (Amazon EC2)
 Amazon Elastic Compute Cloud (Amazon EC2) es un servicio web que proporciona capacidad informática con tamaño modificable en la nube. Se ha diseñado con el fin de que la informática web resulte más sencilla a los desarrolladores. La sencilla interfaz de servicios web de Amazon EC2 permite obtener y configurar capacidad con una fricción mínima. Proporciona un control completo sobre sus recursos informáticos y permite ejecutarse en el entorno informático acreditado de Amazon. Amazon EC2 reduce el tiempo necesario para obtener e iniciar nuevas instancias de servidor a cuestión de minutos, lo que permite escalar con rapidez su capacidad de procesamiento y almacenamiento de datos para satisfacer la demanda de aplicaciones. Amazon EC2 proporciona a los desarrolladores las herramientas necesarias para crear aplicaciones resistentes a errores y para aislarse de los casos de error más comunes.
 Category: catálogos y recursos
 Tags: consumidor desarrollo sostenible empresa arquitectura de computadores sistemas operativos economía y empresa
 Saved by 1(1) person ... Seen 0 times

An Analysis of Power Consumption in a Smartphone
 Mobile consumer-electronics devices, especially phones, are powered from batteries which are limited in size and therefore capacity. This implies that managing energy well is paramount in such devices. Good energy management requires a good understanding of where and how the energy is used. To this end we present a detailed analysis of the power consumption of a recent mobile phone, the Openmoko Neo Freerunner. We report on the power consumption by the device's main hardware components. We present this power breakdown for micro-benchmarks as well as for a number of realistic usage scenarios. These results are validated by overall power measurements of two other devices: the HTC Dream and Google Nexus One. We develop a power model of the Freerunner device and analyse the energy usage and battery lifetime under a number of usage patterns. We discuss the significance of the power drawn by various components, and identify the most promising areas to focus on for further improvements of power management. We also analyse the energy impact of dynamic voltage and frequency scaling of the device's application processor.
 Category: artículos
 Tags: internet energía consumidor arquitectura de computadores dispositivos y periféricos sistemas operativos redes
 Saved by 1(1) person ... Seen 7 times

Cyrix XenServer
 Citrix Systems was born from the idea of unlocking applications from datacenters and employees from the office – creating new ways for people and IT to work. Today, this is the promise of virtual computing, and Citrix is at the epicenter.
 Category: catálogos y recursos
 Tags: innovación avances tecnológicos sistemas operativos
 Saved by 1(1) person ... Seen 0 times

EyeOS
 EyeOS es un escritorio web disruptivo al que se accede desde un navegador web. Incluye una suite de informática y aplicaciones colaborativas, y a la vez es un completo framework para desarrollar nuevas aplicaciones similares a las de escritorio clásicas. Es libre y Open Source.
 Category: catálogos y recursos
 Tags: innovación avances tecnológicos sistemas operativos
 Saved by 1(1) person ... Seen 0 times

Installing Programs Without Licensing

Category
 Declaraciones Congresos y Conferencias Audiovisuales Cursos
Catálogos y Recursos
 Legislación Revistas Entidades doctorales Manuales Tests Artículos

Tags
 asignaturas innovación empresa salud calidad de vida contaminación economía y empresa justicia método derechos humanos formación sociedad ^{bebida digital} competencias transversales sistemas operativos cambio climático arquitectura de computadores agua desarrollo sostenible avances tecnológicos recursos cambio cultural ^{urbanismo} política energía ^{consumo} instituciones ^{producción industrial}

Figura 2: Consulta de ejemplo

Habitualmente, al acceder a la herramienta se accede con perfil de usuario invitado. Este perfil es el asignado por defecto, no requiere identificación alguna y autoriza al usuario a realizar cualquier tipo de búsqueda.

Existe también el perfil de usuario editor, el cual requiere identificación, que además puede añadir nuevas entradas o modificar las existentes. Para poder tener un perfil de editor hay que solicitarlo, y sólo se requiere ser profesor universitario del área de conocimiento de las TIC o bien acreditar un interés y perfil profesional relacionado con el proyecto. La autorización la gestiona, de momento, el equipo del proyecto.

Por supuesto, tras el lanzamiento inicial la herramienta está en constante evolución. Los esfuerzos actuales se están centrando en:

- Mejorar la interfaz de búsqueda, permitiendo la realización de búsquedas más específicas y elaboradas mediante combinaciones algebraicas de las etiquetas y el texto libre (por ejemplo, listar las entradas con una etiqueta u otra, pero en cualquier caso, con cierto texto libre) y proporcionando un uso más gráfico, menos orientado a texto. Esto implica que el ejemplo de uso aquí presentado puede haber variado cuando el lector acceda a la aplicación.
- Mejorar el diseño gráfico.
- Realizar un análisis pormenorizado de su uso. Para ello es necesario que la herramienta sea utilizada por un buen número de usuarios, por lo que estamos publicitándola.

5. Reflexiones finales y conclusiones

Introducir las competencias transversales en las titulaciones de Grado no es una tarea fácil. Podemos encontrar profesores que nieguen su importancia, otros interesados pero que no sepan por donde empezar, o profesores con algún conocimiento de la competencia, pero insuficientemente formados en la misma.

En todos estos casos, lo más importante es la información. Disponer de artículos, libros, webs y otras fuentes de información que expliquen la importancia de cada competencia en los estudios de Ingeniería Informática, con buenas prácticas y ejemplos es fundamental para poder incorporar en las asignaturas correctamente el desarrollo y evaluación de la competencia.

El caso de la competencia “Sostenibilidad y Compromiso Social” (SyCS) es especialmente complejo, ya que muchos profesores tienen experiencia en otras competencias como “trabajo en equipo” o “comunicación”, pero hay un gran desconocimiento en lo que respecta a SyCS.

En este trabajo hemos presentado una base de conocimiento orientada a ayudar a los profesores de Ingeniería Informática a relacionar su asignatura con temas de SyCS.

Disponer de una herramienta de este estilo es muy importante para tener éxito en la incorporación de esta competencia a una asignatura concreta. La herramienta presentada es de acceso universal, de manera que todos los profesores interesados pueden consultarla. Además, profesores especialmente interesados podrán solicitar acceso como editores para ayudar a ampliar sus contenidos (el acceso de escritura no es universal para evitar la degradación de la información).

Queremos añadir que la herramienta es orgánica, en plena evolución. Esto significa que cuando el lector acceda a la misma, es muy posible que se hayan producido cambios en la interfaz de búsqueda, o entre las relaciones descritas en la Tabla 1, o en otros elementos. Sin embargo, el funcionamiento básico seguirá siendo como el descrito aquí.

Finalmente, queremos añadir que algunos colegas que han visto la herramienta y el proceso de generación de las etiquetas nos han solicitado usar (y modificar) la herramienta para implementar esta competencia en otras áreas de conocimiento (como Edificación y Urbanismo, por ejemplo). Igualmente, los conceptos utilizados y la propia herramienta pueden usarse para trabajar otras competencias transversales.

Agradecimientos

Queremos agradecer a Isaac Muro, Marcos Estévez, Martha Velasco y Jordi Piguillem (nuestros becarios de soporte) por habernos soportado en nuestras largas discusiones.

Este trabajo ha sido realizado con el apoyo del proyecto STEP 2015, del Centre de Cooperació per al Desenvolupament (CCD), de los Departamentos de *Arquitectura de Computadors* (DAC), *Enginyeria de Serveis i Sistemes d'Informació* (ESSI) y *Organització d'Empreses*

(OE) , así como de la *Facultat d'Informàtica de Barcelona (FIB)*, todos ellos de la *UPC-BarcelonaTECH*.

Referencias

- [1] Alier, M., Casany, M.J., Piguillem, J., Lapuente, R. "Creating wiki communities in blended learning and the creation of the Moodle new wiki". *Int'l J. of Social and Humanistic Computing* 1(3):300-313. 2010.
- [2] Coyle, E.J., Jamieson, L.H. and Oakes, W.C. "EPICS: Engineering Projects in Community Service". *Int'l J. of Engin. Educ* 21(1) 2005, pp. 139-150.
- [3] Felder, R.M. "Sermons for grumpy campers". *Chemical Engineering Education*, 41(3):183 - 184, 2007.
- [4] Franquesa, F.; Cruz, J. Ll; Álvarez, C.; Sánchez, F.; Fernández, A. y López, D. "Cómo formar Ingenieros en Informática en la competencia Sostenibilidad y Compromiso Social". *JENUI* 2009, pp 271-278. Barcelona, julio de 2009
- [5] García García, M.J.; Terrón López, M.J. y Blanco Archilla, Y. "Desarrollo de Recursos Docentes para la Evaluación de Competencias Genéricas". *ReVisión* 3(2):17-36, Diciembre 2010
- [6] Gößling-Reisemann, S. and Von Gleich, "A. Training Engineers for Sustainability at the University of Bremen". *Int'l J. of Engin. Educ.* 23(2) 2007, pp. 301-308.
- [7] Huntzinger, D.N., Hutchins, M.J. Gierke J.S. and Sutherland, J.W. "Enabling sustainable thinking in undergraduate engineering education." *Int'l J. of Engin. Educ.* 23(2) 2007, pp. 218-230.
- [8] McLaughlan, R.G. "Instructional Strategies to Educate for Sustainability in Technology Assessment". *Int'l J. of Engin. Educ.* 23(2) 2007, pp. 201-208.
- [9] Sánchez, F.; López, D. y Garcia, J. "El desarrollo de la competencia Sostenibilidad y Compromiso Social en la Facultat d'Informàtica de Barcelona". *JENUI* 2010, pp 249-256. Santiago de Compostela, Julio 2010.
- [10] Tam, E. "Developing a Sustainability Course for Graduate Engineering Students and Professionals". *Int'l J. of Engin. Educ.* 23(6) 2007, pp. 1133-1140.
- [11] Villa, A.; Poblete, M. (Dir.). "Aprendizaje basado en competencias. Una propuesta para la evaluación de las competencias genéricas". Bilbao: Mensajero/ICE Universidad de Deusto. 2007.

BlueState: un entorno para el aprendizaje de máquinas de estados de UML

Alfredo Ortigosa Aguilera Carlos Rossi Jiménez

Departamento de Lenguajes y Ciencias de la Computación

Universidad de Málaga

ETSI Informática. Campus de Teatinos

29071 Málaga

alfredo.ortigosa@gmail.com, rossi@uma.es

Resumen

El modelo de máquinas de estados es el más complejo de los que integran el lenguaje UML, base actual de la docencia en la mayoría de las asignaturas de Ingeniería del Software. La complejidad (y la consiguiente dificultad de aprendizaje) del modelo viene dada tanto por el elevado número de componentes que lo integran como por el nivel de abstracción propio del modelo.

Para solventar estas dificultades se ha desarrollado BlueState, una herramienta que permite al alumno ejecutar una simulación de la ejecución de cualquier máquina de estados, acompañada de un seguimiento gráfico. BlueState también aporta un módulo de generación de código, que permite al alumno aprender a implementar el comportamiento de un sistema empleando una metodología dirigida por modelos.

Summary

The state machine model is the most complex in the UML language, that is the current basis for teaching most of the Software Engineering subjects. The complexity (and subsequent learning difficulty) of the model is due both to the elevated number of components that integrate it, and to the abstraction level of the model.

To solve these difficulties, we have developed BlueState, a tool that allows the student to simulate and visually monitor the execution of any state machine. BlueState also includes a code generation module that lets the student learn the implementation of the behaviour of a system using a model-driven methodology.

Palabras clave

UML, máquinas de estados, simulación, generación de código, MDE.

1. Motivación

Es un hecho bien conocido que la práctica totalidad de las asignaturas que, en el ámbito de la Ingeniería del Software, incluyen la docencia de técnicas de análisis y diseño toman UML [3] como lenguaje de referencia. Este protagonismo es natural teniendo en cuenta que UML es también el estándar de facto en el desarrollo de software a nivel profesional.

UML es un lenguaje formado por un conjunto de técnicas de modelado que tienen como objetivo ofrecer diversas vistas del software. Así, por ejemplo, los casos de uso constituyen un modelo funcional, los diagramas de clases representan una vista estructural del software, y los diagramas de secuencia y las máquinas de estados permiten especificar el comportamiento dinámico.

Nuestra experiencia impartiendo esta materia durante varios cursos nos dice que son precisamente estos últimos, los modelos de comportamiento, los de más difícil asimilación por parte de los estudiantes. Los modelos funcionales, al estar basados en un concepto sencillo como es el de caso de uso no tienen una dificultad alta. Los modelos de estructura tienen como eje los conceptos de clase, atributo, método, generalización, etc. y por tanto resultan muy familiares a los alumnos que ya han aprendido técnicas de programación orientada a objetos en las asignaturas de los primeros cursos. En cambio, los modelos de comportamiento, y en particular las máquinas de estados, resultan más complicados para los alumnos.

En nuestra opinión, el motivo de la dificultad de asimilación es doble: por una parte, las máquinas de

estados es el modelo de UML con la sintaxis más compleja, en cuanto a número de elementos que puede incluir y a variedad de comportamientos que permite representar. Por otra parte, y en cierto modo como consecuencia de lo anterior, la semántica del modelo también resulta extensa y complicada y, lo que es más importante, su traducción a código no es inmediata. Es justo este último hecho el que le dificulta al alumno la comprensión, ya que le cuesta "materializar" el modelo abstracto que se le está enseñando.

El objetivo fundamental de este trabajo es precisamente ayudar al alumno en el aprendizaje de las máquinas de estados de UML. Para ello, se ha desarrollado BlueState, un entorno que permite al alumno, a partir de una máquina de estados dada o diseñada por él mismo, simular el comportamiento representado en el modelo y hacer un seguimiento gráfico en tiempo real. Además, BlueState incorpora un completo generador de código, de modo que el alumno también puede comprobar cuáles son las estructuras de implementación asociadas al modelo que ha construido. Por tanto, BlueState ayuda en la transición del nivel alto de abstracción del modelo a su concreción.

Es importante destacar que la arquitectura de BlueState se basa en el metamodelo de clases de UML. Esto hace que, como efecto secundario, nuestro entorno permita al alumno reforzar el aprendizaje de la sintaxis y semántica de las máquinas de estados, así como asimilar conceptos propios de *Model-Driven Engineering*, una de las tendencias más actuales de la Ingeniería del Software.

En la siguiente sección se aporta un estudio comparativo que demuestra que BlueState supone una mejora real sobre las soluciones existentes actualmente. En el apartado 3 se describe BlueState, así como su contexto de uso. Finalmente, en la sección 4 se aporta un análisis de nuestra experiencia, así como algunas conclusiones de este trabajo.

2. Estado del arte

Dentro de este apartado, en primer lugar resumimos las principales técnicas de implementación existentes para diagramas de estados, entre las que destacamos el método del metamodelo de clases, empleado en el modelo de BlueState. Este resumen servirá también para ilustrar los fundamentos teóricos de es-

te trabajo.

Más adelante realizamos un análisis de diversas herramientas existentes para la implementación de diagramas de estados, evaluando características deseables para el aprendizaje de estos diagramas. Entre ellas destacamos las posibilidades de simulación, seguimiento de ejecuciones, comprensión del código generado, ajuste a la especificación o facilidad de uso.

2.1. Técnicas de implementación para diagramas de estados

Como técnicas más tradicionales para la implementación de diagramas de estados básicos, pueden considerarse las tablas de transiciones [4], las sentencias de condición [7] y el patrón de diseño estado [8]. En [1] puede consultarse un estudio de técnicas de implementación no específicas para UML.

Diversos trabajos [16, 17, 19, 20, 23, 24, 27, 28] plantean modelos ampliados o alternativos para contemplar más características de un diagrama de estados y acercarse más a la especificación UML.

Muchos de estos modelos emplean estructuras poco intuitivas y sobre todo no siguen el enfoque orientado a objetos de UML, lo que dificulta su comprensión y ampliación. Sin embargo, la técnica del metamodelo de clases [6, 11, 15, 18, 25], muy afín al paradigma de la programación orientada a objetos, solventa estos inconvenientes haciendo de la implementación de un diagrama de estados de UML un proceso natural y acorde a la especificación UML; más aún teniendo en cuenta que esta especificación usa un metamodelo de clases para definir su diagrama de estados.

Empleando este método de implementación se facilita que el alumno afiance la comprensión del diagrama de estados de UML, ya que ve representada su estructura y comportamiento de una manera directa, ajustada fielmente a su especificación formal y en un lenguaje de programación de alto nivel.

2.2. Herramientas de generación, simulación y seguimiento de diagramas de estados

En la actualidad cada vez es mayor el número de soluciones para la generación de código para diagramas de estados, ya sea integradas en una herramienta de modelado o bien como una herramienta dedicada

específicamente a esa funcionalidad. Sin embargo, las soluciones existentes no presentan un alto nivel de madurez y poseen en general diversos inconvenientes, tal como analizamos en este apartado.

Más concretamente, las herramientas que vayan a ser usadas en entornos académicos para complementar el estudio de los diagramas de estados, su implementación y ejecución, deberían satisfacer las siguientes características:

- Establecer una correspondencia formal con la especificación UML para no distorsionar los conceptos estudiados.
- Indicar al alumno posibles errores de construcción del diagrama evaluando las restricciones de la especificación UML.
- Ser independiente de una herramienta de modelado en concreto, para que pueda emplearse en diferentes contextos.
- Permitir la generación de código en los lenguajes de alto nivel más extendidos y estudiados.
- Respetar los conceptos de programación orientada a objetos al implementar los comportamientos de las clases de un sistema.
- Diferenciar de forma clara el código implementado propio del diagrama de estados, del código que deba añadirse para completar los procesos del sistema desarrollado.
- Poseer un manejo sencillo y facilitar la realización de proyectos de ejemplo en pocos pasos.
- Incluir mecanismos para que pueda realizarse una simulación del diagrama de estados diseñado y así poder realizar ejemplos de sistemas diversos simulando la recepción de eventos externos.
- Mostrar de manera gráfica y en tiempo real, y en el mismo diagrama de estados, las transiciones que se están llevando a cabo para que el alumno tenga una rápida y completa perspectiva de cómo evoluciona el sistema desarrollado.

Partiendo de estas características y añadiendo otros puntos de interés, hemos realizado un completo análisis de las principales soluciones existentes para la generación y ejecución de diagramas de estados de UML. El resumen de este análisis se muestra en la tabla comparativa de la figura 1.

Como se aprecia en el análisis, el generador BlueState desarrollado en este trabajo es el que obtiene una valoración global más alta, quedando el

resto de soluciones con puntuaciones significativamente más bajas.

En el ámbito de la simulación y seguimiento, la nuestra es la solución que más funcionalidad ofrece, seguida de la solución de UniMod [5] que posee una depuración visual pero no integra un módulo de simulación.

BlueState destaca asimismo por una alta correspondencia con la especificación UML, una gran independencia de la herramienta de modelado empleada gracias a un completo analizador XMI, y sobre todo por una gran facilidad de uso y de integración de código.

Otras ventajas de BlueState, importantes desde el punto de vista técnico, son una gestión de eventos avanzada, ejecuciones siguiendo el concepto de paso run-to-completion, mecanismos de sincronización en la inicialización y finalización de máquinas de estados, y diversas posibilidades de interacción entre máquinas de estados ejecutándose en paralelo. Además, se incluye una herramienta de medición de rendimiento.

3. Descripción y contexto de uso

En este apartado describimos los elementos y consideraciones básicas en la construcción de BlueState. Además, presentamos los módulos principales de nuestro entorno de aprendizaje de diagramas de estados, enmarcados en su contexto de uso.

3.1. Descripción

Debemos destacar que BlueState se fundamenta en el metamodelo de clases de UML [9]. De esta forma, se garantizan las buenas características de este método de implementación, descritas en el apartado anterior. En [21] se expone con todo detalle la estructura del metamodelo empleado y su implementación.

En este trabajo se ha empleado el lenguaje C# para realizar la implementación del metamodelo de clases que define un diagrama de estados de UML. Este lenguaje ha ofrecido tanto los conceptos de orientación a objetos necesarios como otros mecanismos importantes de ejecución concurrente y sincronismo.

La parte más importante de la implementación ha consistido en trasladar la semántica del metamodelo a operaciones específicas en cada clase. Pese al carácter informal de la semántica que ofrece la especi-

	UML State Wizard Pro v1.62	UniMod v1.0.0	IBM Rational Rhapsody v7.5.3	Visual Paradigm v7.2	Poseidon for UML Embedded v6.0.2	Enterprise Architect v7.5	Smart State v4.1	XMI2SM v1.0.12	FXU v2.1	SinlaboreRT v1.7	BlueState v2.0
Diseñador gráfico integrado	Sí	Sí	Sí	Sí	Sí	Sí	Sí	No	No	No	No
Simulador de ejecución y eventos	No	No	No	No	No	No	No	No	No	No	Sí
Seguimiento gráfico	No	Medio	No	No	No	No	No	No	No	Bajo	Alto
Depuración de ejecuciones	Medio	Alto	Bajo	Bajo	No	No	No	No	Medio	Alto	Alto
Validación de restricciones	No	No	Bajo	Bajo	No	No	Bajo	Bajo	Alto	Medio	Alto
Elementos UML contemplados	Medio	Bajo	Medio	Bajo	Bajo	Bajo	Bajo	Bajo	Alto	Bajo	Medio
Ajuste a la especificación UML	Bajo	Bajo	Medio	Medio	Bajo	Bajo	Bajo	Bajo	Medio	Medio	Alto
Documentación general	Medio	Bajo	Bajo	Bajo	Bajo	Bajo	Bajo	Bajo	Bajo	Medio	Alto
Importación XMI	No	No	Alto	Bajo	Bajo	Bajo	No	Bajo	Bajo	Medio	Alto
Lenguajes destino	Bajo	Bajo	Alto	Medio	Bajo	Alto	Alto	Alto	Bajo	Bajo	Medio
Asistente de generación	Sí	Sí	Sí	Sí	Sí	Sí	Sí	No	Sí	No	Sí
Facilidad de uso	Bajo	Bajo	Medio	Alto	Bajo	Medio	Bajo	Bajo	Medio	Bajo	Alto
Mantenimiento de código	Bajo	Bajo	Medio	Medio	Bajo	Bajo	Bajo	Bajo	Bajo	Medio	Alto
Método de implementación	PE	PE	SC	PE	SC	SC	PE	SC	MC	SC	MC

* Método de implementación base: PE – Patrón estado, SC – Sentencias de condición, MC – Metamodelo de clases.

Figura 1: Comparación de soluciones.

ficación de UML, se ha seguido ésta de la forma más rigurosa posible, documentando en detalle las operaciones implementadas y las consideraciones tenidas en cuenta.

El resultado final de esta implementación es una biblioteca de clases, en este caso una biblioteca de clases para la plataforma .NET, que deberá ser incluida en los proyectos generados con BlueState.

Hemos de reseñar que en BlueState se han implementado la mayor parte de los elementos del diagrama de estados UML, a saber: estados simples y compuestos, pseudoestados iniciales, estados finales, comportamientos de estados entry, exit y doActivity, guardas, eventos de llamada, señales, transiciones de tipo external y local, regiones y pseudoestados de historia superficial y profunda.

Este conjunto de elementos son los más habitualmente empleados y mediante los cuales se permite definir casi cualquier comportamiento de un diagrama de estados.

Otra importante característica añadida al modelo es la posibilidad de ejecución concurrente de diagramas de estados y el envío de eventos entre éstos. De este modo, se refuerza la utilidad de BlueState como herramienta de aprendizaje, ya que permite mostrar la interacción entre diversas máquinas de estados, fundamental desde la perspectiva de la orientación a objetos.

3.2. Contexto de uso

El uso docente de BlueState debe enmarcarse en el contexto de sesiones prácticas en laboratorio. Se asume que el alumno ya está familiarizado con el empleo de herramientas de modelado. Además, se supone que se ha explicado y aplicado en la asignatura un modelo de proceso de software afín a UML, como puede ser el Proceso Unificado [14]. En este marco, el alumno ya debe estar familiarizado con los modelos de casos de uso y clases, y preferiblemente también con otros modelos de comportamiento de UML como los diagramas de secuencia.

Como se puede comprobar, este contexto docente es el habitual en la mayoría de programaciones de las asignaturas de Ingeniería del Software en nuestro entorno.

Entrando más en detalle, describimos a continuación una sucesión de posibles pasos para la utilización docente de la herramienta BlueState. No obs-

tante, por motivos de espacio nos vemos obligados a omitir muchos detalles. En [22] se profundiza en la descripción, ilustrada con imágenes de la interfaz de usuario de BlueState, así como vídeos que muestran su funcionamiento.

1. Diseño del diagrama de estados. Inicialmente se diseñará en la herramienta de modelado utilizada en la asignatura el diagrama de estados que represente el comportamiento sobre el que se trabajará.

2. Incorporación del código de operaciones y guardas. La implementación específica del código que deba ejecutarse en las operaciones de estados (entry, exit o doActivity) o las sentencias condicionales de guardas se incluirán directamente en el diagrama de estados. Para aumentar la compatibilidad entre herramientas de modelado se han contemplado los atributos "name" de estas operaciones o guardas para indicar los métodos o código en el lenguaje destino.

3. Importación y análisis de modelos. Una vez se ha diseñado el diagrama de estados, éste se exportará en un documento XMI desde la herramienta de modelado. El asistente de generación BlueState facilita al alumno la importación de este documento a través de un completo analizador XMI implementado *ad hoc*.

Este analizador permite identificar la información XMI de un documento XML y localizar la información propia de los diagramas de estados definidos. Una vez seleccionado un diagrama de estados, éste es transformado en una representación de objetos siguiendo el metamodelo de clases. En este punto se realiza la validación de restricciones del metamodelo UML y otras específicas.

Aunque en la implementación de este analizador se ha seguido de forma rigurosa el estándar XMI [10], también ha sido necesario incluir algunos parámetros de configuración para aumentar la compatibilidad con ciertas herramientas de modelado. Se ha comprobado su correcto funcionamiento con las herramientas de modelado Enterprise Architect [26], MagicDraw [12], Altova Umodel [2] y Visual Paradigm [13]. Este alto nivel de compatibilidad sin duda facilita la utilización de nuestra solución en otros contextos, ya que se posibilita su uso independien-

temente de la herramienta que se esté empleando en las prácticas.

4. Generación automática de código. El siguiente paso en la práctica será indicar, desde el asistente de generación BlueState, la clase para la que se generará el diagrama de estados y el lenguaje de programación destino de la implementación. El generador de código BlueState contempla los lenguajes C# y Visual Basic .NET, y puede expandirse a los lenguajes C++, J# y JScript.

El asistente de generación BlueState realizará automáticamente la implementación del código de la clase seleccionada a partir del diagrama de estados importado.

5. Ejecución de las máquinas de estados. Una vez realizada la implementación automática, las máquinas de estados son inicializadas para dar comienzo a su ejecución. Éste es el punto en que el alumno inicia la simulación y monitorización de la máquina de estados que ha diseñado.

Antes de describir estas funcionalidades, debemos destacar que el modelo implementado por BlueState permite la ejecución concurrente de múltiples máquinas de estados y el envío de eventos entre éstas. Desde el punto de vista del aprendizaje del modelado orientado a objetos, consideramos que de esta forma el alumno asimila mucho mejor la transmisión de los eventos incluidos en los modelos de comportamiento. Este concepto es básico en la implementación de cualquier software actual.

6. Simulación y depuración de la máquina de estados. Como se ha indicado anteriormente, una de las claves de la utilidad docente de BlueState radica en disponer de un mecanismo que permite una simulación de la ejecución de una máquina de estados. Esta simulación se controla a través de una interfaz gráfica que se carga de forma dinámica con los eventos definidos en el diagrama y permite generar éstos a través de pulsaciones de botones. En la figura 2 se muestra la interfaz del simulador configurada automáticamente a partir del diagrama de la figura. Asimismo, esta interfaz permite realizar un seguimiento de las entradas y salidas de estados y ajustar un retardo de ejecución para facilitar el seguimiento.

Al alumno también le resulta muy ilustrativo el

análisis del registro de activaciones de estados, que le permite realizar un seguimiento o depuración del comportamiento de cada uno de los objetos que entran en ejecución en una máquina de estados.

7. Seguimiento visual en tiempo real. Una gran ventaja de la utilización de BlueState es que, además del mecanismo de simulación descrito anteriormente, incorpora un importante módulo de seguimiento gráfico en tiempo real de la ejecución de diagramas de estados. Es decir, el alumno puede hacer un seguimiento visual de la ejecución de la máquina de estados que ha diseñado e implementado. El módulo de seguimiento, que puede emplearse junto con el simulador, permite la monitorización de la ejecución “en vivo” del código generado en su proyecto (o práctica).

Este módulo está disponible para cualquier diagrama de estados generado y permite realizar un seguimiento tanto de ejecuciones locales como de diagramas que se estén ejecutando en remoto. Además, permite también el seguimiento de la ejecución de varios diagramas de estados de forma simultánea. Estas últimas características permiten al profesor el diseño de una gran variedad de prácticas y ejercicios, ya que se posibilita la monitorización visual de máquinas de estados diseñadas por diferentes alumnos y que se están ejecutando simultáneamente en ordenadores diferentes.

La visualización se realiza empleando un complemento desarrollado para la herramienta de modelado Enterprise Architect, y que muestra de forma visual las transiciones y los estados activos de un diagrama de estados en ejecución.

4. Conclusiones

En este trabajo se ha conseguido el objetivo de crear una herramienta que facilite el aprendizaje de los diagramas de estados de UML, cubriendo las carencias existentes en la actualidad en cuanto a soluciones de este tipo.

En primer lugar, se ha desarrollado un entorno de aprendizaje totalmente funcional y que ofrece importantes posibilidades. Además de un generador de código, se incluyen otros componentes, fácilmente integrables, que permiten al alumno realizar una simulación y seguimiento visual en tiempo real de

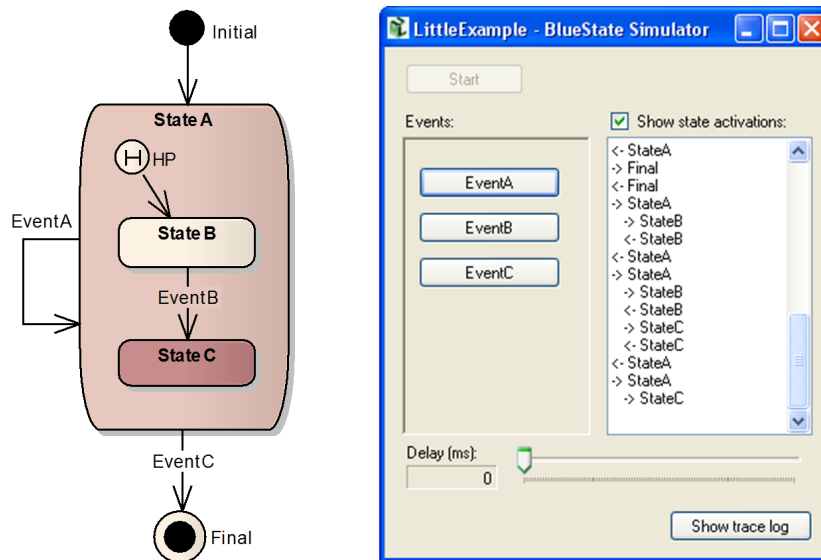


Figura 2: Simulación de una máquina de estados

los estados transitados, muy útil para la asimilación de los conceptos de diagramas de estados. Además, una de las características que pueden destacarse de BlueState es su facilidad de uso.

Por otra parte, nuestra solución es fácilmente integrable en el marco de la docencia práctica de Ingeniería del Software. La arquitectura de BlueState permite combinarla con diferentes herramientas de modelado e implementación comúnmente empleadas en asignaturas de esta materia. De este modo, se ofrece a los alumnos una experiencia eminentemente práctica del diseño, implementación, monitorización y depuración de las máquinas de estados. Además, dado que la herramienta se basa en el metamodelo de clases de UML, el alumno también profundiza más en el conocimiento de este lenguaje.

Debido a la reciente finalización de la aplicación que se describe en este trabajo, aún no se ha hecho un uso intensivo de ella en clase y por tanto no disponemos de datos cuantitativos de la mejora en el aprendizaje. No obstante, las experiencias realizadas hasta el momento de escribir este artículo confirman que los conocimientos sobre máquinas de estados adquiridos son mayores y más profundos. Para contrastar esta afirmación se compararán las calificaciones obtenidas este curso en ejercicios relacionados

con diagramas de estados con las obtenidas en ejercicios similares en cursos anteriores. Además, como valoración cualitativa, la mayoría de los alumnos ha respondido en un cuestionario anónimo que considera BlueState una herramienta muy útil como soporte al aprendizaje de los diagramas de estados.

Por último, para facilitar su integración en otros contextos, se ha intentado que nuestro entorno sea lo más independiente posible. Apoyándonos en estándares como XMI se posibilita su uso combinado con cualquier herramienta de modelado que satisfaga dicho estándar. Además, la estructura intermedia en la que se sustenta el motor de generación de código de BlueState permite generar código en diferentes lenguajes de programación. Además, se ha elaborado una extensa documentación técnica y diversos estudios de rendimiento que se puede consultar en [21].

Por último, concluimos que la utilización del entorno descrito en este trabajo refuerza considerablemente el aprendizaje del alumno, especialmente en uno de los pilares actuales de la Ingeniería del Software como es el lenguaje de modelado UML. Desde un punto de vista más general, el uso de BlueState hace que el alumno asimile la importancia de la correspondencia directa y correcta entre una especificación formal y su implementación, una de las metas

de la buena práctica del desarrollo de software.

Referencias

- [1] P. Adamczyk. The anthology of the finite state machine design patterns. In *Proceedings of the Pattern Languages of Programs Conference (PLoP)*, 2003.
- [2] Altova. UModel. <http://www.altova.com/umodel.html>.
- [3] G. Booch, J. Rumbaugh, and I. Jacobson. *El lenguaje unificado de modelado: guía del usuario*. Pearson Educacion, 2a edition, 2006.
- [4] T. Cargill. *C++ programming style*. Addison-Wesley Pub. Co., 1992.
- [5] eVelopers Corporation. UniMod. <http://unimod.sourceforge.net/intro.html>.
- [6] A. Derezińska and R. Pilitowski. Realization of UML class and state machine models in the C# code generation and execution framework. *Informatica*, 33(4):431–440, 2009.
- [7] B. Douglass. *Real-time UML : developing efficient objects for embedded systems*. Addison-Wesley, 1998.
- [8] E. Gamma. *Design patterns : elements of reusable object-oriented software*. Addison-Wesley, 1995.
- [9] Object Management Group. UML 2.3. <http://www.omg.org/spec/UML/2.3/>, 2010.
- [10] Object Management Group. XMI specifications. www.omg.org/technology/documents/spec_catalog.htm, 2011.
- [11] J. V. Gurp and J. Bosch. On the implementation of finite state machines. In *Proc. IASTED International Conf. on Software Engineering and Applications, (SEA'99)*, pages 172–178, Scottsdale, AZ, USA, 1999.
- [12] No Magic Inc. MagicDraw UML. <http://www.magicdraw.com/>.
- [13] Visual Paradigm International. Visual paradigm for UML. <http://www.visual-paradigm.com/product/vpuml/>.
- [14] I. Jacobson, G. Booch, and J. Rumbaugh. *El proceso unificado de desarrollo de software*. Addison Wesley, 2006.
- [15] A. Knapp and S. Merz. Model checking and code generation for UML state machines and collaborations. In *Proc. 5th Workshop on Tools for System Design and Verification*, pages 59–64, Reisenburg, Germany, 2002.
- [16] H. J. Kohler, U. Nickel, J. Niere, and A. Zundorf. Integrating UML diagrams for production control systems. In *Proceedings of the 22nd International Conference on Software Engineering - ICSE '00*, pages 241–251, Limerick, Ireland, 2000.
- [17] P. Metz, J. O'Brien, and W. Weber. Code generation concepts for statechart diagrams of the UML v1.1. Universitat Wien, 1999.
- [18] C. Mocek. UML statechart framework. <http://uml-statecharts.sourceforge.net/>.
- [19] I. A. Niaz and J. Tanaka. Mapping UML statecharts to Java code. In *Proc. IASTED International Conf. on Software Engineering (SE 2004)*, pages 111–116, 2004.
- [20] I. A. Niaz and J. Tanaka. An Object-Oriented approach to generate Java code from UML statecharts. *International Journal of Computer & Information Science*, 6(2):83–98, 2005.
- [21] A. Ortigosa. *Generador de código para diagramas de estados UML*. 2010.
- [22] A. Ortigosa and C. Rossi. BlueState. <http://bluestate.lcc.uma.es>, 2011.
- [23] G. Pintér and I. Majzik. Program code generation based on UML statechart models. *Periodica Polytechnica-Electrical Engineering*, 47:187 – 204, 2003.
- [24] M. Samek and P. Montgomery. State-oriented programming. *Embedded Systems Programming Magazine*, 13(8):22–43, 2000.
- [25] North State Software. UML state machine code framework. <http://www.northstatesoftware.com/>.
- [26] Sparx Systems. Enterprise architect UML modeling tool. <http://www.sparxsystems.com/>.
- [27] T. Tomura, K. Uehiro, S. Kanai, and S. Yamamoto. Developing simulation models of open distributed control system by using object-oriented structural and behavioral patterns. In *4th IEEE International Symposium on Object-Oriented Real-Time Distributed Computing. ISORC 2001*, pages 428–437, 2001.
- [28] S. Yacoub and H. Ammar. A pattern language of statecharts. In *Proceedings of Fifth Annual Conference on the Pattern Languages of Programs, PLoP'98*, 1998.

Animación interactiva de algoritmos para cursos de introducción a la programación

Lluís Ribas Xirgo

Departament de Microelectrònica i Sistemes Electrònics (MiSE)

Universitat Autònoma de Barcelona (UAB)

Edif. Q, Campus UAB

08193 Bellaterra (Cerdanyola del Vallès)

Lluís.Ribas@uab.cat

Resumen

La enseñanza de los algoritmos básicos y de los principios de la programación requiere, entre otras cosas, ilustrar cómo se ejecutan los programas. Para ello, pueden emplearse distintas técnicas de animación, desde realizar el seguimiento de la ejecución con esquemas en una pizarra hasta el empleo de material multimedia más o menos interactivo. En este trabajo se presenta una herramienta basada en Javascript que permite realizar un seguimiento visual de la ejecución de programas simples sobre un navegador cualquiera, interactuar con él y, en última instancia, que profesores y estudiantes puedan manipularlo libremente para observar los efectos de los cambios en los programas que realicen.

Summary

Teaching basic algorithms and programming principles needs, among other things, to illustrate how programs are executed. For it, different animation technologies can be used, ranging from following program executions with schemes in a blackboard to the more or less interactive usage of multimedia material. In this work, it is presented a Javascript-based tool that allows doing visual tracing of the execution of simple programs on any web-browser, interacting with it and, last but not least, freely manipulating programs by teachers and students so to show off the effects of the changes.

Palabras clave

Algoritmos animados, Javascript, presentación interactiva, programación, visualización de *software*.

1. Motivación

Para un estudiante que se inicia en la programación resulta fundamental comprender cómo se ejecutan los programas y el efecto que causan en la información que manejan.

En un aula tradicional, los profesores ilustran la ejecución de un programa escribiendo, por ejemplo, la secuencia de instrucciones en una parte de la pizarra y, con más o menos fortuna, la representación gráfica de los datos que se manipulan en la parte adyacente. Con el tiempo, la tiza y la pizarra han dado paso a otras herramientas, incluidas las pizarras electrónicas, pero el principio de funcionamiento sigue siendo el mismo.

En muchos casos, los seguimientos de ejecuciones se han transformado en animaciones específicas, más o menos “programables”. Este último término se refiere, por ejemplo, a que para los algoritmos de ordenación se pueda variar el número de elementos o la distribución inicial de los mismos para mostrar sus ventajas e inconvenientes.

De hecho, se pueden encontrar videos en Youtube que ilustran el funcionamiento de una gran variedad de algoritmos, desde los que manipulan pilas y colas hasta otros que trabajan con grafos. También hay una gran variedad de aplicaciones especializadas ([2] y [7], por ejemplo) que permiten ciertos grados de interacción con la animación.

Las animaciones de los algoritmos han permitido, entre otras cosas, agilizar la presentación, introducir elementos gráficos más atractivos y hacerlo accesible a los estudiantes, a través de la web.

Sin embargo, se carece de aplicaciones que se ejecuten directamente en los navegadores, con lo

que los usuarios de las aplicaciones correspondientes deben de realizar una instalación previa de las mismas en plataformas concretas.

En cualquier caso, las animaciones resultan de gran ayuda para la comprensión de los algoritmos pero no permiten interactuar con ellos. Esto les resta interés tanto para los docentes como para los estudiantes. En el primer caso, el uso de la animación en clase resulta útil para ilustrar el concepto pero no para resolver dudas acerca del mismo, especialmente aquellas que deban aclararse con mecanismos que se alejen de las capacidades de programación de la animación. En el caso de los estudiantes, el hecho de que sean receptores pasivos de la información provoca, en muchos casos, el uso de la animación se limita a su función meramente ilustrativa y, a lo sumo, algunos variarán los parámetros de la misma.

A la vista de todo ello, es interesante poder disponer de algún recurso que permita, además de animar la ejecución de los algoritmos, interactuar con ellos. En este sentido, la idea sería parecida a la de disponer de un depurador con una visualización de los datos específica para cada programa.

Por desgracia, el uso de los depuradores de los entornos de programación sólo es factible para los docentes y, además, la presentación de la información (código y datos) no tiene orientación educativa.

A medio camino entre las animaciones estáticas y los depuradores hay una variedad de herramientas de carácter docente [9] que permiten disponer de animaciones programables muy efectivas a cambio de perder generalidad.

El recurso que se presenta pretende resolver los inconvenientes de los depuradores de los entornos de programación en cuanto a su uso en la animación de algoritmos, manteniendo las ventajas que aportan los entornos de programación en cuanto a "interactividad", ya que se puede modificar el programa que se en cualquier momento. Por lo tanto, se sitúa en el mismo plano que las herramientas educativas antes referidas aunque, a diferencia de las mismas, se basa tecnología web para que sea accesible y utilizable desde cualquier dispositivo con navegadores de Internet y se organiza de manera que sea fácilmente modificable y ampliable por sus usuarios.

Para ello, la aplicación se ofrece como *software* de código abierto, con una interfaz de usuario básica que puede complementarse con funciones escritas por los mismos usuarios, de manera que cada profesor pueda adaptarlo a sus necesidades y que, por otra parte, puede servir de ejemplo para estudiantes avanzados.

El recurso ha sido programado Javascript, puesto que se trata de un lenguaje bien soportado por los navegadores de Internet, que es ampliamente conocido y cuyas aplicaciones no requieren de ninguna instalación más que la de algún navegador de Internet capaz de interpretarlo.

Así pues, es posible incluso un escenario en el que el docente emplea el recurso en el ordenador conectado al proyector y los estudiantes lo pueden recrear en sus portátiles o dispositivos móviles. Más aun, pueden realizar ejercicios directamente en ellos.

Antes de pasar a describir la herramienta que se ha desarrollado, se revisará el estado del arte en este tema. En la sección 3 se detalla la arquitectura de la aplicación. La siguiente se dedica a presentar diversos casos de uso de la misma. Finalmente, en la conclusión se repasan los aspectos más destacados tanto del recurso como de su aplicación y se comentan los resultados preliminares de una primera experiencia de su uso real.

2. Animación de algoritmos

La animación de algoritmos es una parte del área de visualización de software, que recibe mucha atención tanto por el impacto que tiene en la productividad del desarrollo de aplicaciones como en el ámbito de la enseñanza.

En [3] se clasifican los distintos sistemas de animación de algoritmos según el método en que se basan: Desde aquellos dirigidos por sucesos hasta los más automatizados. La idea es visualizar la ejecución de un programa cada cierto tiempo:

- Cuando hay cambios de estado significativos (*state-driven*), que afectan a la visualización del mismo.
- Cuando se producen determinados sucesos (*event-driven*) como, por ejemplo, un cambio de valor en una de las variables observadas o llegar a un punto concreto del programa.

- Por observación de su evolución si está construido de forma visual. En cualquier caso, la programación visual puede verse como un elemento complementario de la visualización del software.
- De forma automática, a partir de un análisis del grafo de flujo de control y datos del programa.

En todas las técnicas mencionadas es necesario identificar los datos a mostrar y presentarlos de forma gráfica en una manera bien adaptada al problema que resuelve el programa.

En el ámbito educativo, los problemas son relativamente simples y podrían emplearse formas automáticas de identificación de variables clave, pero sería difícil tener una visualización de las mismas bien ajustada a su uso en docencia.

La programación visual tiene sus ventajas, en cuanto a facilitar el aprendizaje de la algorítmica [4] pero no constituye, por sí misma, una herramienta para la visualización de la información que manipulan los programas. Por otra parte, el empleo de entornos de programación visual requiere de un tiempo de aprendizaje que se resta del dedicado a los entornos de programación en lenguajes textuales como, por ejemplo, el C.

La utilización de técnicas basadas en sucesos proporciona la ventaja de poder visualizar la información del programa sólo en los momentos más interesantes, pero exige determinarlos.

Las actualizaciones de las visualizaciones por cambio de estado son las más comunes en sistemas que se focalizan en la visualización. De hecho, muchos de los sistemas “dirigidos por estados” son sistemas que permiten especificar cómo mostrar la información que manipulan los programas e incorporan mecanismos simples de seguimiento de su ejecución.

El recurso que se presenta se inspira en este último tipo de soluciones, puesto que los algoritmos a animar son relativamente simples y pueden ser seguidos paso a paso.

La visualización se hace con tecnología DHTML, de manera que la especificación de cómo mostrar las variables de un determinado programa se hace mediante una función en Javascript. Los algoritmos se describen en forma de programa con una sintaxis de Javascript restringida, especialmente en lo que atañe a instrucciones de control de flujo.

La ejecución de los programas se lleva a cabo de forma interpretada, igual que en [8], por lo que pueden alterarse en cualquier momento, incluso durante su ejecución (o animación).

3. Organización de la aplicación

El recurso consiste en una pequeña aplicación en Javascript, llamada **jsGorithm**, y un conjunto de casos de uso con ejemplos. En esta sección de comentan los aspectos que atañen la especificación del programa y a su arquitectura.

El programa es de código abierto para poder ser modificado según la conveniencia de cada uso. En cualquier caso, está dividido en dos partes: la de ejecución de programas y la de monitorización de variables.

En general, la primera parte no habría de requerir ningún cambio y la segunda sería la que habría de adaptarse a cada uso que se le diera.

3.1. Edición y ejecución de programas

Los programas pueden editarse con cualquier editor de texto o bien emplear la ventana de edición de la misma aplicación. Los programas tienen que estar construidos como secuencias de instrucciones en una única línea.

Las instrucciones que se pueden emplear se resumen en la tabla 1: son las básicas para realizar cálculos y movimientos de datos y para construir estructuras alternativas e iterativas.

Instrucción
<code>variable = expresión;</code>
<code>if() {</code>
<code> } else {</code>
<code>while(expresión) {</code>
<code> } /* fin de bloque */</code>
<code>do {</code>
<code> } while(expresión);</code>

Tabla 1. Repertorio de instrucciones admitidas.

Con estas instrucciones se puede construir cualquier programa. La instrucción `for` no se ha incluido por ser, de hecho, una estructura compleja que puede simularse con un bucle de tipo “mientras”.

En cualquier caso, podría incorporarse en el conjunto ampliando la función de compilación de programas fuente y modificando ligeramente la que se ocupa de ejecutar los programas ya

compilados. (La compilación consiste en generar el grafo de flujo de control y datos del programa fuente para facilitar el seguimiento de la ejecución.)

Las líneas de comentario iniciadas con “//” también se admiten, de la misma forma que los comentarios de bloque, siempre que el bloque esté comprendido en una misma línea.

Las variables que emplea el programa pueden ser definidas en cualquier momento con una asignación. Todas ellas son de carácter global y ninguna puede empezar por “_gr”, puesto que es la denominación de todos los objetos de la aplicación.

Para ejecutar un programa hay que cargarlo en la ventana de ejecución haciendo clic en el botón de “Load”. Al hacer la carga, el código se verifica y, en caso de haber errores, se comunican al usuario. Los errores más comunes atañen al uso de

instrucciones no previstas y a construcciones de control no estructuradas como, por ejemplo, un *else* fuera de lugar.

De hecho, con la carga del código se determina su grafo del flujo de control y de datos para que pueda seguirse su ejecución. En la ventana del código se numera cada instrucción, se indica la posición actual del contador de programa con “>>” y también se muestran los valores a sumar al contador en caso de que el resultado de la ejecución sea una condición cierta o falsa. Las instrucciones que no evalúan ninguna condición tienen ambos valores a 1.

Las instrucciones del programa se muestran con un sangrado a la derecha que refleja a qué estructura pertenecen.

Después de cargar el programa en la ventana de ejecución, puede iniciarse la misma apretando el botón “Play” o el de “Step”.

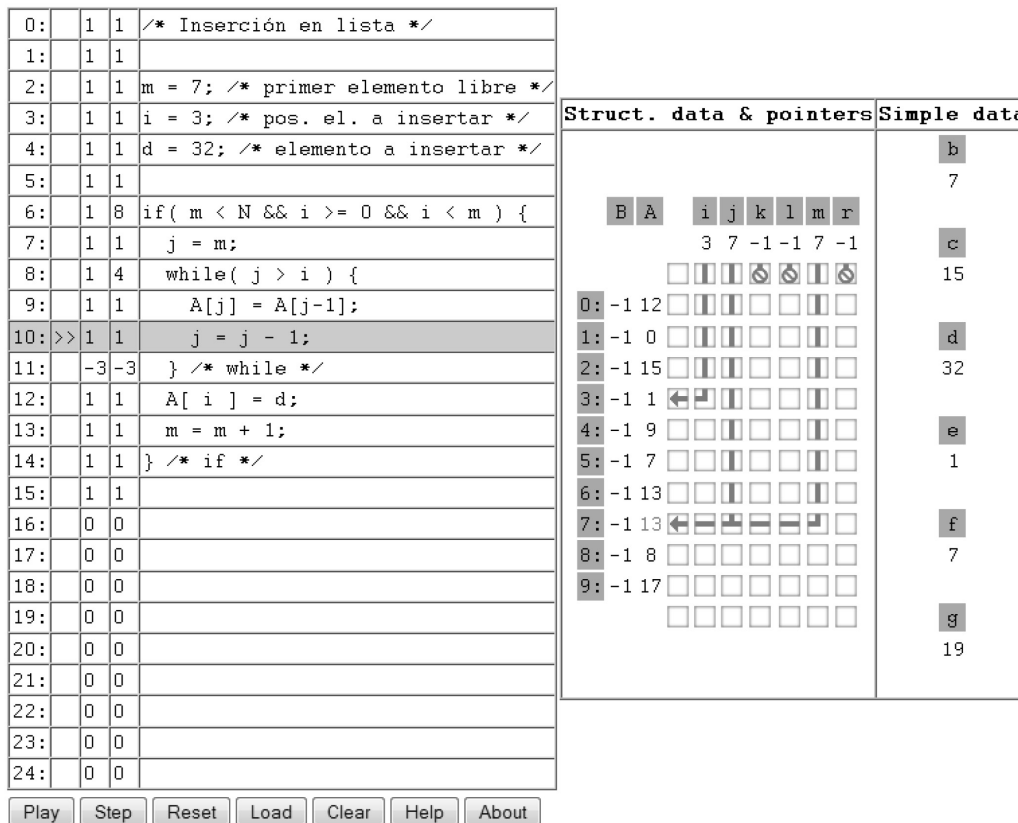


Figura 1. Ventanas de seguimiento de código y visualización de variables de jsGorithm.

El contador de programa apunta a la siguiente instrucción a ejecutar. Después de que se haga clic en “Step”, el contador se modifica convenientemente: o se incrementa en una unidad o se le suma un desplazamiento, en caso de que la condición de salto se cumpla.

La ejecución de cada paso implica una llamada a la función `eval()` de Javascript para que evalúe la instrucción. Por este motivo, los programas que pueden construirse son, de hecho, secuencias de instrucciones en Javascript que pueden ser vistas, igualmente, como si fueran en C, C++ o C#, por ejemplo.

De hecho, algunos cursos de introducción a la programación ya emplean directamente Javascript [5] en lugar de lenguajes compilados. Otros siguen prefiriendo estos últimos, aunque emplean intérpretes como el Ch [1].

La ejecución en modo continuo no es más que la realización de una secuencia de pasos que se inicia al apretar el botón de “Play”. La ejecución se detiene al llegar a la última instrucción del programa o cuando se aprieta “Pause”.

Haciendo clic en “Reset”, la ejecución del programa se para, el contador de programa se pone a 0 y se reinicializa la ventana de variables. Tiene, de hecho, el mismo efecto que si se cargara la página de nuevo.

3.2. Monitorización de variables

La organización de la aplicación tiene su reflejo en la estructura de la página web que la contiene: se divide en varias áreas. El área de visualización de las variables es una sección `<DIV>` del documento HTML correspondiente que se denomina “varArea”.

Dicha área debe ser inicializada y actualizada con las funciones `_grVarWinInit()` y `_grVarWinUpdate()`, respectivamente. Esta última función se llama después de ejecutar una instrucción (con “Step” o “Play”) o de hacer un “Reset”.

Aunque la idea original es que cada usuario se cree las funciones que visualizan los datos que maneja un programa, resulta más práctico contar unas funciones en la aplicación que ya estén listas para ser usadas.

Por ello, `jsGorithm` lleva incorporadas dos funciones de inicialización y actualización del área de variables para un conjunto de datos suficientemente amplio como para cubrir la

mayoría de casos de uso habituales de la herramienta.

A continuación, se detalla la configuración básica de visualización del área de variables que se ofrece con las funciones `_grVarWinInit()` y `_grVarWinUpdate()` que se ofrecen con `jsGorithm`. Dichas funciones trabajan con un conjunto dado de variables:

- `A` es un vector de N (10) elementos que se inicializa a valores enteros aleatorios, para que pueda ser empleado en algoritmos de ordenación, por ejemplo.
- `B` es otro vector de N elementos que puede actuar como vector auxiliar o bien como vector de direcciones en casos de listas con apuntadores.
- `i` es el índice principal del primer vector
- `j` es un índice secundario
- `k` es un índice del elemento clave (*key*)
- `l` es un índice de elemento en la izquierda
- `m` significa índice de elemento en el medio
- `r` es un índice de elemento en la derecha
- `b` es una variable para datos y de *buffer* para resultados intermedios
- `c` es un contador
- `d` es para datos, en general
- `e` es una variable extra para datos
- `f` es para contener resultados finales
- `g` es una variable de propósito general

Todas las variables se pueden emplear libremente en el programa. Los significados que se les han dado en el listado anterior son a efectos mnemónicos.

Las funciones que inicializan y actualizan el espacio de estas variables muestran los dos vectores en dos columnas en el lado izquierdo y las variables de índice en las columnas siguientes, de manera que puedan trazarse flechas que indiquen a qué elementos de los vectores apuntan. Las variables de datos de muestran en la parte derecha.

Se trata de una visualización relativamente simple, pero efectiva. Es conveniente, en cualquier caso, limitar las variables visualizadas a aquéllas que se empleen en los programas. Para ello, las funciones que se proveen emplean una lista que puede ser modificada libremente. Así, por ejemplo, puede mostrarse sólo un vector o un conjunto más reducido de índices, cosa que

facilita la comprensión de la animación de los algoritmos.

De manera similar, es posible, además, emplear nombres de variables mucho más significativos que los provistos.

4. Casos de uso

En este apartado se comentan los casos de uso probados con el programa jsGorithm a modo de ilustración de sus aplicaciones.

4.1. Algoritmos básicos de programación estructurada

Se trata de ejemplos de cálculos sencillos que requieran de concatenación de estructuras fundamentales de programación, como la alternativa o la iterativa.

En todos los casos basta con unas pocas variables: las que definen el problema, las que representan la solución y alguna de intermedia.

Como se ha comentado, la función de visualización que se incluye en jsGorithm emplea una lista de las variables que mostrará en la ventana correspondiente. Así pues, bastará con adaptar esa lista a las variables que se requieran para cada algoritmo.

El profesor habrá de hacer los programas correspondientes con cualquier editor de texto y, en el momento de utilizar jsGorithm, será necesario que copie el texto de uno de los programas en la ventana de edición de la aplicación.

Suponiendo que esté realizando una presentación en clase, bastará con cargar la página de *jsGorithm.html* desde cualquier navegador de Internet. La ventaja añadida de este sistema de hacer presentaciones es que los navegadores permiten variar el tamaño de lo visualizado, con lo que se puede adaptar a las necesidades de cada auditorio.

Para poner estas animaciones al alcance de los estudiantes basta con que tanto el texto de los programas como la versión correspondiente de jsGorithm sea accesible a ellos a través de Internet.

Esto abre varias posibilidades, incluida la del uso simultáneo por profesores y alumnos en clase, con lo que pueden realizarse clases de teoría mucho más dinámicas o de problemas en los que los estudiantes puedan resolver de una manera

más intuitiva los ejercicios que les propone el profesor.

4.2. Esquemas algorítmicos para el tratamiento de secuencias

La importancia del procesamiento secuencial hace que merezca la pena que los estudiantes dediquen un tiempo a comprender y saber poner en práctica los esquemas algorítmicos correspondientes.

Se trata de esquemas de recorrido y de búsqueda para los que pueden encontrarse ejemplos tanto de secuencias temporales como espaciales.

Para las segundas, es necesario incluir vectores que almacenen la secuencia a procesar e índices de los mismos. Con las funciones de visualización de datos que hay, se pueden mostrar tanto vectores como índices. Los índices, además de mostrarse como valor, se muestran con una flecha que señala qué posición del vector están apuntando, tal como se puede apreciar en la fig. 1.

Así pues, se ha desarrollado una versión de jsGorithm para esquemas algorítmicos de recorrido y búsqueda, con varios programas de ejemplo.

De manera similar al caso anterior, los profesores pueden emplearlo en clase y también ofrecerlo a los estudiantes para que resuelvan ejercicios en el tema.

4.3. Algoritmos de ordenación

Se trata de una clase de algoritmos de la cual existe una gran variedad de recursos educativos, por lo que debe fijarse cuál es el objetivo de la animación.

En el caso que el objetivo sea mostrar un algoritmo que requiera de un anidamiento de esquemas de procesado de secuencias, basta con emplear la misma versión de jsGorithm que para el caso anterior con los programas de ordenación que se quieran mostrar.

Hay que tener presente que jsGorithm construye el grafo del flujo de ejecución de los programas que carga, que no pueden incluir definiciones de funciones. Por lo tanto, no puede animar la ejecución de programas recursivos. Así pues, es válido para aquellos algoritmos de ordenación que se resuelvan de esta manera.

Esta restricción no es muy significativa para cursos de introducción a la programación o de fundamentos de informática.

Si el objetivo es comparar la ejecución de varios algoritmos, hay que emplear el mismo número de ventanas del navegador con el jsGorithm. Por cuestiones prácticas, sólo se deben de comparar dos algoritmos que deben estar en sendas ventanas, una al lado de la otra. Dado que no es posible iniciar los dos programas al mismo tiempo, se recomienda introducir alguna instrucción de relleno (basta una línea de comentario, por ejemplo) al principio del programa cuya ejecución se empiece más tarde. Hay que tener presente también que los programas a comparar tengan una inicialización de los vectores a ordenar con los mismos valores.

Finalmente, especialmente para ilustrar los temas de la complejidad algorítmica, es interesante disponer de una serie de casos con vectores inicializados a valores concretos y que los mismos programas realicen un conteo del número de comparaciones e intercambios.

4.4. Estructuras de datos dinámicas

Para ilustrar las aplicaciones de las estructuras de datos dinámicas se pueden emplear listas en vectores. (No se tratan aquí ni árboles ni grafos.)

En el caso de las pilas y de las colas, puede bastar con un único vector, pero hay que tener un programa para cada operación: uno de inserción y otro de eliminación, como mínimo. De cara a realizar las presentaciones, es conveniente disponer de un programa de inicialización de la estructura de datos.

El guión de la presentación puede ser, para el caso de una pila, como sigue. Se explica la representación de la pila en jsGorithm y se inicializa con algunos valores, cargando el programa correspondiente. Luego se carga el *push* y se repiten varias ejecuciones hasta llenar la pila. Después se vacía completamente con ejecuciones repetidas del programa de *pop*. (La repetición se consigue haciendo un "Reset" seguido de un "Play" después de que el programa cargado haya llegado a su fin.) Finalmente, puede cargarse de nuevo el programa de *push* para poner algún elemento nuevo en la pila.

Hay que hacer notar que el "Reset" no modifica el contenido de las variables del programa y que, en la actualidad, el cambio de esos valores sólo puede hacerse por programa.

4.5. Variables dinámicas

En la actualidad, jsGorithm no ofrece soporte para trabajar con variables dinámicas porque se tratan de forma diferente en Javascript que en C y porque su representación visual es más compleja.

Aun así, se puede emular con dos vectores, uno de ellos representando la dirección y otro el contenido de una determinada variable.

En cualquier caso, se trata de una opción de relativa utilidad, puesto que lo interesante sería visualizar las variables y sus interrelaciones de forma más gráfica.

5. Conclusiones

La animación de los algoritmos y programas es muy útil para la enseñanza y el aprendizaje en cursos de introducción a la informática o a la programación. En este artículo se ha presentado un recurso destinado a la animación de programas que puede ser empleado tanto en el aula como de forma individual.

El programa jsGorithm incorpora tanto una parte de seguimiento de programas como otra para la visualización de las estructuras de datos que manipulan. Esta segunda parte se puede adaptar a cada uso en particular para hacer más comprensible la visualización del conjunto.

Este recurso está disponible en [6] para su descarga y para su prueba directa vía web. En este sentido, hay que destacar que las ventajas del recurso propuesto respecto a otras herramientas similares residen, precisamente, en que es posible emplearlo directamente en cualquier navegador tanto de forma remota como local y, sobre todo, en que se puede modificar libremente. En concreto, es posible cambiar la parte de visualización de variables para adaptarla a casos de uso específicos.

En este trabajo también se han presentado diversos casos de uso para ilustrar su funcionamiento. Todos ellos han sido probados durante el primer semestre del curso 2010/11 en la asignatura de Fundamentos de Informática, buena parte de la cual es una introducción a la programación.

Los resultados en el aula de este primer curso de aplicación son positivos en cuanto a la mejora de la agilidad en la presentación de los algoritmos y de su posterior seguimiento en clase. La percepción de los estudiantes es buena pero las

encuestas informales realizadas han reflejado también un cierto descontento con la rapidez de las explicaciones. En cuanto al uso de este recurso por los estudiantes, cabe decir que ha sido bastante residual.

Así pues, se prevé continuar con el uso de este recurso en clase e introducir mecanismos para facilitar su uso por parte de los estudiantes.

En próximas ediciones del curso, se realizarán encuestas específicas que permitan mejorar el entorno de jsGorithm de acuerdo con las necesidades que se detecten.

Referencias

- [1] H. H. Cheng. *C For Engineers & Scientists, An Interpretive Approach*, McGraw-Hill March 2009.
- [2] Concepción, A.I.; Cummins, L.E.; Moran, E.J.; Do, M.M. “Algorithma 98: An Algorithm Animation Project” In *Proc. of SIGCSE’99*. 301–305. ACM, New York, NY, USA. 1999.
- [3] Kerren, A.; Stasko, J.T. “Algorithm Animation” *Software Visualization State of the Art Survey*. Chapter 1, 1–15. Springer Lecture Notes in Computer Science LNCS 2269, (Editor Stephan Diehl), 2002.
- [4] Powers, K.; Gross, P. (moderators) “Tools for Teaching Introductory Programming: What Works?” In *Proc. of SIGCSE’06*. 560–561. Houston, Texas, USA. March 2006.
- [5] D. Reed. *A balanced introduction to computer science*, Prentice-Hall, 2008.
- [6] Ribas Xirgo, Ll. “jsGorithm: simple algorithm animation,” Sourceforge. Since 2011-04-18. [jsgorithm.sourceforge.net/]
- [7] Rößling, G. *ANIMAL-FARM: An Extensible Framework for Algorithm Visualization*. VDM Verlag. Saarbrücken, 2008. [Disponibile en www.algoanim.info]
- [8] Sankupellay, M.; Subramanian, P. “Teaching C Programming with the Aid of an Interpreter – Online Interpreter for Novice C Programmer” *Jurnal Teknologi*, 43(D). 33–44. Universiti Teknologi Malaysia. Dec. 2005.
- [9] J. Urquiza-Fuentes, J.A. Velázquez-Iturbide. “A Survey of Successful Evaluations of Program Visualization and Algorithm Animation Systems,” in *ACM Trans. on Computing Education*, Vol. 9, No. 2. June 2009.

Hiperión: Sistema personalizado para la recomendación de actividades en base a competencias no adquiridas

Jesús Serrano-Guerrero, Andrés Cerezo,
Francisco P. Romero, José A. Olivas
Departamento de Tecnologías y Sistemas de Información
Universidad de Castilla-La Mancha
Paseo de la Universidad 4
13071 Ciudad Real (España)
{jesus.serrano, franciscop.romero, emilio.fernandez,
joseangel.olivas}@uclm.es

Víctor Menendez-
Dominguez
Facultad de Matemáticas
Universidad Autónoma de Mérida
Periférico Norte
97110 Mérida (Yucatán, México)
mdoming@uady.mx

Resumen

El presente trabajo muestra una herramienta llamada Hiperión cuyo objetivo es la recomendación de actividades evaluables de forma personalizada, con el fin de que aquellos alumnos que no hayan adquirido una serie de competencias a lo largo del periodo de clases lectivas, las adquieran durante los periodos extraordinarios. Dicha herramienta permite varias funcionalidades que otros sistemas como Moodle no permiten. Entre estas funcionales encontramos la definición y estructuración de asignaturas en base a competencias, la definición y evaluación de actividades evaluables en base a competencias y la recomendación automática y personalizada de actividades en función de las calificaciones individuales de cada alumno. Para comprobar la efectividad de dicho framework, se ha probado su funcionamiento con una clase de Fundamentos de Programación obteniendo unos buenos resultados.

Summary

This work presents a tool called Hiperión whose objective is to recommend personalized tasks to be assessed. These tasks allow students to acquire skills which have not been reached during the academic year. This tool allows several functionalities that other systems like Moodle do not allow. Among these capabilities, we can highlight the definition of each subject based on skills, the definition of activities based on skills and the automatic and personalized recommendation of tasks for each student depending on his/her marks. In order to assess the effectiveness of this framework, Hiperión has been used with the students of a class of Programming Foundations reaching good results.

Palabras clave

Evaluación personalizada, recomendación personalizada, competencias, actividades evaluables, etiquetas lingüísticas

1. Motivación

Según afirma el REAL DECRETO 1393/2007, de 29 de octubre [5], las competencias que deben adquirirse en cada título, deberían poder ser evaluables. Así pues, conforme a dicha afirmación, los docentes deberían plantear las asignaturas de tal manera que su objetivo fuera ser capaces de evaluar las competencias que van o no adquirieron sus alumnos.

Para poder ayudar a los docentes a cumplir dicho objetivo, son necesarias herramientas que les apoyen a la hora de enfocar sus asignaturas en base a competencias. Hoy por hoy, dado que la docencia en base a competencias está todavía en sus inicios, no existen herramientas docentes cuyo enfoque principal sean las competencias.

Así pues el objetivo de este trabajo es la presentación de una herramienta cuyo enfoque principal son las competencias. La herramienta Hiperión permite, entre otras opciones, la definición de una asignatura conforme a las competencias que puedan estar definidas en la Memoria de Grado de cualquier escuela o facultad. De igual manera, esta herramienta permite definir las pruebas evaluables que realizarán los alumnos a lo largo del curso en base a las competencias propias de la asignatura, así como la calificación de las mismas. Y finalmente, la funcionalidad principal en la que se centra este trabajo, la recomendación de actividades

evaluables, de forma automática y personalizada, que permitirán a cada alumno adquirir aquellas competencias que no han adquirido durante el curso académico.

Así pues, nuestra herramienta es capaz de, a medida que vaya transcurriendo el curso, recomendar unas actividades u otras para ir cubriendo las deficiencias observadas en cada alumno y así, conseguir una evaluación mucho más personalizada y adecuada a las necesidades de cada estudiante.

El resto del trabajo se organiza de la siguiente manera: la siguiente sección presenta algunos trabajos relacionados con la evaluación personalizada y las herramientas docentes que existen actualmente. En la sección 3 se presentan las principales características de la plataforma Hiperión, mientras que la sección 4 presenta el algoritmo implementado para la recomendación de actividades de forma personalizada. En la sección 5 presenta un caso de estudio sobre el funcionamiento de dicha plataforma, y finalmente en la sección 6 se presentan algunas conclusiones sobre este trabajo.

2. Trabajos relacionados

Según García, “la evaluación pedagógicamente considerada tiene un sentido orientador. A través de ella se intenta ver cuál es la situación de un estudiante, con objeto de ayudarlo a tomar decisiones más adecuadas para el desarrollo de sus posibilidades y la comprensión de sus limitaciones” [4]. Así pues, la forma que tiene el docente de conocer a cada individuo es la evaluación. Un perfecto conocimiento del alumnado permite a cada profesor situarse en una posición estratégica para afrontar los retos que cada individuo puede plantear. Según Estupiñán, “no es posible emitir juicios de valor sin un mínimo de conocimiento acerca del objeto evaluado. Sólo así es posible superar la fiabilidad” [2].

Así pues queda patente la necesidad de evaluar de forma individual a cada alumno, pero no hay que confundir el concepto de recomendar ciertas actividades de forma individual con el de hacer un seguimiento personalizado del alumno. Puede ser que los alumnos sean seguidos de forma personal pero las exigencias sean las mismas para todos los alumnos, especialmente en los

comienzos de una asignatura, cuando no es posible adecuar las actividades en función de las deficiencias de cada alumno o grupo de alumnos porque aún se desconocen. Por ejemplo en [3] se sigue de forma personalizada a cada alumno o grupo de alumno, pero todos desarrollan la misma actividad. Solamente una vez terminada la actividad, el profesor sería capaz de poner de forma personalizada las siguientes actividades en función de las deficiencias observadas en cada alumno.

Algunos autores proponen la recomendación de ciertos ejercicios o actividades diferentes para cada alumno [1] pero sin tener en cuenta cuáles son las deficiencias mostradas por el propio alumno.

Una de las características principales de estos trabajos es que, en ningún caso, están orientados a la consecución de competencias. Pero si es difícil encontrar trabajos sobre recomendaciones en base a competencias, muchos más aún es encontrar herramientas docentes que estén adaptadas para la calificación y recomendación de competencias.

Las principales herramientas docentes que se pueden encontrar hoy en día en las universidades son, por ejemplo, Moodle¹ o Merlot². Estas plataformas ofrecen un gran número de posibilidades como la definición de actividades, la calificación de alumno, foros, etc. Sin embargo, no presentan facilidades para orientar una asignatura en base a competencias y mucho menos recomendar actividades de forma personalizada. Pueden encontrarse algunas herramientas experimentales como la plataforma Ágora, que hagan recomendaciones tanto para alumnos como para los propios profesores [6] pero tampoco manejan el concepto de competencia.

3. Características de Hiperión

La herramienta Hiperión consiste en una aplicación Web desarrollada a partir del framework Liferay³. Liferay es uno de los más famosos gestores de contenidos (Content Management System) que existen en la actualidad. Se caracteriza por una herramienta de código abierto, desarrollado en Java que incorpora una

¹ <http://moodle.org/>

² www.merlot.org

³ www.liferay.com

gran flexibilidad gracias a otros frameworks configurables integrados en el propio sistema como Spring⁴ para manejar la capa de presentación o Hibernate⁵ para la gestión de la capa de persistencia. Incorpora un gran número de portlets que permiten añadir un gran número de funcionalidades de forma rápida al sistema y permite al propio usuario diseñar sus propios portlets en función de sus necesidades.

A continuación describiremos las principales características de nuestra herramienta.

3.1. Roles

Cada funcionalidad de esta herramienta va a asociada a un rol distinto. Entre los roles que pueden desarrollar los usuarios del sistema tenemos: profesores, alumnos y administrativos.

1. Administrativo: El rol de administrativo sería el de aquella persona que básicamente se dedica a rellenar datos relativos a las personas que pueden acceder al sistema, definir sus perfiles de usuario, darles de alta en el sistema y definir las características de las asignaturas que conformarían una carrera o grado. La definición de las asignaturas podría sacarse perfectamente de una guía docente donde aparecieran básicamente las personas que la imparten, las competencias que se van a trabajar y los temas que la componen.
2. Profesor: Los profesores accederían a sus asignaturas con el fin de definir cuáles son los bloques temáticos que las componen, qué competencias se van a trabajar en cada una de ellas y cuáles serían las actividades que van a permitir evaluar a los alumnos.
3. Alumnos: Pueden acceder al sistema para ver tanto las actividades que tienen que realizar como las notas que van obteniendo en cada prueba de cada asignatura de las que estén matriculados.

También cabría destacar como otro posible rol el llevado a cabo por el propio sistema, que de forma automática calcula las notas finales en función de las calificaciones insertadas por cada profesor y además recomienda, de forma automática a cada alumno, las actividades que necesitaría llevar a cabo para poder adquirir las

competencias no adquiridas durante el curso y aprobar así la asignatura.

3.2. Operaciones

La herramienta Hiperión permite una serie de funcionalidades que se resumen a continuación:

1. Alta de usuarios (Profesores y alumnos) y asignaturas.

Las personas con rol de administración recibirían los datos de los profesores, los alumnos y las asignaturas, y podrían meterlos en la aplicación para que estuvieran disponibles.

2. Entrada en el sistema en función del perfil: alumno, administrativo o profesor.

Esta es la función de acceso al sistema. Dependiendo del perfil de cada usuario se podrán realizar una serie de operaciones y otras.

3. Definición de una asignatura

Por ejemplo, en función de la guía docente desarrollada por cada profesor puede definirse cada asignatura.

Para ello es necesario definir qué competencias son las que debe abarcar la asignatura (ver Figura 2). Estas competencias son cargadas a partir de un fichero XML donde se pueden definir todas las competencias que se pueden encontrar en la Memoria de Grado de la escuela o facultad donde se esté trabajando y que lógicamente se conocen a priori.

El formulario muestra dos campos de entrada: 'Nombre:' con un campo de texto rectangular y 'Comentarios:' con un campo de texto rectangular más grande.

Figura 1. Creación de una asignatura

Una vez seleccionadas todas las competencias a trabajar, el siguiente paso consiste en la definición de los bloques temáticos que componen la asignatura, así pues, se permite estructurar la asignatura en bloques para posteriormente asociar a cada bloque ciertas actividades evaluables (ver Figura 3).

⁴ www.springsource.org

⁵ www.hibernate.org

Código	Descripción	Seleccionado
BA1	Capacidad para la resolución de los problemas matemáticos	<input checked="" type="checkbox"/>
BA2	Comprensión y dominio de los conceptos básicos de campos y ondas	<input type="checkbox"/>
CO1	Capacidad para diseñar, desarrollar, seleccionar y evaluar aplicaciones	<input type="checkbox"/>
CO2	Capacidad para planificar, concebir, desplegar y dirigir proyectos, servicios y	<input checked="" type="checkbox"/>

Figura 2. Selección de competencias

The screenshot shows two windows. The left window is a calendar for Sunday, 16/01/11, with a large '16' in the center. The right window is titled 'ECTS' and contains a table with columns 'Código' and 'Descripción'. The table lists B1 Bloque 1, B2 Bloque 2, B3 Bloque 3, and B4 Bloque 4. A mouse cursor is pointing at the 'B1 Bloque 1' row.

Figura 3. Definición de bloques temáticos

Bloques/Competencias	B1	B2	B3	B4
C1	Very low	Very low	Very low	Very low
C2	Low	Very low	Very low	Very low
C3	More less low	Very low	Very low	Very low
C4	High	Very low	Very low	Very low

Figura 4. Asignación de importancias a cada competencia en cada bloque

The screenshot shows a user interface for 'Tema 1'. It features a section titled 'Inicio actividades del bloque 1'. Below this, there is a rating system with 'Your Rating' and 'Average (0 Votes)'. A 'Start Date: 1/22/11' is also visible. There is a 'View in Context' link and a signature 'Asignatura: Tema 1'. Below this, there is a 'pdf' icon and another rating system with 'Your Rating' and 'Average (1 Vote)'. A second signature 'Asignatura: Tema 1' is present at the bottom.

Figura 5. Recomendación y valoración de idoneidad de actividades

No todos los bloques temáticos tienen la misma importancia, de ahí que se puede asignar un grado de importancia mayor o menor a cada uno de ellos. En este caso, los grados de importancia se hacen en base a una serie de etiquetas lingüísticas: importancia alta, importancia media, importancia baja, etc., como se indica en la Figura 4. Estas etiquetas representan una serie de valores configurables que no son el objetivo de este artículo pero que diremos que están basados en la teoría de conjuntos borrosos introducida por Zadeh [8]. Su objetivo es el de evitar pensar únicamente en números como se hacía hasta ahora a la hora de evaluar, y así se podrá indicar en qué grado se alcanzado una competencia: alto, bajo, medio, etc., de forma más natural, ya que decir que la competencia sobre trabajo en grupo tiene un 5.3 no parece ser la forma más adecuada de valorarla, sin embargo, decir que dicha competencia se ha logrado en grado alto parece mucho más intuitivo y entendible tanto por el alumno como por el profesor. De esta manera también se logra que los alumnos se centren más en trabajar una competencia y no en buscar la nota exacta que les hace falta para aprobar, aunque finalmente todas estas etiquetas lingüísticas se transforman en números y permiten calcular la nota final de la asignatura.

4. Definición de actividades

Cada profesor puede crear una actividad, asignársela a un bloque o dejarla para realizar posteriormente, asignarle las competencias que se crean convenientes de ese bloque y ponerle una importancia a cada competencia. Además el profesor podría crear actividades sin necesidad de asignarlas a un bloque concreto o al menos que no tienen por qué ser visibles para los alumnos para poder luego esas recomendárselas a los alumnos que estén suspensos.

Las actividades pueden estar encuadradas en varios tipos: Ejercicios en casa, ejercicios en clase, prácticas, tests, etc. Y tendrían luego pues un enunciado y algunos documentos adjuntos.

5. Realización de actividades

Los alumnos pueden subir archivos, como código, trabajos, tutoriales, etc., para completar las actividades que el profesor les manda. Estas actividades pueden ser online, es

decir, el alumno sube los documentos que el profesor después corregirá u offline, por ejemplo, un examen tipo test que el profesor realiza en clase y que después cuelga las notas correspondientes a cada competencia evaluada.

6. Calificación de actividades

Una vez realizada cada actividad, puede puntuarse cada competencia según el criterio del profesor. En este caso también se puntúan las competencias usando etiquetas lingüísticas para indicar su grado de consecución: alta, baja, media, etc.

7. Visualización de calificaciones

Tanto los alumnos como los profesores pueden ver las notas alcanzadas por cada alumno en cada prueba, la nota final de cada competencia y la nota final del curso. La nota final de cada competencia y de la asignatura es el resultado de agregar las puntuaciones alcanzadas en cada actividad junto con las importancias asignadas a cada bloque y a cada competencia en cada prueba. Este concepto de agregación se basa en las ideas de Yager [7], que tampoco se tratarán en este trabajo dado que no es relevante para el propósito del mismo, y básicamente lo que representan es una media ponderada de las calificaciones de cada competencia.

4. Recomendación de actividades

Una vez finalizadas las clases y evaluadas todas las actividades que estuvieran propuestas, cada alumno puede ver qué competencias tiene adquiridas conforme a una fórmula matemática que tiene en cuenta tanto las calificaciones de cada competencia en cada actividad y la importancia asignada a las competencias en cada actividad o bloque. Así pues tanto el alumno, como el profesor como el sistema conocen cuáles son las debilidades de cada alumno y en qué competencias debe hacer especial hincapié con el fin de poder superar la asignatura.

En este punto es cuando el sistema debe ser capaz de recomendar a cada alumno una serie de actividades que refuercen aquellas menos consolidadas. Para ello es necesario que los profesores de la asignatura, hayan definidos en el sistema una serie actividades que no hayan sido asignadas a ningún bloque en concreto como ya se comentó en la cuarta operación de la subsección 3.2.

```

C: Entrada //Competencias no adquiridas por el alumno
B: Entrada //Competencias adquiridas por el alumno
T: Entrada //Actividades programadas por los profesores
P: Salida // Actividades que debe realizar cada alumno
Leer C, B, T
P ← ∅
Para cada a ∈ A
S ← Competencias_a_cubrir (a)
Si grado (S ∩ C) > α AND grado (S ∩ B) > β entonces
P ← a
Fin_si
Fin_para
Devolver P
    
```

Algoritmo 1. Recomendación de actividades

Uno de los requisitos que deben cumplir las actividades diseñadas por cada profesor es que deben cubrir, en la mayor medida posible, varias competencias, tanto transversales como horizontales, con el fin de que haya una gran variedad de posibilidades en el sistema y así, se puedan asignar las actividades a cada estudiante con la mayor precisión posible. El objetivo de esto es evitar que se asigne a un estudiante una actividad que cubra varias competencias que ya haya superado y apenas una competencia que no tenga adquirido. Lo ideal sería que cada actividad recomendada cubriera el mayor número de competencias no adquiridas posibles, con el fin de obligar al alumno a trabajarlas de forma insistente.

Así pues, es necesario diseñar una batería de actividades muy amplia para que los ejercicios no se repitan entre los alumnos, evitándose así que se copien ejercicios, y permitiendo así una mejor personalización de las recomendaciones hechas. Esto obliga a que el esfuerzo del cuerpo docente sea muy alto para diseñar esta batería de actividades.

Una vez esté diseñada la batería de tareas a recomendar e insertada en el sistema junto con la importancia de cada competencia a trabajar en cada actividad, el siguiente paso es la recomendación de las actividades.

Una vez finalizadas las clases, el profesor podrá darle la orden al sistema para que recomiende a cada alumno actividades que deberán ser entregadas en una fecha determinada para su posterior corrección por parte del profesor. Esta recomendación se hace en base al Algoritmo 1. La idea principal de este algoritmo es, a partir de las competencias tanto adquiridas como no, ver en qué grado se solapan las competencias no adquiridas con las competencias de cada posible actividad recomendable. Si ese grado de

solapamiento es mayor que un umbral α fijado de antemano, entonces se comprueban las demás, también hay competencias que aunque hayan sido adquiridas en un grado bajo, podrían reforzarse siempre que se supere un umbral β . En función de esos dos parámetros, se recomienda una actividad o no.

Una vez que el algoritmo descrito es ejecutado por orden de un profesor, las actividades recomendadas pueden verlas cada alumno en su perfil de usuario. Además, cada alumno, después de realizar cada actividad, puede valorar en una escala de 1 a 5 (ver las estrellas en la Figura 5) si considera que la actividad recomendada ha sido útil para adquirir las competencias que no tenía o por el contrario, ha supuesto un esfuerzo innecesario que no ha dado los frutos esperados.

5. Caso de estudio

Para comprobar la bondad del sistema de recomendaciones presentado se ha propuesto su utilización en la asignatura de Fundamentos de Programación impartida en la Escuela Superior de Informática de Ciudad Real.

La asignatura consta de 3 bloques temáticos: (i) elementos fundamentales de la programación, (ii) sentencias selectivas e iterativas y estructuras de datos básicas, y (iii) modularización y recursividad. Se han diseñado un total de 12 pruebas, 4 por cada bloque temático. Estas pruebas se catalogaban en 4 grupos: tests, ejercicios en clase, ejercicios en grupo y prácticas. A cada prueba se le han asignado una serie de competencias a cubrir de entre las presentadas en la guía docente de la asignatura, como mínimo 3 competencias.

A parte de estas actividades a trabajar durante el periodo de clases, se diseñaron otras 25 pruebas

más, con el fin de recomendar la realización de varias de ellas a los que no lograran ciertas competencias durante el periodo extraordinario. Estas pruebas se catalogaban en: ejercicios en grupo, ejercicios individuales, definición y resolución de ejercicios por el propio alumno, detección de errores y trabajos de investigación.

5.1. Resultados

El grupo de alumnos elegido para trabajar con esta herramienta constaba de 27 estudiantes aunque 2 de ellos no aparecieron por la asignatura en ningún momento. Al finalizar el curso, de entre los alumnos que realizaban regularmente todas las pruebas evaluables, la plataforma Hiperión consideró que 12 de ellos no habían sido capaces de adquirir todas las competencias conforme a las calificaciones puestas por los profesores, por lo que debían realizar más actividades en las que trabajarlas.

En líneas generales, los alumnos suspensos presentaban más de 3 competencias verticales no adquiridas, mientras que las aprobaban la mayoría de las horizontales salvo el conocimiento de idiomas donde no aprobó ninguno.

El sistema recomendó una media de 7 tareas por cada alumno dado que la mayoría no adquirieron casi ninguna competencia. El alumno que menos actividades recibió, tuvo que hacer 4 mientras que el alumno que más tareas recibió, tuvo que hacer 12 tareas, el resto promediaron unas 7 por cada alumno. La reina de las tareas fue la implementación de algoritmos famosos como “quicksort”, que no habían sido vistos en clase, a partir de documentación proporcionada en inglés.

Una de las recomendaciones más conflictiva es la referente a la competencia de trabajo en grupo porque no todos los alumnos son capaces de coordinarse, bien porque algunos dejan la asignatura, bien porque entre ellos mismos no se llevan bien o tienen dificultades por temas de la situación laboral o personal. Así pues, la decisión de crear algunos grupos no fue automática sino que a partir de las recomendaciones hechas por el sistema, se consensuó con los alumnos qué miembros iban a pertenecer a cada grupo. En este caso el número de alumnos fue muy bajo y fue fácil llegar a acuerdo dado que los alumnos posiblemente más problemáticos optaron por dejar la asignatura, y los realmente interesados en

aprobarla pusieron bastante de su parte para poder llevar a cabo las actividades encomendadas.

Una vez llegada la fecha de entrega de los trabajos, los trabajos no sólo fueron corregidos por el profesorado sino que se le obligó a cada alumno a defender varios de ellos en unas sesiones presenciales para garantizar que el trabajo no había sido realizado por otras personas.

A pesar de que la recomendación de los recursos ha sido de forma personalizada, es decir, cada alumno ha recibido tareas para superar sus propios problemas y no simplemente se le ha dicho que se estudie todo para aprobar un examen, los resultados obtenidos por los alumnos tampoco han sido demasiado brillantes. Sólo 3 de los 12 alumnos suspensos aprobaron la asignatura, los 3 que más posibilidades tenían de hacerlo, el resto se dedicaron a presentar trabajos a medias, copiados de Internet o hechos en academias.

Cabe recordar que el sistema Hiperión permite al alumno opinar sobre si considera que la actividad recomendada cubre o no sus deficiencias (ver estrellitas en la Figura 5). Así pues, respecto a la calificación que obtuvieron las actividades, según si eran o no apropiadas para adquirir las competencias que les faltaban, la mayoría de los alumnos coinciden en que sí eran apropiadas, la puntuación media de la recomendación de actividades fue de un 3.6 sobre 5. Sin embargo el profesorado no está totalmente de acuerdo con esta calificación y cree, una vez revisadas las tareas asignadas a cada alumno, que la adecuación de las tareas debe superar el 4 sobre 5, aunque lógicamente todos estos criterios son totalmente subjetivos.

También cabe destacar algunos comentarios realizados por ciertos alumnos, quienes consideraban que las actividades recomendadas no fueron acertadas porque no eran capaces de cuantificar el esfuerzo y el tiempo para realizarlas. Aun siendo cierto que los temas relativos a esfuerzo y tiempo no están controlados por el sistema, todas las tareas estaban programadas para realizarse en periodos de tiempo similares y con esfuerzo parecido, pero lógicamente los alumnos que peores notas tenían, debían hacer un esfuerzo mucho mayor y realizar más actividades que el resto de los alumnos para conseguir las competencias. Es obvio que estos comentarios provienen de los alumnos con peores calificaciones.

6. Conclusiones

El presente trabajo ha presentado una herramienta que pretende cubrir ciertas deficiencias que tiene las herramientas docentes actuales y que surgen como consecuencia de la implantación del modelo de Bolonia.

Las principales ventajas ofrecidas por nuestra aplicación son la definición de las asignaturas en función de las competencias que se vayan a cubrir, pudiéndole asignarle a cada una de ellas un grado de importancia mayor o menor dependiendo del hincapié que se vayan a hacer en ellas a lo largo de la asignatura. A su vez, permite la definición actividades para el alumnado en función de las competencias que se quieran cubrir y la evaluación de dichas actividades. Finalmente, en función de las competencias adquiridas o no a lo largo del curso, permite recomendar de forma individual a cada alumno, una serie de actividades a desarrollar durante el periodo extraordinario de la asignatura, con el fin de trabajar las competencias no adquiridas durante el año.

La herramienta ha sido probada en una asignatura consiguiendo como principales resultados la transmisión del concepto de competencia a los alumnos, lo cual ya era un reto bastante considerable, y una mejor comunicación entre los alumnos y el profesorado, hablando siempre en términos de competencias.

Agradecimientos

Este proyecto está financiado gracias al proyecto FIDELIO (TIN2010-20395, MEC-FEDER), proyecto de excelencia SCAIWEB-2 (PEIC09-0196-3018, JCCM), proyecto PLINIO (POII10-0133-3516, JCCM), proyecto de Innovación Docente HIPERION (013702011, UCLM) y el proyecto HECAOT (TC20112330, UCLM).

Referencias

- [1] Cernuda, A. *Una experiencia de generación personalizada de ejercicios de prueba del software*. En Actas de las XV Jornadas de Enseñanza Universitaria de Informática, Jenui 2007, pp. 189-196, Santiago de Compostela, Julio 2010.
- [2] Estupiñán, E., Ospina, L. E. *Hacia una evaluación personalizada en el sistema educativo colombiano*. Educación y Educadores, 2003, vol 6, pp. 45-66.
- [3] Fiol Roig, G., Fiol Roig, C., Miró Julià, M. *La práctica monitorizada: una herramienta válida en el aprendizaje activo de la asignatura Estructuras de la Información*. En Actas de las XV Jornadas de Enseñanza Universitaria de Informática, Jenui 2007, Teruel, Julio 2007.
- [4] García Hoz, V. *Educación personalizada*. Madrid, Rialp, 1986.
- [5] *REAL DECRETO 1393/2007, de 29 de octubre, por el que se establece la ordenación de las enseñanzas universitarias oficiales*, <http://www.mec.es/>
- [6] Vidal, C., Segura, A., Menéndez, V., Prieto, M. E. *Knowledge-based Architecture for Instructional Engineering*". International Journal of Knowledge and Learning, 2009, vol. 5, pp. 371-388.
- [7] Yager, R.R. On ordered weighted averaging aggregation operators in multicriteria decision making. IEEE transactions on Systems, Man and Cybernetics. 1988, vol. 18(1), pp. 183-190.
- [8] Zadeh, L.A. *The concept of a linguistic variable and its application to approximate reasoning*. Information Sciences. 1975; vol 8(3), pp. 199-24.

Diseño de material específico docente para el aprendizaje de microcontroladores y sistemas USB

Gabriel Jiménez Moreno, Angel Jiménez Fernández,
Manuel Domínguez Morales, Elena Cerezuela Escudero,
Daniel Cascado Caballero, Francisco Gómez Rodríguez,
Manuel Rivas Pérez, Alejandro Linares Barranco

Dpto. Arquitectura y Tecnología de Computadores
Universidad de Sevilla
Escuela Técnica Superior de Ing. Informática, Avda. Reina Mercedes s/n
41012 Sevilla
gaji@atc.us.es

Resumen

En la enseñanza en profundidad y especializada en materias de tipo informática nos encontramos con que los contenidos no paran de crecer y cada vez son más difíciles de abarcar. Por otra parte hay que empezar a pensar que los alumnos deben ser creadores de nuevos productos, y no sólo diseñadores de las ideas de otros, debemos educar no sólo a un profesional diseñador si no también innovador. Para conseguir esto hemos puesto en marcha varios proyectos de innovación docente utilizando la metodología basada en el aprendizaje por proyectos (ApP). En el uso de dicha metodología caben destacar dos ventajas: facilitar al alumno la comprensión y estudio de materias con contenidos cada vez más complejos y numerosos, y entrenarlo en el diseño de sistemas con microcontroladores tal como se hace en la industria. Por otra parte se aportan dos novedades metodológicas: se desarrolla un material que se presta al alumno para que pueda hacer los proyectos en su casa, en las horas no presenciales, con lo que se aprovecha mejor el tiempo para dar los contenidos; y se modifica la metodología ApP para que el alumno sea el que proponga nuevos proyectos que puedan tener un cierto éxito comercial.

Summary

In Specialized and deep teaching about informatics subjects we can find contents that are growing every day, and are more difficult to cover. Otherwise there is needed to start thinking than students should to be creators of new products, and not only designers of existing ideas,

we must educate not only a professional designer but also innovative. To achieve this we have implemented several projects using innovative teaching methodology based on project based learning (PBL). In the use of this methodology should be highlighted two advantages: supply students with the understanding and study of materials with content increasingly complex and numerous, and trains the student in the design of systems with microcontrollers just like as in industry. On the other hand two methodological novelties are contributed: develops a material that is borrow to the student to enable him to do projects at home, in no teaching hours, so time is best used to give content, and PBL methodology is modified for student proposes new projects whose could have some commercial success.

Palabras clave

Aprendizaje por Proyectos, actualización contenidos, enseñar a diseñar, entrenar en innovación, microcontroladores, USB.

1. Introducción y antecedentes

La adaptación del Espacio Europeo de Educación Superior (EEES) y, en particular, la adopción del sistema de crédito europeo (ECTS) están determinando un cambio de modelo educativo. Evolucionamos desde un modelo centrado en la “enseñanza del profesor” a uno basado fundamentalmente en el “aprendizaje del alumno” [1].

Esto ha determinado que prácticamente todas las universidades hayan promovido planes de innovación docentes que faciliten la implantación

de nuevas metodologías por parte del profesorado. En la Universidad de Sevilla, en el curso 2007, se puso en marcha el Plan de Renovación de las Metodologías Docentes 2007 (PRMD) [2]. Este plan estaba dividido en varias líneas, la cinco se dedicada a los proyectos de Innovación Docente, en concreto, a sufragar la implantación de nuevas metodologías docentes. El trabajo que aquí se presenta se enmarca dentro de dicha iniciativa de la Universidad de Sevilla.

La asignatura en la que se centra este trabajo es Tecnología de Microcontroladores (TM), de 3º de Ingeniería Técnica en Informática de Sistemas en la Universidad de Sevilla. Desde la primera convocatoria del PRMD (2007), en esta asignatura, se han ido desarrollando una serie de recursos docentes que sin esa financiación hubieran sido imposibles de llevar a la realidad. El principal de ellos es el de una “placa de entrenamiento”, que los alumnos se pueden llevar a casa a modo de préstamo, y sobre la que gira gran parte de la asignatura.

1.1. Cuando los contenidos no dejan de crecer.

La principal dificultad que tienen alumnos y profesores de las titulaciones con una gran base tecnológica, como son las Ingenierías en Informática, es la velocidad de cambio que experimenta la materia que le es propia de estudio. El profesor para evitar el desfase tecnológico suele preparar nueva materia cada curso, lo que implica que el alumno cada vez tenga que manejar más información, sobre todo debido al aumento de complejidad que supone la “nueva tecnología”, los contenidos de este tipo de materias son fundamentales para mantenerse actualizados en lo que a la tecnología respecta. Al final esta dinámica lleva a una “saturación” del alumno, o a que la docencia que se imparte está muy alejada de la realidad cotidiana que percibe el propio alumno, convirtiéndose para él todo lo que observa en cajas negras incomprensibles, buen ejemplo de ello son los dispositivos USB. Es decir, nos encontramos con la disyuntiva entre disminuir contenidos o pedir que el alumno abarque aun más materia en cada curso. El problema es más grave aun si tenemos en cuenta que la tendencia actual en toda la docencia es dar menos importancia a los contenidos y más a la metodología [3], cuando precisamente lo necesario para no perder el impulso tecnológico debiera ser primar los contenidos innovadores en

las carreras de ingeniería (sin descuidar por ello la metodología docente).

Por otra parte, las empresas de nuestro entorno exigen que los contenidos que se imparten en las universidades se adapten a sus necesidades productivas, como casi ninguna compite realmente en el mercado exterior, el resultado es que sólo necesitan que se enseñe lo que el mercado local demanda. Consideran muchas de estas empresas que no hay que enseñar aquello que ya hacen otros países. En el campo de la informática hay un gran número de empresas que sólo se dedican, y tienen como casi único cliente, a la administración, entidades financieras locales o grandes almacenes. Estas empresas podrán innovar en su ámbito pero difícilmente se convertirán en exportadoras, y menos aun cuando no haya nadie en sus plantillas que conozca lo más básico de la tecnología que nos rodea. La innovación está muy limitada cuando se carece de un conocimiento profundo de la tecnología actual. Un ejemplo de “atraso tecnológico” que se viene produciendo en el campo de la informática es el abandono en los estudios universitarios de ciertos lenguajes de programación a favor de otros más demandados por empresas locales, nos referimos al lenguaje C frente al Java, jamás se debería de haber dejado de impartir C en las escuelas de informática, pero es que además sí se debería de dar Java, este es el típico caso en el que se deberían haber aumentado los contenidos y no de dejar de dar unos frente a otros. Un profesional informático, sin dominar C, no podrá hacer prácticamente nada a nivel de Sistema Operativo y casi nada en sistemas empotrados, pero como en España no tenemos fabricantes de este tipo de cosas, pero sí de páginas web y aplicaciones Java, lo importante es este último y se abandona el primero. Pero es que precisamente esa carencia estructural se perpetúa en cuanto quitamos los contenidos necesarios para que los profesionales tengan algunas ideas nuevas en campos no explotados hasta ahora en nuestro entorno.

En la asignatura de TM se pretende romper con estas tendencias, para ello se actualizan los contenidos aunque esto suponga que el alumno tenga que abarcar más; por el contrario también se intenta, apoyándonos en nuevas metodologías docentes, que al alumno le sea más fácil asimilar esos contenidos en constante crecimiento.

1.2. Los antecedentes

La asignatura TM, tiene como objetivo principal, y entre sus competencias, mostrar cómo se realizan desarrollos con microcontroladores, de forma que el alumno en su futuro profesional pueda utilizarlo en sus diseños. Esta asignatura se encuentra bastante influenciada por los problemas apuntados en el apartado anterior. Desde hace unos años y, sobre todo, a través de un proyecto de renovación de metodologías docentes llevado a cabo en cursos pasados, hemos optado por dar en las clases de teoría los fundamentos de la materia y utilizar las prácticas para adecuarse al progreso tecnológico. El resultado ha sido que las prácticas aumentan cada año en complejidad, teniendo el alumno que dedicar varias sesiones de prácticas para llegar a un resultado que medianamente se aproxime a lo que es usual en cualquier entorno informático e incluso en el hogar. Por ejemplo, una de las materias que se imparte en dicha asignatura, y que ha demostrado a lo largo de estos años que es muy interesante para el futuro profesional del alumno, es el desarrollo de dispositivos basado en USB. Este bus está tan extendido que en principio parecería que es fácil diseñar dispositivos para él, pero la realidad es

que ha llegado a un grado de complejidad que precisa del manejo de una gran cantidad de información y de numerosas “destrezas” para poder diseñar dispositivos USB (en informática “facilidad de uso” no equivale a “facilidad de diseño”, suele ocurrir al contrario). De las 30 horas de laboratorio (para el alumno) de que dispone la asignatura de TM, 15 se dedicaban a la realización de prácticas basadas en dicho bus. Pero el principal problema de estas prácticas radica en que las sesiones de laboratorio son excesivamente cortas para que el alumno llegue a captar o a realizar un diseño tan complejo. Por otra parte, debido a que los laboratorios deben “rotar”, ya que el material es utilizado por numerosos grupos de alumnos, no se pueden realizar prácticas complejas que requieran que dicho material quede “cautivo” (sin que pueda utilizarlo otro grupo).

Para resolver este problema incorporamos en el curso 2007/08 la metodología del Aprendizaje por Proyectos (ApP), esta metodología está ampliamente documentada (tanto en internet como en diversos libros y artículos), de hecho la propia convocatoria del PRMD la propone como prioritaria.

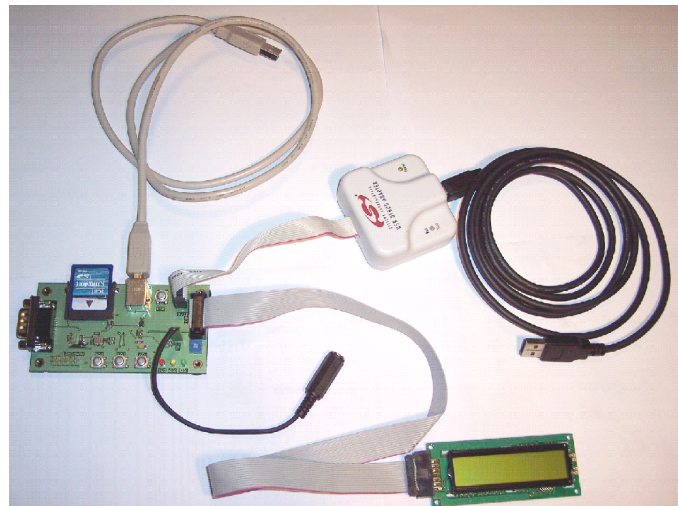
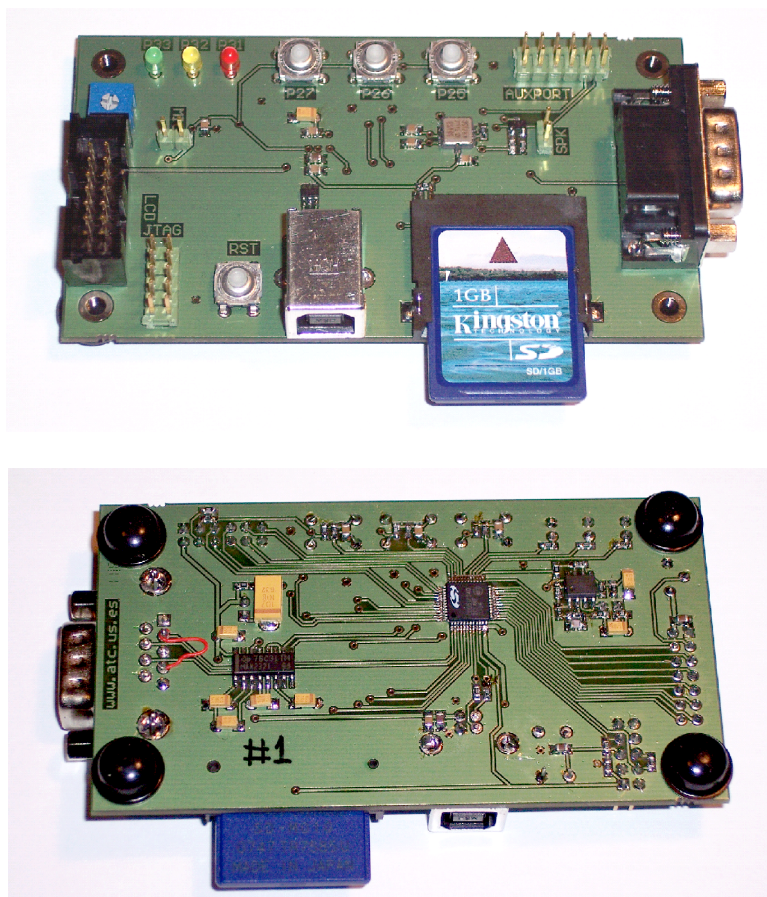


Figura 1: Sistema completo para el ApP diseñado en el curso 2007/08



Figuras 2 y 3: Vista superior e inferior de la placa desarrollada para el ApP, curso 2007/08

En nuestro caso el alumno se lleva el material a casa (mostrado en las fotografías) y desarrolla proyectos con dicho material, dedicando una parte de las horas de laboratorio a explicar los proyectos y a realizar su seguimiento. La principal ventaja de la metodología ApP con respecto a las prácticas tradicionales es la “continuidad”, de forma que el alumno puede desarrollar el trabajo a su ritmo hasta conseguir el objetivo final. El cual, por supuesto, estará más próximo a la realidad que encuentra en su entorno, o en su futuro trabajo profesional, con respecto a lo que se puede hacer en una práctica de laboratorio. Por otra parte, no deseamos hacer que el alumno “trabaje” mucho más de lo que tiene asignado esta asignatura en

créditos ECTS, lo que pretendemos es aprovechar la parte “no presencial” y la de laboratorio, coordinando ambas, para lograr unos objetivos de aprendizaje que realmente sean útiles al alumno y abarquen el máximo contenido posible

1.3. Los contenidos son importantes, pero ¿cómo enseñar a tener nuevas ideas y desarrollar nuevos productos?

La innovación es un vocablo de moda en nuestro país debido a la necesidad de un cambio estructural en el tejido productivo. Como venimos apuntando en este trabajo el estudio en profundidad de la tecnología es fundamental para que la innovación se produzca, y dicho estudio

sólo se puede realizar si en los temarios de las asignaturas se tratan con suficiente profundidad dichas materias como para que podamos, al menos, recrear y fabricar lo que cualquiera encuentra en el mercado tecnológico actual. Si no sabemos hacer desde cero un ratón óptico ¿acaso puedes pretender tener otras ideas, de algo más complejo?

Pero una vez conocida en profundidad la tecnología actual ¿puede surgir la innovación de forma espontánea? ¿se puede enseñar a crear nuevas cosas? Esto parece difícil, ya que es imposible de sistematizar el proceso de creación de una nueva idea. Sin embargo, sí podemos hacer que los alumnos lo intente y propongan nuevos diseños con objeto de llevarlos a la práctica, incluso desarrollar lo que sería un nuevo producto comercial. En este trabajo se muestra una ampliación de la metodología basada en ApP, aplicada a la asignatura de TM, en la que los proyectos son propuestos por los alumnos, de forma que ellos se hacen sus propios dispositivos, premiándose a aquellos con ideas innovadoras. Un posible ejemplo a seguir son algunas asignaturas de Proyectos en el ámbito de la arquitectura, al alumno se le enseñan las técnicas para construir, pero se le entrena en la creación nueva, realmente el arquitecto realiza una obra nueva cada vez que proyecta un edificio, lo cual es un esfuerzo de imaginación que se entrena durante la carrera. Algo parecido es lo que se propone, es subir un escalón más: normalmente el profesor propone el problema y el alumno lo resuelve, aquí hablamos de que el alumno proponga el problema y lo resuelva con la tecnología que conoce, siendo el problema un nuevo producto.

2. Objetivos

Con respecto a la mejora docente en la asignatura de TM confluyen cuatro objetivos de índole diversa:

- Proporcionarle al alumno la oportunidad de aprender en profundidad el diseño con microcontroladores, y eso a pesar de la amplitud de contenidos que requiere la materia. Como se ha comentado se trata de sacar el máximo partido a la parte no presencial que nos viene dada por la incorporación al EEES. Esto prácticamente se

consiguió con la primera convocatoria del PRMD [4].

- Incitar al alumno a que una vez conocida la tecnología desarrolle nuevos productos o proyectos con perspectiva de comercialización (aunque sea de forma virtual). Se trata de que el alumno realice unos miniproyectos fijos propuestos por el profesor, pero que además proponga sobre el papel otros proyectos que sean innovadores y esté dispuesto a realizar. Estos proyectos los puede llevar a la práctica de forma voluntaria el propio alumno con ayuda de los profesores.
- Que los costes de material para el desarrollo de este tipo de experiencia no recaigan directamente en la universidad. Si tenemos en cuenta que además al 40% de los alumnos les hubiera gustado quedarse con el material que se les presta, cabría pensar que hay un cierto número de alumnos dispuestos a gastar parte de su dinero en disponer para ellos de este tipo de tecnología, algunos incluso lo llegan a ver como un entretenimiento además de una inversión en su futuro profesional. Se trata de que el alumno compre por su cuenta los componentes (relativamente baratos) y que utilizando el material de laboratorio ya disponible y, bajo la supervisión y asesoramiento de los profesores, monten su propio sistema. Siendo además premiado el mejor proyecto propuesto por los alumnos con que los componentes son costeados por el propio departamento (a parte de mejorar la nota el alumno).
- Mejorar la confianza del alumno en que puede innovar y desarrollar nuevos productos, ser creativos y que no se trata siempre de hacer o diseñar aquello que ha pensado otro.

3. Metodología

El aprendizaje basado en problemas o proyectos (ABP, también de las siglas en inglés, PBL) está tratado ampliamente en numerosos trabajos [3,4,5]. A pesar de que es una metodología relativamente nueva ha demostrado ser eficaz en la enseñanza de las más diversas ramas de conocimiento [5], pero en particular su aplicación en las carreras técnicas viene demostrando ser especialmente provechosa [6,7].

Cuando se menciona esta metodología en numerosas ocasiones cambiamos proyecto por problema y viceversa, llegando a tratar ambos términos como sinónimos cuando no lo son. En principio la metodología siempre es la misma, que se utilice una u otra denominación creemos que se debe al contexto donde se lleva a cabo y del tamaño del problema/proyecto que se plantea. Dentro de las ingenierías, y siempre que se trate de un trabajo práctico amplio, es probable que se utilice el vocablo proyecto en lugar de problema. Esto entronca además con los clásicos proyectos fin de carrera de las ingenierías o la arquitectura, con los que encontramos una cierta similitud con respecto a la metodología PBL, sobre todo en lo que se refiere a su carácter transversal.

Por otra parte, la materia sobre la que trata este trabajo ha crecido de importancia con los años. Los microcontroladores se han convertido en dispositivos fundamentales en la mayoría de diseños electrónicos. Es por ello que cada vez hay más asignaturas dedicadas a esta materia en todas las universidades. La enseñanza de los microcontroladores debe ser eminentemente práctica. Como en toda materia, hay un núcleo común teórico, pero, si pretendemos formar a un diseñador, es necesario que el alumno realice numerosos proyectos prácticos lo más aproximados a la “realidad industrial”. En este sentido la metodología que sin lugar a dudas se adapta mejor a este tipo de materia es la PBL. Lo cual no ha pasado desapercibido a los educadores; ya en 1997 se utilizaba expresamente esta metodología para la enseñanza de microcontroladores [8].

La implantación del sistema ECTS, con la división de las horas de trabajo del alumno en presenciales y no presenciales, determina que gran parte de la formación del mismo se deba realizar fuera de las aulas y laboratorios. En este caso la aplicación de la metodología PBL en la mayoría de las ocasiones recae sobre la parte no presencial. Esto determina que numerosos docentes opten por proponer proyectos o problemas que no requieran de un material costoso, o simplemente pueda resolverse con un ordenador y conexión a internet. El problema surge cuando se pretende dar una docencia práctica que precisa cierto material específico, la solución menos costosa, y bastante eficaz, es usar simuladores. Pero si lo que queremos es que de verdad el alumno se instruya

con un sistema real podemos proponer dos tipos de soluciones:

1. Habilitar laboratorios como si fueran “salas de estudio”, donde el alumno disponga libremente del material [8].
2. Proporcionar al alumno el material necesario para que desarrolle los proyectos donde quiera.

Normalmente la opción a elegir dependerá del coste final de cada una de ellas. En la mayoría de los casos se opta por la primera opción, y si se elige la segunda, es repercutiendo el coste al alumno. Por ejemplo, en el Georgia Tech (USA) la propia librería de la institución vendía hace años, junto con el libro de la asignatura [9], el entrenador fabricado por Motorola para que el alumno hiciese los ejercicios por su cuenta.

Como es evidente, el principal problema de utilizar PBL para el aprendizaje de los microcontroladores, como en tantos otros casos, es el coste.

Pero precisamente los microcontroladores han llegado a un nivel en su evolución que permite la fabricación de equipos de desarrollo y entrenamiento a un coste bastante reducido. En este sentido no parece descabellado pensar que el alumno pudiera asumir dicho coste, como se ha hecho siempre, por ejemplo, con el material de dibujo. Sin embargo, nuestro entorno no permite tal extremo de forma universal. Es por ello que durante el curso 2007/08 se hizo un sistema de diseño y construcción propia (financiado con la primera convocatoria del PRMD) que pudiera ser prestado al alumno, de forma que su coste no repercutiera directamente en el mismo y, además, pudiera ser reutilizado y, por tanto, amortizado en varios cursos por parte de la Universidad.

El ApP es una metodología fácilmente adaptable para que se entrene el alumno en la innovación y la creación. Una vez realizados algunos miniproyectos que se correspondan con productos comerciales, es fácil que al alumno se le ocurran muchísimas más aplicaciones de la tecnología que estudia. En los microcontroladores esto es especialmente fructífero, a los alumnos se les ocurren ideas que van desde medir velocidades con ratones ópticos (usando el “flujo óptico”) hasta tener mandos a distancia para el televisor con sensores de movimiento (acelerómetros tipo Wii). El problema principal es que no queda tiempo para poder llevar a cabo la mayoría de los

proyectos propuestos por los alumnos. Esto se debe en parte a que seguimos dedicando durante el curso demasiado tiempo a las prácticas tradicionales y a los miniproyectos de PRMD. Lo que se ha hecho es que se les ha dado la oportunidad a algunos alumnos de llevar a cabo su idea como Proyecto Final de Carrera. Ha habido alumnos que se han comprado el material para hacerlo ya que pretenden quedarse después con los resultados materiales del proyecto.

Para el curso 2010/11 se plantea mejorar la carga temporal de forma que de las 30 horas de prácticas presenciales se distribuirán así: 14 serán de prácticas tradicionales, 8 de seguimiento de los miniproyectos propuestos por los profesores (realizados en horas no presenciales) y 8 de estudio y seguimiento de los proyectos propuestos por los propios alumnos (realizados en horas no presenciales).

4. Resultados

El préstamo del material desarrollado bajo la primera convocatoria del PRMD se está llevando a cabo de forma habitual en cada nuevo curso sin incidencias. Hemos podido comprobar que en general resulta una forma eficaz de aumentar la dedicación del alumno a la asignatura y de que capte muchos más contenidos consiguiendo los objetivos de aprendizaje. No ha supuesto un excesivo problema el hecho de que el alumno deba dedicar más tiempo a la asignatura, por ser TM una asignatura optativa y tener alumnos motivados. Una de las claves principales para tener éxito en este tipo de metodologías es que el alumno perciba que el tiempo que dedica a la asignatura es provechoso: les sirve para aprender y aprobar.

De los cuatro cursos que se llevan prestando el material sólo un alumno se ha quedado con él sin devolverlo, habiéndose realizado ya más de 200 préstamos. Además hasta la fecha cinco alumnos han solicitado ayuda y han montado sus propios equipos.

Con respecto a que los alumnos propongan nuevas ideas basadas en lo estudiado en los miniproyectos, esto se empezó a hacer durante el curso 2008/9 de forma parcial, mezclando la asignatura de TM con la de Proyecto Final de Carrera, de manera que los proyectos propuestos por alumnos, considerados viables y de suficiente

envergadura, se han empezado a realizar con objeto de presentarlos como Proyectos Fin de Carrera. En cierta forma, la adopción de proyectos propuestos por los alumnos para la realización del Proyecto Final de Carrera no es nueva, se viene haciendo desde hace años. Y en particular con proyectos basados en microcontroladores con perspectivas empresariales/comerciales (hay proyectos finales de carrera basados en microcontroladores y propuestos por los alumnos de nuestro centro que han tenido un cierto éxito comercial). En la actualidad (2010/11) una parte significativa de TM se basa precisamente en que el alumno proponga nuevas aplicaciones con los microcontroladores. Para ello, los alumnos deben realizar un trabajo durante el curso en el que propondrán sus ideas de nuevos productos, haciendo un estudio aproximado de costes y mercado para dicho producto (no es objeto de esta asignatura este tipo de estudios pero es conveniente, aunque sea de forma testimonial, que el alumno sepa la importancia de estos dos aspectos), finalmente deben realizar un prototipo, aunque sea de forma parcial, si el proyecto es excesivamente ambicioso.

5. Conclusiones

Es muy importante que en las materias que sufren un cambio continuo desde el punto de vista tecnológico se mantengan actualizados los contenidos, aunque ello suponga aumentar temarios. La innovación metodológica debe emplearse para facilitar el trabajo de aprendizaje del alumno, compensando la mayor carga de los contenidos con facilidad a la hora de adquirir los mismos. Si queremos que haya innovación no podemos dedicarnos a entrenar al alumno exclusivamente en lo que demanda el tejido productivo actual de nuestro país, puesto que sus necesidades se basan en una estructura local y obsoleta, en su mayor parte. Es importante que en las materias que se presten a ello por su contenido se empiece a preguntar a los alumnos qué harían con lo que saben, y se incite a la innovación.

Como resultado de nuestra experiencia, podemos concluir que todo lo expuesto aquí hace trabajar a alumnos y profesores mucho más, con lo que sólo alumnos y profesores fuertemente motivados pueden tener éxito en este tipo de esfuerzos. Como estos cambios se realizan por

iniciativa del profesor, es natural que estos estén motivados en principio, el problema surge cuando los alumnos se enfrentan a trabajos y dificultades crecientes, hay siempre un cierto porcentaje del alumnado al que le resulta difícil seguir el ritmo, y que acaba desmotivando al profesorado, al mismo tiempo que retrasa los planes previstos para el resto de alumnos. Para evitar esto, y siendo optativa la asignatura, debe dejarse claro al alumno desde el principio que se le va pedir un gran esfuerzo, esto hace que muchos alumnos acaben desistiendo de matricularse de la asignatura, pero como recompensa los alumnos matriculados suelen ser los verdaderamente interesados. En estos niveles, tan importante es la motivación del alumno como la del profesor, ambos son adultos maduros que han elegido a qué dedican su tiempo.

En este marco podemos comprobar como en TM ha ido disminuyendo el número de alumnos matriculados en los dos últimos cursos, en parte es una tendencia generalizada en toda la universidad, pero en nuestro caso también es consecuencia de concienciar al alumno de lo que le espera si se matricula. Esto favorece que el alumnado de TM esté especialmente motivado e interesado.

Por último cabe destacar que desde la universidad se suele dar mucha importancia a la innovación docente frente a la de contenidos, por ser esta última muy difícil de tener en consideración, siendo además la necesidad de innovación en contenidos limitada a ciertos tipos de carreras y estudios, con lo que se olvida la importancia de la misma para mantener el conocimiento actualizado. Creemos que de la misma forma que hay un plan de renovación de metodologías docentes, en ciertos centros, debería dotarse un plan para la renovación y actualización de contenidos.

Agradecimientos

La experiencia que se muestra en este trabajo ha sido financiada por el proyecto VULCANO (TEC2009-10639-C04-02) y por el Plan de Renovación de las Metodologías Docentes 2007 (PRMD) de la Universidad de Sevilla.

Referencias

- [1] Valero-García, M. Navarro, J. *FAQ sobre la docencia centrada en el aprendizaje*. <http://epsc.upc.edu/~miguel%20valero/contingut3.html>.
- [2] Universidad de Sevilla.(2007) *Plan de Renovación Metodologías Docentes 2007*. <http://www.institucional.us.es/planrenovación>
- [3] Civit, A. Jimenez, G. Diaz, F. y otros (2008). Innovación docente: ¿Metodología o Contenidos? XVI Congreso Universitario de Innovación Educativa en las Enseñanzas Técnicas (CUIEET), Cádiz.
- [4] Jimenez, G. Jimenez, A. Senhadji R. y otros (2008) Aprendizaje por proyectos en la enseñanza de los microcontroladores. XVI Congreso Universitario de Innovación Educativa en las Enseñanzas Técnicas (CUIEET), Cádiz.
- [5] Markham T. (2003) Project Based Learning, a guide to Standard-focused Project based learning for middle and high school teachers. Buck Institute for Education.
- [6] Solomon G. (2003) Project Based Learning: a Primer, Technology a Learning, vol. 23, num. 6.
- [7] Dolamns H. y otros. (2005) Problem-based learning: future challenges for educational practice and research. Blackwell Publishing. Medical Education.
- [8] Montero J.M. y otros. (2004) Herramientas automáticas de apoyo al aprendizaje y evaluación en asignaturas basadas en proyectos. XII Congreso Universitario de Innovación Educativa en las Enseñanzas Técnicas (CUIEET), Barcelona.
- [9] Del Canto Rodrigo P. y otros. (2007) Aprender a programar ordenadores mediante una metodología basada en proyectos. XV Congreso Universitario de Innovación Educativa en las Enseñanzas Técnicas (CUIEET), Valladolid
- [10] Hedley M. Barrie S. (1998) An undergraduate microcontroller systems laboratory. IEEE Transactions on Education, Volume: 41, Issue: 4.
- [11] Peatman J. B. (1998) Design with Microcontrollers, Mcgraw-Hill

Organizadores



Patrocinadores



jenui2011.us.es