

Programación en C con aprendizaje activo, evaluación continua y trabajo en equipo: caso de estudio

Abelardo Pardo, Iria Estévez-Ayres, Pablo Basanta-Val, Damaris Fuentes-Lorenzo

Departamento de Ingeniería Telemática

Universidad Carlos III de Madrid

Avenida Universidad 30

28911, Leganés (Madrid)

{abel, ayres, pbasanta, dfuentes}@it.uc3m.es

Resumen

La adaptación al Espacio Europeo de Educación Superior ha supuesto el rediseño de parte de los planes de estudios por parte de las universidades. En este artículo se describe el proceso de concepción, diseño y despliegue de una asignatura de programación en C que incluye como principales novedades la adopción de una metodología de aprendizaje activo, un esquema de evaluación continua y la inclusión de trabajo en equipo. La asignatura se ha impartido en el curso 2009/2010 a cerca de 200 alumnos. Durante este período se ha obtenido abundante realimentación por parte de los alumnos sobre aquellos aspectos del curso más relevantes (positivos y negativos). Esta información ha permitido comprender más en detalle el proceso de asimilación y la efectividad de los cambios introducidos.

1. Introducción

La adaptación al Espacio Europeo de Educación Superior ha propiciado el rediseño de la gran mayoría de planes de estudios universitarios. En la actualidad, el diseño de una titulación se debe realizar de forma que cuando un alumno se gradúa, haya obtenido un conjunto de competencias que se subdividen en transversales (o genéricas) y técnicas (o específicas). Este cambio en el diseño de una titulación ha propiciado un “giro” en las metodologías docentes y por tanto un replanteamiento de múltiples aspectos.

El tipo de competencias a tener en cuenta al desarrollar una asignatura requiere abandonar metodologías en las que el alumno tenía un papel pasivo y plantear actividades de aprendizaje activo [6] [9]. Este cambio tiene un impacto muy grande en el papel del profesor y los estudiantes en el aula, pero también en el tipo de material que se necesita y el coste de su preparación para la plantilla docente.

La necesidad de seguimiento del trabajo del alumno fuera del aula hace que los esquemas de evaluación basados en un único examen final dejen paso a la adopción de esquemas de evaluación continua con diferentes tipos de pruebas [7]. Pero la evaluación continua requiere un control más exhaustivo de la carga de trabajo, tanto para el alumno como para el profesor, para que su aplicación sea correcta.

La inclusión de competencias transversales que hacen referencia al trabajo en equipo requiere también ajustes muy importantes en una asignatura, tanto más cuanto mayor es el número de estudiantes. Las sesiones de trabajo en equipo necesitan un seguimiento del tutor, con lo que el número de grupos y la gestión de sus sesiones se convierten en aspectos esenciales para ofrecer una asignatura efectiva.

Existen numerosas iniciativas y experiencias piloto en las que estos cambios han sido implementados y sus resultados debidamente documentados [1]. En este artículo se describe el proceso de diseño y despliegue de una asignatura de programación en C que se imparte en el segundo curso de tres titulaciones de grado en el ámbito de la ingeniería de telecomunicación en el curso 2009/2010 en la Universidad Carlos III de Madrid.

Además de los cambios mencionados anteriormente, aparecen aquellos propios de poner en marcha una asignatura con un número elevado de estudiantes, con una parte significativa del alumnado procedente de titulaciones antiguas y en instalaciones que, en algunos casos, no ofrecen el mejor de los entornos para este tipo de metodologías.

El artículo comienza con una descripción en la sección 2 del proceso para obtener los objetivos de aprendizaje a cubrir. A continuación, en la sección 3 se explican las condiciones de contorno impuestas por el contexto en el que se imparte la asignatura. La metodología docente se describe en la sección 4. En la sección 5 se mencionan las herramientas utiliza-

das para la implantación de la metodología docente. La sección 6 describe cómo se organizó el trabajo en equipo para la realización de un proyecto. El artículo concluye con las principales observaciones derivadas del proceso de adaptación y los trabajos futuros que pueden desarrollarse tras esta experiencia.

2. Objetivos del aprendizaje

La metodología utilizada para el diseño de la asignatura ha sido la propuesta por Valero [17] aplicada a dos asignaturas que son afines a la que se pretende diseñar, tanto en temática como en la titulación en la que se imparten. De los cinco pasos que ahí se proponen, los dos primeros han sido los que han supuesto un mayor nivel de dificultad.

En estos dos primeros pasos, la selección de objetivos generales y la caracterización de los objetivos generales, se consultaron múltiples fuentes tales como el ACM Computing Curricula [16] o los criterios de acreditación de ABET (Accreditation Board for Engineering and Technology) [2] [3]. Otras fuentes, como la definición de materias troncales en el BOE o los objetivos generales del plan de estudios, todavía no ofrecían, en la fase de diseño, una descripción adecuada de estos objetivos.

En el tercer paso, la proyección de los objetivos específicos en la asignatura, es donde se enunciaron los cuatro objetivos a cubrir:

1. Diseñar y desarrollar aplicaciones en el lenguaje de programación C. Incluye el uso de estructuras de datos no triviales, gestión de memoria dinámica y operaciones de entrada salida por fichero.
2. Usar herramientas de desarrollo de aplicaciones. Se ofreció un entorno de desarrollo con *Eclipse*¹, el controlador de versiones *Subversion*² y el entorno de compilación cruzada *Scratchbox*³.
3. Aplicar técnicas de trabajo en equipo para desarrollar una aplicación en un dispositivo móvil.
4. Desarrollar técnicas de auto-aprendizaje.

Las aplicaciones diseñadas a lo largo del curso se ejecutan en un dispositivo móvil (concretamente el

Nokia N810). Esta decisión se tomó por dos razones. La primera es que con la utilización de estos dispositivos se expone a los estudiantes por primera vez a los entornos de desarrollo y compilación cruzada a nivel de usuarios. La segunda razón ha sido para aumentar el grado de motivación para participar en los ejercicios prácticos. El poder comprobar cómo los programas que se diseñan realizan una función en un dispositivo real cercano al que ellos poseen hace que crezca el nivel de interés por la asignatura. Esta última observación ha sido corroborada empíricamente por la realimentación obtenida a mitad del curso.

3. Condiciones de contorno

Esta asignatura se ha desplegado en un entorno con múltiples condiciones de contorno que han influido en su estructura. La asignatura se imparte en el primer cuatrimestre del segundo curso y tiene una carga de seis ECTS.

3.1. Normas de la universidad

A nivel institucional se ha establecido que toda asignatura con esta carga tenga dos sesiones semanales de 90 minutos. La asignatura se divide en grupos de un máximo de 40 estudiantes de forma que la primera sesión de la semana se imparte a un grupo denominado “magistral” compuesto por todos los estudiantes de la asignatura hasta un máximo teórico de 120 estudiantes, y la segunda sesión semanal se imparte a cada grupo de 40 estudiantes por separado. Además, se dispone de cuatro sesiones adicionales en el cuatrimestre que se imparten a grupos de hasta 20 estudiantes (cada grupo de 40 se desdobra). La planificación de estas cuatro sesiones se deja a discreción del coordinador de la asignatura.

Respecto al sistema de evaluación, la normativa a nivel institucional establece que el examen final no puede en ningún caso superar el 60 % de la nota de la asignatura. Por tanto, al menos el 40 % de la nota debe ser obtenida a lo largo del cuatrimestre mediante un esquema de evaluación continua. Si un alumno renuncia a esta evaluación continua tiene derecho a presentarse al examen final, pero optando únicamente al 60 % de la nota. Aquellos estudiantes que suspendan la convocatoria ordinaria pueden presentarse a la extraordinaria pero optando únicamente al 75 %

¹www.eclipse.org

²subversion.tigris.org

³www.scratchbox.org

de la nota.

El tiempo de dedicación del alumno a esta asignatura se establece en siete horas y media semanales. Esto incluye las tres horas semanales de clase (magistral y en grupo de hasta 40 estudiantes), dejando las horas restantes para trabajo fuera del aula.

3.2. Perfil de los alumnos

Dado que como objetivo la asignatura incluye el diseño e implementación de aplicaciones en el lenguaje de programación C, para su impartición se asume que los alumnos tienen las nociones básicas de programación obtenidas en las dos asignaturas del primer curso. Sin embargo, tal y como está diseñado el programa de la titulación, la asignatura no tiene ningún tipo de prerrequisito, por lo que cualquier alumno que haya cursado primero el año anterior, independientemente de las notas obtenidas, puede matricularse de ella. Esta situación es una de las que más ha influido en el despliegue de la primera edición. Un alto porcentaje de alumnos (entre el 30 % y el 50 %) se matriculó en ella sin haber aprobado alguno de los cursos de programación de primero.

La asignatura contó con una matrícula inicial de alrededor de 250 alumnos al comienzo del curso. Se estableció a nivel institucional un plazo para anular la matrícula en la asignatura, sin consecuencias académicas, en la semana 8 del cuatrimestre. El número de matriculados descendió hasta los 198 alumnos al final del curso (una reducción, por tanto, del 20 %).

4. Metodología docente

El esquema de evaluación adoptado ha sido tal que la totalidad de la nota final del curso se obtiene durante el cuatrimestre; tras acabar el período lectivo cada alumno tiene su nota final. El examen tan sólo se administra a aquellos estudiantes que hayan decidido no participar en el esquema de evaluación continua.

Para aumentar la motivación de los estudiantes y establecer un vínculo con el futuro mercado laboral, se describe un escenario en el que los alumnos forman parte de un departamento de diseño de aplicaciones de una empresa ficticia. El objetivo de este escenario es reforzar la percepción de que la materia y problemas que se exigen tienen una traducción instantánea en un entorno laboral.

Las sesiones “magistrales” del curso se realizan en un aula convencional, y las de grupo reducido se realizan siempre en aula informática, y por tanto, las prácticas con el ordenador se intercalan con las sesiones más teóricas.

Para cada sesión se publican actividades previas y actividades a realizar en clase. Cada una de ellas está diseñada de forma que se estima el tiempo de dedicación del alumno para su resolución. La descripción de la actividad incluye una lista de recursos (enlaces a documentos auxiliares), un plan de trabajo, un criterio de evaluación y un formulario en el que introducir la dedicación real a dicha actividad (la descripción de las actividades están en formato HTML con un formulario acoplado a un sistema de recogida de información) [14]. El material de la asignatura se encuentra disponible a través de su página de descripción⁴.

Las actividades previas a las sesiones “magistrales” incluyen lecturas y resolución de problemas. Las actividades previas a las sesiones en la sala informática incluyen la preparación de programas para su resolución final en el aula con el tutor.

En la descripción de la asignatura se sugiere a los estudiantes un esquema de trabajo para resolver las actividades previas. Éste consta de dos sesiones de estudio intercaladas por una de consultas a la plantilla docente de las dudas que hayan surgido. En la sección de evaluación de las tareas previas se suelen incluir labores de auto-evaluación o mencionar el uso que de los resultados se va a hacer en la siguiente sesión. Con este esquema se pretende que los estudiantes asistan a clase con el trabajo realizado para poder sacar el máximo partido a las sesiones.

Durante la primera mitad del curso, los estudiantes realizan sus trabajos de forma individual para las sesiones magistrales y en parejas (creadas por ellos mismos) para las sesiones en el aula informática. En la segunda mitad del curso se introduce la metodología de trabajo en equipos de cuatro personas creados por la plantilla docente. En la figura 1 se muestra la organización del curso así como la distribución de puntos de la evaluación. El total suma 110 en lugar de 100 para facilitar la gestión de imprevistos que puedan interferir con la asistencia regular a clase.

Este esquema ofrece a los estudiantes un ritmo de trabajo sostenido a lo largo del cuatrimestre y que

⁴http://www.it.uc3m.es/abel/labas/syllabus_es.html

	Sept.				Dic.
	El lenguaje de programación C	Gestión dinámica de memoria en C	Arquitectura de la plataforma Nokia N810 Maemo	Diseño del proyecto	
Evaluación					
Magistral:	18	13	19	7	
Prácticas:	4	9	3	15	
Total:	22	22	22	22	22 = 110
Prácticas	Parejas fijas elegidas entre alumnos			Equipos de 4 personas creados por los profesores	

Figura 1: Estructura del curso y distribución de la evaluación

combina el trabajo individual con el de la clase teórica y su posterior refuerzo, en la misma semana, con una sesión práctica. A cada alumno se le han realizado a lo largo del curso un total de 17 pruebas de diferente tipo para obtener su nota final.

Uno de los principales riesgos de este esquema es la sobrecarga de trabajo. Actividades previas que requieran un tiempo mucho más elevado del planificado pueden originar conflictos con el resto de asignaturas en el mismo curso, además de un rendimiento mucho menor del esperado por parte de los alumnos.

Este problema no tiene fácil solución, y su dificultad se incrementa sensiblemente por la ausencia de prerequisites. Aunque el sistema de obtención de medidas de la dedicación de los estudiantes era utilizado con frecuencia, la verdadera medida de la carga de trabajo se obtuvo mediante cuestionarios realizados a mitad de cuatrimestre [15].

Del esquema de evaluación continua también se deriva un incremento muy importante de las labores de docencia. Si el trabajo del alumno ahora incluye las horas que le dedica a la asignatura fuera del aula, creemos que se debe hacer un esfuerzo por cuantificar también este trabajo para la plantilla docente.

5. Herramientas de desarrollo

Como consecuencia de incluir en los objetivos el desarrollo de aplicaciones en dispositivos móviles, se precisaba un entorno de compilación cruzada. Estos entornos suelen tener unos requisitos estrictos en lo referente al sistema operativo y las versiones de aplicaciones auxiliares. Por ésto, y para simplifi-

car las tareas de administración, se optó por crear y distribuir una máquina virtual⁵ con este entorno ya preinstalado y configurado. La herramienta utilizada para la compilación cruzada ha sido *Scratchbox* sobre el sistema operativo Linux. También se incluyó una versión del entorno integrado de desarrollo *Eclipse*.

El uso de máquinas virtuales ha tenido múltiples ramificaciones. La principal ventaja ha sido la de realizar las labores de configuración del entorno en el momento de la creación de la máquina. Una vez verificada la instalación, se procedió a su distribución entre los estudiantes. Además, los alumnos en su mayoría disponen de ordenadores personales pero con sistemas operativos diferentes a Linux. La máquina virtual ofreció una alternativa sin apenas umbral de adopción para realizar todos los trabajos relativos a la asignatura.

La principal desventaja de esta máquina virtual apareció cuando se combinó su uso con las actividades en las salas informáticas. Dado el número de estudiantes que podían utilizar el entorno de forma simultánea se optó por instalar una máquina en cada cuenta de estudiante. Pero esto, combinado con el esquema de directorios compartidos vía sistema de ficheros distribuido supuso una carga excesiva en los servidores que se tradujo en numerosas anomalías en las primeras sesiones en el aula informática, aunque estas deficiencias acabaron solventándose.

Además del entorno de desarrollo, el intercambio de documentos de la actividad en el aula se realizó utilizando el controlador de versiones *Subversion*.

⁵Creada con VirtualBox www.virtualbox.org

En las primeras actividades del curso se incluyó información sobre los principales comandos y cada grupo de trabajo dispuso a lo largo del curso de dos depósitos (uno para las prácticas en pareja y otro para el trabajo en equipo) en los que se subían los ficheros a entregar.

El despliegue de estos depósitos facilitó mucho el intercambio de información entre alumnos y profesores, agilizando la resolución de dudas y preguntas sobre los trabajos.

6. Trabajo en equipo y proyecto

El trabajo en equipo se introduce en la segunda mitad del curso. La primera sesión se dedica a describir qué se entiende por trabajo en equipo y a enfatizar la dificultad de conseguir un rendimiento elevado. Además, se incluye como parte de las lecturas previas un artículo sobre posibles conflictos [13] y en clase se planifica una actividad para analizar cómo se pueden solventar estos conflictos [8].

El objetivo de estas actividades es doble. El primero poner de relevancia lo importante y a la vez escurridizo que es el objetivo de trabajar en equipo de forma efectiva. El segundo es anticipar los posibles conflictos que aparecerán conforme se avanza en esta fase y reducir el número de situaciones extremas que requieren la intervención de la plantilla docente.

Este período de trabajo en equipo coincide con el diseño e implementación del proyecto. Su enunciado se sitúa en el escenario de la asignatura descrito en la sección 4. Durante un periodo de siete semanas cada equipo de cuatro o cinco alumnos debe cubrir los hitos y realizar las entregas que se estipulan. Se exige a los equipos que se reúnan al menos una vez al margen del trabajo en clase y que se levante acta de esa reunión.

En las semanas que dura el proyecto, las sesiones magistrales continúan cubriendo aspectos más avanzados de la materia apoyados en ejercicios, y las sesiones en la sala informática se dedican a supervisar el progreso de los equipos, comprobar que se están cubriendo los hitos, comprobar los entregables (si procede) y revisar el funcionamiento del equipo.

La formación de los equipos ha sido uno de los aspectos que más debate ha generado entre la plantilla docente. La primera idea fue la de distribuir a los es-

tudiantes en grupos con rendimientos dispares (para así replicar un posible escenario en la vida real). De esta forma, un estudiante avanzado tendrá por compañero a uno rezagado y se espera que el primero ayude al segundo a mejorar su dominio de la asignatura.

Sin embargo, la experiencia de las primeras semanas hizo patente el gran número de estudiantes cursando la asignatura sin haber aprobado las dos asignaturas de programación anteriores a ésta. Por este motivo, se optó por un segundo criterio en el que los estudiantes simplemente se ordenan por las notas obtenidas en el proceso de evaluación continua, y se agrupan aquéllos con notas similares. La ventaja de este esquema es que en este entorno, aquellos equipos que tengan miembros con mejores notas necesitan una supervisión más baja y el tiempo extra se puede dedicar a la supervisión del trabajo en el resto de equipos [18].

7. Seguimiento

Un curso en el que se han desplegado tantos cambios respecto a la metodología docente convencional necesita un sistema de seguimiento detallado y exhaustivo. Por un lado, los formularios de evaluación de la actividad docente tienen una latencia demasiado elevada. La realimentación se obtiene a final del curso y los cambios no se despliegan hasta la edición siguiente. Por contra, si se piensa en obtener realimentación de los estudiantes de forma más frecuente, la carga de trabajo no permite introducir largos formularios explorando cada posible aspecto del curso, o con demasiada frecuencia, como en [4].

El enfoque que se ha adoptado sigue un esquema de “realimentación rápida” y preguntas cortas que se plantea en las semanas 5, 10 y 14 (y última) del cuatrimestre [5][10], donde participaron 67, 44 y 43 alumnos respectivamente. El formulario solicita que se describa el aspecto crítico más positivo (negativo) de lo que va de curso (o desde el último formulario). Las respuestas se limitan a 300 letras para forzar a los estudiantes a pensar detenidamente sobre uno de los múltiples aspectos y expresarlo de forma concisa [15].

Los aspectos más relevantes mencionados como negativos son:

1. *Tiempo de dedicación a la asignatura.* En la

primera tanda de preguntas (semana 5), este aspecto apareció de forma abrumadora como el más negativo (29 alumnos lo categorizaron así, frente a uno que lo situó como aspecto positivo). Este resultado fue igualmente corroborado mediante una entrevista con los representantes de los alumnos. Se adoptaron medidas para corregir el problema y en las dos ocasiones restantes se alcanzó el equilibrio entre las valoraciones positivas y negativas.

2. *Incremento de la participación del alumno en las actividades de clase.* Este aspecto recibió un número elevado tanto de menciones negativas (24) como positivas (14) en la primera encuesta. En cambio, en las siguientes encuestas la frecuencia de aparición de este comentario se redujo sustancialmente. Cabe extraer, por tanto, la conclusión de que el cambio a la metodología de aprendizaje activo requiere un período de adaptación por parte del alumno.
3. *Evaluación continua.* Al igual que en el caso anterior, este aspecto recibió un número elevado de menciones positivas (13) y negativas (10). Los estudiantes que lo perciben como positivo mencionan lo fácil que es llevar la asignatura al día. Los que lo incluyen como aspecto negativo mencionan la falta de *flexibilidad* para llevar la materia a su propia velocidad. La conclusión que se puede extraer es que este aspecto del curso es, cuando menos, polémico. Dada la variedad de perfiles de aprendizaje, creemos que siempre habrá estudiantes que prefieran un esquema que les obligue a llevar la asignatura al día y otros que prefieran que esto se deje como decisión personal y no impuesta.

Estas observaciones coinciden de forma muy significativa con el proceso de adaptación que necesitan los alumnos para este tipo de metodología descrito por Felder [11]. El problema de sobredimensionamiento detectado en las actividades tiene un análisis complejo. La presencia de estudiantes sin apenas conocimientos de programación, al no haber prerequisites de ningún tipo, hace que sea difícil de estimar el tiempo que un estudiante medio le debe dedicar a una tarea.

Los aspectos más relevantes mencionados como positivos incluyen también dos de los anteriores: el incremento de la participación del alumno en las ac-

tividades de clase y el esquema de evaluación continua. Además de estos, también se incluyen:

1. La organización del material de la asignatura.
2. El trabajo en equipo. Los estudiantes apreciaron la necesidad de esta técnica de trabajo y la identificaron como muy importante. El criterio de creación de los equipos no se menciona ni como positivo ni como negativo.
3. El proyecto final del curso. Se percibe como el colofón necesario para entender la asignatura en su conjunto así como la oportunidad de desarrollar una aplicación a ejecutar en un dispositivo real. En la última encuesta obtuvo 14 calificaciones positivas frente a cero negativas.

Estos dos últimos resultados corroboran la buena acogida que tuvo la estructura de la segunda mitad del curso en la que el hilo conductor es el proyecto y la forma de trabajo es en equipos diseñados por la plantilla docente.

Este proceso de realimentación rápida ha sido crucial para tener una información fiable de lo que está pasando en el despliegue de un curso. El número de diferentes aspectos susceptibles de ser estudiados es tan grande, que un tratamiento mediante estas dos preguntas abiertas se traduce en un alto grado de flexibilidad. Creemos sinceramente que la profundidad y justificación de los ajustes realizados no habría pasado si se tienen en cuenta únicamente las encuestas de evaluación docente que se realizan al final del cuatrimestre, basadas sobre todo en la evaluación al profesor [12].

8. Conclusiones y trabajo futuro

Se ha expuesto la concepción, diseño y despliegue de una asignatura de segundo curso en las titulaciones en el ámbito de la ingeniería de telecomunicación en el curso 2009/2010. En ella se han introducido elementos que suponen un cambio importante en la metodología tradicional: aprendizaje activo, trabajo en equipo y evaluación continua (entre otros).

Con este nuevo enfoque se ha conseguido la introducción del aprendizaje activo por parte del alumno con una considerable tasa de éxito. Además, el alumno no sólo adquiere técnicas de autoaprendizaje, sino que consigue destreza y habilidades en un entorno de trabajo colaborativo. Finalmente, se ha desarrollado una técnica de evaluación con-

tinua de la metodología, basada en formularios de realimentación rápida, crucial en el despliegue del curso y que ha resultado mucho más eficaz que las que se realizan al final del cuatrimestre.

Los resultados obtenidos nos permiten alcanzar conclusiones adicionales. La primera es que cuando se tiene en cuenta el tiempo de dedicación del alumno, una medida fiable de ésta es algo extremadamente difícil de obtener. En segundo lugar, las metodologías más participativas para el estudiante encuentran un cierto rechazo al comenzar. Algunos alumnos se sienten incómodos en aquellas clases en las que se les cede parte del protagonismo y reclaman el entorno magistral pasivo convencional. Una última conclusión es que tras un período transitorio, el nivel de aceptación de estas metodologías es muy alto.

El trabajo futuro en esta asignatura está en su comienzo. Al ser la primera edición, se pretende formalizar el proceso de recopilación de sugerencias de mejora e incorporación de medidas al esquema del curso. Este formalismo realmente garantiza que la calidad del curso se incrementa tras cada edición hasta alcanzar niveles de efectividad elevados. En lo referente al proceso de diseño de la asignatura, se requiere una revisión de los objetivos de aprendizaje para que reflejen de forma más ajustada lo que realmente se trabaja en las actividades de la asignatura. Finalmente, se considerarán medidas para reducir la carga de trabajo del personal docente (más pruebas de auto-evaluación, por ejemplo).

Agradecimientos

El equipo docente agradece al profesor Miguel Valero sus múltiples consejos y sugerencias para el diseño y despliegue de la asignatura. Este trabajo ha sido financiado en parte por el proyecto Emadrid: Investigación y desarrollo de tecnologías para el e-learning en la Comunidad de Madrid (Ref. S2009/TIC-1650), el proyecto Learn3 del Plan Nacional de I+D+I (Ref. TIN2008-05163/TSI) y el proyecto Flexo del Plan Nacional de Investigación Científica, Desarrollo e Innovación Tecnológica (Ref. TSI-020301-2008-19)

Referencias

- [1] Experiencias piloto en las universidades. <http://eees.universia.es>, Accedido 2010/2/12.
- [2] Criteria for accrediting computing programs. Technical report, ABET Accreditation Board for Engineering and Technology, 2007.
- [3] Criteria for accrediting engineering programs. Technical report, ABET Accreditation Board for Engineering and Technology, 2007.
- [4] Thomas A. Angelo y K. Patricia Cross. *Classroom Assessment Techniques: A Handbook for College Teachers (Jossey Bass Higher and Adult Education Series)*. Jossey-Bass, 03-12 1993.
- [5] George R. Bateman y Harry V. Roberts. *Tqm for professors and students.*, 1993.
- [6] Arthur W. Chickering y Stephen C. Ehrmann. *Implementing the seven principles: Technology as lever*, 1996.
- [7] José Luis Pérez de la Cruz y Eva Millán. Estudio comparativo de diversos métodos de evaluación. En *Actas de las Jornadas de Enseñanza Universitaria de la Informática*, 2009.
- [8] Pablo del Canto Rodrigo, María Isabel Gallego Fernández, José Manuel López Canalda, Francisco Javier Mora Serrano, María Angélica Reyes Muñoz, Eva Rodríguez Luna, Joseph Kanapathipillai Sanjeevan, Eduard Santamaria Barnadas y Miguel Valero García. Conflictos en el trabajo en grupo: Dos casos representativos. En *Actas de las Jornadas de Enseñanza Universitaria de la Informática*, 2009.
- [9] Iria Estévez-Ayres, Pablo Basanta-Val y Marisol García-Valls. Docencia de programación concurrente. experiencias en laboratorio. En *Actas de las Jornadas de Tiempo Real*, pages 65–69, 2004.
- [10] Richard Felder. What do they know anyway? *Chemical Engineering Education*, 26(3):134–135, 1992.
- [11] Richard Felder. We never said it would be easy. *Chemical Engineering Education*, 29(1):32–33, 1995.
- [12] James Lang. Did you learn anything? *The Chronicle of Higher Education*, 53(27):C1, 2007.
- [13] Barbara Oakley. It takes two to tango: How 'Good' students enable problematic behavior

- in teams. *Journal of Student Centered Learning*, 1(1):19–27, 2002.
- [14] Abelardo Pardo. A multi-agent platform for automatic assignment management. En *ITiC-SE '02: Proceedings of the 7th annual conference on Innovation and technology in computer science education*, pages 60–64, New York, NY, USA, 2002. ACM.
- [15] Abelardo Pardo, Iria Estévez-Ayres, Pablo Basanta-Val y Damaris Fuentes-Lorenzo. A process for improving course quality based on mid-semester feedback. En *1st International Conference on Technology Enhanced Learning, Reforming Education and Quality of Teaching: Learning Technologies, Quality of Education, Educational Systems, Evaluation, Pedagogies*, Athens, Greece, May 2010.
- [16] Russell Shackelford et al. The computing curricula 2005. The overview report. Technical report, ACM and IEEE Computer Society, 2005.
- [17] Miguel Valero-García. Proyecto docente para las asignaturas introducción a los computadores y laboratorio de programación, 2003.
- [18] Miguel Valero-García. Comunicación personal, 2009.