

Evaluaciones y rúbricas en el aprendizaje de la programación de ordenadores

EVALUACIONES Y RÚBRICAS EN EL APRENDIZAJE DE LA PROGRAMACIÓN DE ORDENADORES

Santiago Tapia Fernández
Universidad Politécnica de Madrid
stapia@etsii.upm.es
Ángel García Beltrán
Universidad Politécnica de Madrid
agarcia@etsii.upm.es

Palabras clave:

De 3 a 5 palabras clave

Resumen:

Este trabajo compara la aplicación de dos metodologías docentes y dos instrumentos de evaluación en dos grupos de la misma asignatura. Uno de los grupos ha seguido una metodología docente a distancia apoyada en una plataforma de e-learning, con contenidos docentes en formato electrónico, tutorías virtuales, un foro de comunicaciones, ejercicios de auto-evaluación, trabajos individuales y trabajos en grupo. El otro grupo comparte los mismos contenidos docentes pero sigue una metodología que se basa en la impartición de clases presenciales de tipo magistral y clases prácticas de resolución de problemas y en la que se realiza un proyecto en equipo que se entrega en tres hitos. En ambos casos se ha utilizado una rúbrica por parte del profesor para evaluar los trabajos y otra por parte del alumno para valorar la actividad y competencia trabajo en equipo. Además se ha realizado un examen tradicional aunque sin efecto en la calificación final del alumno como referencia de comparación. En este sentido se muestra la experiencia en el empleo de la rúbrica en el contexto de la docencia en Ingeniería, y, en este caso, respecto de un instrumento tradicional como es una prueba de evaluación escrita.

1. INTRODUCCIÓN

En la mayoría de las titulaciones actuales de ingeniería es habitual encontrar asignaturas relacionadas con el aprendizaje de lenguajes de programación ya que su uso y programación se considera una competencia imprescindible para el futuro profesional de un ingeniero de casi cualquier área de conocimiento. En este sentido las asignaturas que se ocupan de la capacidad de desarrollar programas de ordenador presentan algunas particularidades que las hacen especialmente interesantes a la hora de aplicar instrumentos de evaluación no tradicionales. Estas características son: (a) el resultado del aprendizaje es estrictamente una competencia, (b) el ejercicio profesional de esta competencia se realiza usualmente en equipo y (c) el proceso de evaluación puede ser más objetivo que en otras materias. Como consecuencia de la aplicación del EEES se están aplicando nuevas metodologías docentes y de estrategias de evaluación a estas asignaturas y resulta relevante conocer qué diferencias (en su caso), introducen en el proceso de aprendizaje de este tipo de materias.

Hace años la metodología empleada para la enseñanza de materias ingenieriles y en particular informáticas era bastante tradicional y se basaban fundamentalmente en la impartición de clases presenciales, el seguimiento de manuales y libros impresos y, en el mejor de los casos, la realización de prácticas de laboratorio con ordenadores [1]. En la actualidad se emplean muchos otros recursos y herramientas tecnológicas como alternativa o apoyo a los métodos docentes clásicos, lo que también puede extenderse a los sistemas de evaluación de las asignaturas. La reestructuración de los nuevos Planes de Estudios dentro del Espacio Europeo de Educación Superior (EEES) está, por un lado, reduciendo y, por otro lado, obligando a modificar las metodologías de enseñanza-aprendizaje orientándolas al fomento del trabajo autónomo del alumno sin menoscabo de reforzar las tareas de seguimiento del alumno por parte del profesor. Las nuevas tecnologías y los sistemas de aprendizaje interactivos pueden ayudar a los alumnos a aprender de forma más cómoda y flexible y facilitar a los docentes la realización de actividades de difícil implementación especialmente con grupos numerosos. En este trabajo se analizan las metodologías, tecnologías de apoyo, sistemas de rúbricas e instrumentos de evaluación empleados en la impartición de una asignatura de introducción a la programación con Java en la Escuela Técnica Superior de Ingenieros Industriales de la Universidad Politécnica de Madrid (ETSII-UPM). La asignatura se presenta en la primera sección y en las siguientes se examinan cada una de las actividades e instrumentos de evaluación implantados indicando su objetivo, su ejecución y los resultados obtenidos.

2. LA ASIGNATURA DE PROGRAMACIÓN EN JAVA

Programación en Java es una asignatura de libre elección cuatrimestral perteneciente al Plan de Estudios de Ingeniería Industrial (Plan 2000) que se imparte en la ETSII-UPM desde el curso 2002-03 en principio sólo de forma presencial y con una carga académica de 3 ECTS. La asignatura es impartida por un único profesor y tiene una matriculación de unos 25 alumnos de media por curso.

2.1. CARACTERÍSTICAS GENERALES

Su objetivo es que los alumnos, que previamente han superado, al menos, una asignatura de introducción a la programación en TurboPascal o C, adquieran conocimientos de elementos básicos del lenguaje Java y de la metodología de la programación orientada a objetos.

El profesor emplea la plataforma de tele-formación AulaWeb [2] de la ETSII-UPM para facilitar a los alumnos documentación del curso en formato electrónico. En esta plataforma se han publicado un tutorial de programación en Java [3], varios listados de enunciados de ejercicios y prácticas y las presentaciones empleadas en las clases presenciales. La documentación y contenidos del curso también pueden encontrarse en el portal del proyecto OpenCourseWare de la UPM [4]. La plataforma AulaWeb permite además la realización de otras actividades docentes de apoyo como la autoevaluación y la entrega de trabajos [5].

Como se especifica en el proyecto de organización docente de la ETSII-UPM, al tratarse de una asignatura de libre elección no existe un examen tradicional, presencial y por escrito, al final del periodo docente para el cálculo de la calificación final de cada alumno. Por este motivo se implementan otras actividades a la hora de calificar a los alumnos que tienen en cuenta los tres principales puntos a considerar en el Proceso de Bolonia: (a) todo lo que implique un esfuerzo por parte del alumno debería ser medido, (b) la retroalimentación del alumno debería ser continua y (c) la monitorización de las actividades debería contemplar la evolución de la asignatura [6].

Desde el curso 2005-06 la asignatura se imparte en dos modalidades: la primera sigue una metodología presencial (con código 9013) y la segunda tiene una metodología no presencial o telemática (código 9122). Hasta el curso 2009-10 las asignaturas se ubicaron en el primer cuatrimestre (septiembre-enero) del periodo académico y a partir de ese curso se dispusieron en el segundo cuatrimestre (febrero-junio).

Desde el curso 2010-11 se lleva a cabo una prueba final de conocimientos adquiridos mediante un examen tradicional de control que no se considera para

el cálculo de la calificación final y que sólo sirve como referencia comparativa con los resultados obtenidos con otros instrumentos de evaluación en otras actividades y otras ediciones del mismo curso. Además, las preguntas que componen la prueba son las mismas año a año para asegurar la uniformidad de los resultados.

En las siguientes secciones se describen las características específicas de cada modalidad docente.

2.2. METODOLOGÍA PRESENCIAL

Desde el curso 2002-03 la asignatura 9013 – Programación en Java presencial tiene asignadas dos horas semanales de clase y otras tantas de tutorías presenciales durante las 14 semanas reales del cuatrimestre y ha tenido tres profesores diferentes que han compartido básicamente la misma sistemática. Las clases se imparten en un aula en el que cada alumno dispone de un ordenador personal con el entorno de programación de Java. El profesor también dispone de un ordenador conectado a un cañón de proyección lo que le facilita mostrar una presentación, el código fuente de un programa o la interacción del programa con el usuario al ejecutarse durante la clase. La metodología de las clases es eminentemente práctica ya que tras una breve explicación de los contenidos teóricos se pasa inmediatamente a la puesta en práctica de los mismos mediante ejercicios de programación con el ordenador que se van implementando y corrigiendo conforme avanza la clase.

La normativa de asignaturas de libre elección de la ETSII-UPM establece como condición necesaria y suficiente para aprobar la asistencia, al menos, al 80% de las clases en las asignaturas presenciales. Hasta el curso 2010-11 se establece la posibilidad de realización y entrega de un trabajo individual de programación. Esta actividad es de carácter voluntario pero permite establecer diferentes calificaciones por encima del aprobado para los alumnos que la lleven a cabo.

La realización de trabajos individuales se sustituye por un trabajo en grupo en el curso 2011-12. A principios de este curso se les comunica a los alumnos que se pueden organizar en grupos de trabajos de entre 3 y 4 alumnos para la entrega de un trabajo de programación a final del curso. Los miembros de cada equipo de trabajo deben defender el trabajo al final del cuatrimestre delante del profesor y la calificación del trabajo es compartida por todos los miembros del mismo grupo. El profesor se ayuda de una rúbrica para la evaluación de los trabajos.

2.3. METODOLOGÍA A DISTANCIA

Desde sus inicios en el curso 2005-06 la asignatura 9122 – Programación en

Java se imparte completamente a distancia y ha tenido siempre el mismo profesor. Al no tener clases presenciales, el profesor establece la comunicación con los alumnos fundamentalmente a través del servicio de correo electrónico y de la plataforma de tele-educación AulaWeb. El alumno sólo necesita un ordenador con conexión a internet, un navegador web y el Kit de Desarrollo Java en su ordenador para seguir todas las actividades del curso.

Los alumnos tienen a su disposición la planificación de todas las actividades de la asignatura desde el inicio del curso. Para garantizar un esfuerzo continuado en el aprendizaje el profesor informa todas las semanas por correo electrónico de la publicación de los contenidos del temario y de la configuración de ejercicios de autoevaluación y de programación en la plataforma AulaWeb que los alumnos tienen que entregar individualmente con un plazo límite de, típicamente, una semana.

Otra actividad característica es el planteamiento de un trabajo final en grupos cuya constitución corre a cargo del profesor y en el que se permite que cada equipo elija el alcance de su desarrollo. Desde la primera edición todos los miembros de cada grupo deben presentar el trabajo final en la única actividad presencial obligatoria de la asignatura en una fecha al final del cuatrimestre determinada de acuerdo con el profesor.

Además en el curso 2011-12 se han programado otros dos pequeños ejercicios en grupo (para la semana 5 y la semana 9 del cuatrimestre) con objetivos muy concretos para que los alumnos puedan ir tomando experiencia en el trabajo colaborativo.

Tanto para los ejercicios individuales como para los proyectos en grupo el profesor se ayuda de una rúbrica para la evaluación de los trabajos.

Debido a la no existencia de clases presenciales la participación del alumno en la realización del trabajo final se considera condición necesaria para el aprobado de la asignatura. Para la calificación final de cada alumno en la asignatura se considera a partes iguales el trabajo personal (autoevaluación y ejercicios individuales) y el trabajo colaborativo (ejercicios en grupo y trabajo final).

3. RECURSOS E INSTRUMENTOS DE EVALUACIÓN

En la Tabla I se resumen los diferentes recursos, tecnologías, herramientas y actividades docentes que se emplean en cada una de las asignaturas y que se describen en las siguientes secciones.

TABLA I
Recursos y actividades implementadas

	9013 (Presencial)	9122 (A distancia)
Documentación online	PDF+OCW	PDF+OCW+SCORM
Autoevaluación (AulaWeb)	Sólo para control de asistencia	Evaluación de contenidos (x 10)
Trabajos individuales	Uno hasta 2010-11 (para calif. final)	Entre 5 y 10 (dep. del curso)
Trabajos en grupo	Uno en 2011-12 (para calif. final)	Entre 1 y 3 (dep. del curso)
Prueba final de conocimientos	Desde 2010-11	Desde 2010-11
Entorno de programación	Java SDK (hasta 2010-11) y NetBeans	Java SDK

3.1. CONTENIDOS TEÓRICO-PRÁCTICOS

Los contenidos de la asignatura se organizan en documentos teóricos y prácticos en distintos formatos. La parte teórica explica la especificación del lenguaje Java para cada uno de los capítulos del temario. La parte práctica se estructura de la misma forma e incluye ejercicios y problemas resueltos así como el código fuente de programas que pueden ser editados, modificados, compilados y ejecutados por los alumnos con el entorno de programación de Java [7]. Los problemas propuestos están dispuestos en orden creciente de dificultad. Además de la documentación en formato PDF disponible en AulaWeb y de los contenidos del curso en la web de la OpenCourseWare de la UPM se generó una versión SCORM [8] de los contenidos en inglés que también fue implementada en la plataforma AulaWeb [9]. Entre las razones de utilizar este formato se basa la portabilidad y, en este caso, en la necesidad de seguimiento de la actividad de los alumnos, especialmente en la asignatura de formato a distancia. Finalmente se dispone de un manual impreso actualizado con todo el contenido del curso que los alumnos pueden pedir prestado en la biblioteca de la ETSII-UPM o adquirir en la Sección de Publicaciones [10]. Como se ha indicado previamente todo el material anterior se pone a disposición de los alumnos independiente de la asignatura que cursen.

3.2. ENTORNO DE TRABAJO

Desde el inicio de la impartición de la asignatura presencial el entorno de trabajo, es decir, las herramientas que utilizan los alumnos para desarrollar los programas han sido un sencillo editor de texto y el compilador en línea del Software Development Kit de Java (Java SDK). De forma experimental en el

curso 2011-12 en la asignatura presencial se ha utilizado desde el primer día de clase el IDE Netbeans [12] como entorno de desarrollo. En la asignatura a distancia se ha mantenido el entorno de desarrollo Java SDK desde la primera edición del curso.

3.3. EJERCICIOS DE AUTOEVALUACIÓN

La plataforma AulaWeb permite la configuración de ejercicios que se pueden corregir de forma inmediata y automática cuando el alumno finaliza su realización. Estos ejercicios se pueden utilizar de forma sistemática y periódica para focalizar el trabajo de los alumnos y animarles a esforzarse durante todo el periodo académico. Las preguntas previamente introducidas por el profesor se distribuyen en los distintos capítulos del temario y pueden incorporar gráficos y elementos multimedia. Hay varios tipos de preguntas (test de respuesta simple, test de repuesta múltiple, respuesta numérica,...) pero existe un tipo especial específico para esta asignatura que permite la introducción de respuestas de código de Java de tamaño pequeño e intermedio que pueden ser procesadas, compiladas y ejecutadas mediante un applet en el navegador del alumno [11].

Conforme avanza el curso el profesor puede configurar los ejercicios con una fecha de finalización de unos días. Por su parte los alumnos reciben las correcciones de forma inmediata conforme los van finalizando y pueden ir rectificando los errores en los conceptos y aplicaciones de forma dinámica. Los resultados se van almacenando en una base de datos y el profesor puede acceder cómodamente a ellos tanto a nivel individual de cada alumno como del grupo para facilitar el seguimiento de la asignatura desde ambas perspectivas.

Este mismo sistema de ejercicios de autoevaluación se utiliza para controlar la asistencia de los alumnos a las clases presenciales mediante una pregunta cuya respuesta consiste en un código facilitado por el profesor al finalizar la clase.

3.4. TRABAJOS INDIVIDUALES DE PROGRAMACIÓN

En la asignatura presencial se han pedido trabajos individuales hasta el curso 2010-11. De hecho los trabajos individuales han sido el mecanismo para evaluar a los alumnos. Estos trabajos se han entregado al final de la asignatura. Sin embargo, el último curso 2011-12 la evaluación de la asignatura se ha realizado mediante trabajos en equipo.

En la asignatura a distancia los alumnos deben afrontar cinco y diez tareas individuales de programación a lo largo del semestre. Las tareas se configuran y se entregan mediante AulaWeb. Los alumnos tienen una semana para llevar a cabo la tarea. El profesor puede recuperar el trabajo de los alumnos para

revisarlos y posteriormente calificarlos mediante una rúbrica y realizar los comentarios correspondientes. Esta actividad sólo se emplea en la asignatura a distancia y contribuye con un 20% a la calificación final del alumno en la asignatura.

3.5. TRABAJOS FINALES EN GRUPO DE PROGRAMACIÓN

El objetivo de esta actividad es el desarrollo en equipo de un programa en Java.

En la asignatura a distancia el profesor configura los equipos de tres o cuatro alumnos y cada grupo debe autogestionarse para planificar el diseño e implementación del proyecto de software. Antes de iniciar el desarrollo, todos los equipos deben consultar con el profesor el alcance del programa. Todas las entregas deben incluir el código fuente de un programa con el fin de poder revisarlo y verificar su funcionamiento. Este trabajo contribuye con el 30% en la calificación final del alumno. De manera experimental en el curso 2011-12 se han implementado previamente dos trabajos en grupo adicionales con el objetivo de recopilar referencias web sobre programación en Java (TG1) y confeccionar preguntas de autoevaluación de la asignatura (TG2). Todos estos trabajos se evalúan mediante una rúbrica ad hoc.

En el curso 2011-2012 de la asignatura presencial, para evaluar y como parte del proceso de aprendizaje, los alumnos realizan un trabajo tutelado en grupos de 4 o 5 alumnos. Este trabajo consiste en el desarrollo de una aplicación informática en Java de cierta complejidad. El trabajo se ha dividido en tres hitos de entrega, los dos primeros han sido de control y, de ellos, se ha evaluado la competencia de trabajo en equipo mediante una rúbrica que han cumplimentado los propios alumnos. Desde el punto de vista de programación en Java, estas entregas se corresponden con las etapas de planteamiento y diseño de la aplicación y sirven para que el profesor pueda comprobar si el trabajo está bien encaminado. Del último hito de entrega también se ha evaluado la competencia de trabajo en equipo y además se ha utilizado una rúbrica para la evaluación de la calidad del trabajo. Esta rúbrica es de elaboración propia y recopila distintas evidencias respecto de aspecto muy variados del trabajo: corrección de la implementación, utilización de elementos de programación orientada a objetos en el diseño, calidad del diseño, complejidad, alcance, presencia de comentarios en el código, número de tecnologías (o librerías) empleadas, etc. La rúbrica cubre cuatro competencias con un total de 34 evidencias. Las competencias son (con sus pesos):

- a) Conocimiento de la sintaxis y uso del lenguaje de programación Java (20%)
- b) Diseño de un programa orientado a objetos en Java (30%)

- c) Implementación y desarrollo de un programa en Java (35%)
- d) Competencias complementarias: alcance y utilidad del programa... (15%)

3.6. PRUEBA FINAL DE CONOCIMIENTOS

Para poder tener una referencia común que permita la comparación de resultados entre los diferentes cursos desde el curso 2010-11 todos los alumnos cumplimentan una prueba de nivel de tipo examen escrito tradicional al final del periodo docente. Los alumnos tienen conocimiento de esta prueba desde el inicio del curso así como su no consideración en el cálculo de la calificación final de la asignatura. La prueba tiene una duración de 30 minutos y consta de 20 preguntas de las que cada uno de los dos profesores elaboró 10 preguntas en el primer año de su implantación. En la asignatura presencial se aprovecha la última clase para que todos los alumnos puedan hacerla de forma simultánea. En la asignatura a distancia la prueba se lleva a cabo inmediatamente después la presentación del trabajo final por parte de los alumnos de cada grupo. Como se ha indicado anteriormente las preguntas de la prueba han sido las mismas para ambos grupos de alumnos y desde el primer año de implantación.

4. RESULTADOS

En este apartado se muestran los valores obtenidos en indicadores significativos de la asignatura. Se incluyen las tasas de eficiencia y los coeficientes de correlación entre las calificaciones de las diferentes actividades que se realizan a lo largo del curso.

4.1. CORRELACIÓN EN LA ASIGNATURA PRESENCIAL

En la Tabla II se muestra la correlación entre las calificaciones obtenidas en distintas actividades por los alumnos de la asignatura 9013 en el curso 2010-11. Es necesario subrayar que la calificación final se tomó directamente de la calificación obtenida en el trabajo final individual.

TABLA II

Correlación entre las calificaciones (9013 – Presencial, 2010-11)

	TFa	P1	P2	R	NF
TF	1				
P1	0.573	1			

P2	0.648	0.643	1		
R	0.669	0.922	0.889	1	
NF	1	0.573	0.648	0.669	1

aTF: trabajo final individual; P1: 1ª parte de la prueba de nivel; P2: 2ª parte de la prueba de nivel; R: Prueba de nivel; NF: Calificación final de la asignatura

En la Tabla III se muestra la correlación entre las calificaciones obtenidas en distintas actividades por los alumnos de la asignatura 9013 en el curso 2011-12. Es necesario destacar que la calificación final se basó en gran medida en la calificación obtenida en el trabajo final en grupo.

TABLA III

Correlación entre las calificaciones (9013 – Presencial, 2011-12)

	TFa	P1	P2	R	NF
TF	1				
P1	0.196	1			
P2	-0.169	0.468	1		
R	0.010	0.850	0.863	1	
NF	0.924	0.341	0.038	0.217	1

aTF: trabajo final en grupo; P1: 1ª parte de la prueba de nivel; P2: 2ª parte de la prueba de nivel; R: Prueba de nivel; NF: Calificación final de la asignatura

4.2. CORRELACIÓN EN LA ASIGNATURA A DISTANCIA

En la Tabla IV se muestra la correlación entre las calificaciones obtenidas en distintas actividades por los alumnos de la asignatura 9122 en el curso 2010-11.

TABLA IV

Correlación entre las calificaciones (9122 – A distancia, 2010-11)

	AE	TI	TF	P1	P2	R	NF
AE	1						
TI	0.845	1					
TF	0.181	0.161	1				
P1	0.458	0.327	0.067	1			
P2	0.205	0.152	0.317	0.502	1		

R	0.386	0.278	0.218	0.872	0.860	1	
NF	0.615	0.533	0.560	0.504	0.471	0.563	1

aAE: autoevaluación, TI: trabajos individuales; TF: trabajo final en grupo; P1: 1ª parte de la prueba de nivel; P2: 2ª parte de la prueba de nivel; R: Prueba de nivel; NF: Calificación final de la asignatura

En la Tabla V se muestra la correlación entre las calificaciones obtenidas en distintas actividades por los alumnos de la asignatura 9122 en el curso 2011-12.

TABLA V

Correlación entre las calificaciones (9122 – A distancia, 2011-12)

	AEa	TI	TG1	TG2	TF	P1	P2	R	NF
AE	1								
TI	0.371	1							
TG1	-0.104	0.498	1						
TG2	0.054	-0.161	0.024	1					
TF	-0.077	0.105	0.211	0.278	1				
P1	0.327	0.274	0.148	0.295	0.214	1			
P2	-0.062	0.255	-0.044	-0.048	-0.015	0.196	1		
R	0.365	0.397	0.206	0.052	0.139	0.805	0.739	1	
NF	0.521	0.518	0.350	0.033	0.464	0.341	0.128	0.718	1

aAE: autoevaluación, TI: trabajos individuales; TG1: trabajo en grupo 1; TG2: trabajo en grupo 2; TF: trabajo final en grupo; P1: 1ª parte de la prueba de nivel; P2: 2ª parte de la prueba de nivel; R: Prueba de nivel; NF: Calificación final de la asignatura

4.3. DIFERENCIAS DE MEDIAS

Se ha realizado un test de Student de diferencia de medias sobre la nota del control de los grupos presencial y a distancia obteniéndose como resultado (alfa = 0.05) que en el curso 2010-2011 la diferencia no fue significativa y en el curso 2011-2012 la diferencia sí es significativa.

5. OPINIONES DE LOS ALUMNOS

Mediante un cuestionario implementado en AulaWeb el profesor puede recoger de forma flexible, anónima y generalizada las opiniones de los alumnos de las

asignaturas presencial y a distancia. En general todos los alumnos valoran como muy provechoso el seguimiento de las asignaturas de Programación en Java como parte del itinerario formativo de un ingeniero y recomendarían a otros compañeros su matriculación en las mismas.

Los alumnos de la asignatura a distancia puntúan positivamente la flexibilidad temporal a la hora de organizarse en la dedicación a la asignatura así como los contenidos facilitados para su seguimiento. Muchos de ellos recalcan el significativo tiempo que lleva la preparación de los contenidos de la asignatura y la realización de todas las actividades en general. También aprecian la actividad de trabajo en grupo aunque algunos preferirían poder elegir los compañeros del trabajo en grupo.

Los alumnos de la asignatura presencial destacan la notable cantidad de enorme tiempo empleado en el desarrollo de los trabajos de programación, tanto en el formato individual como en grupo. En este último caso inciden en la existencia de un esfuerzo desigual entre los distintos componentes del equipo.

6. DISCUSIÓN

6.1. Uso de la rúbrica para la evaluación de los trabajos

Un problema que se ha encontrado en el uso de la rúbrica ha sido su extensión. Esto dificulta su implantación en grupos con una ratio elevada de nº de alumnos/profesor. Por un lado, la rúbrica para evaluar un trabajo (complejo) en Java tiene que ser muy completa para evitar subjetividad y para evitar la agrupación de evidencias que pueden ser dispares y, por otro lado, tiene que ser suficientemente concisa para que el alumno se preocupe de analizarla. En este caso parece que esto último no se ha conseguido. Dentro de la rúbrica se han incluido algunas evidencias muy fáciles de cumplir. Como ejemplo es especialmente significativo que en la asignatura presencial una evidencia, que evalúa la introducción de comentarios en el código fuente del programa, solo ha sido completamente cumplimentada por dos de los siete equipos de alumnos.

Otra diferencia entre la evaluación resultante del trabajo y el examen tradicional está en el contenido de la materia estudiada. El trabajo se ha centrado en la aplicación de librerías y tecnologías donde Java es un instrumento para un fin que consiste en obtener un resultado práctico de esa librería o tecnología ya existente. El trabajo (y la rúbrica que lo evalúa) pone el énfasis en la comprensión del lenguaje y en el entendimiento y manejo de librerías. Sin embargo, en el examen el énfasis está en el lenguaje en si y sus detalles. Algunos de estos detalles no tienen tanta relevancia en la práctica o también puede ocurrir que el alumno emplee una solución que encuentre sin entenderla completamente. Se podría decir que el trabajo favorece la capacidad de

encontrar soluciones aunque no se entiendan mientras que el examen requiere un entendimiento previo.

Finalmente, los trabajos tienen una mayor concreción que la asignatura en conjunto, y por tanto, mayor concreción que el examen tradicional que se basa en el conjunto de la asignatura. Esto provoca un conocimiento más especializado, los alumnos saben mucho del tema de su trabajo, al profesor le consta que han dedicado un gran esfuerzo a comprender y manejar temas que no son en absoluto triviales y que requieren un grado de especialización importante. Este esfuerzo y las capacidades que se han adquirido a su través se puede medir en la rúbrica, pero no en el examen tradicional que debe ser obligatoriamente genérico.

6.2. Sobre los resultados de la prueba final.

Existe un problema de motivación a la hora de realizar el examen tradicional, los alumnos saben que el resultado del mismo no condiciona su calificación y eso hace que se esfuercen menos en la prueba. Una manera en que se demuestra este hecho es que el profesor ha tenido que pedir a algunos alumnos que no entreguen antes de tiempo el examen y que intenten solucionar todos los ejercicios propuestos. A este efecto contribuye que las normas de evaluación de la asignatura presencial no permiten suspender a los alumnos que asistan a clase.

6.3. Diferencias entre las asignaturas con diferentes metodologías

Los factores que hacen distintos a los grupos el grupo son: la modalidad de clase, los ejercicios de autoevaluación, el método de evaluación, las rúbricas empleadas para los trabajos entregados por los alumnos, los profesores que imparten la asignatura y, en el último año, el entorno de desarrollo. Los factores comunes son el contenido del temario de la asignatura, la documentación facilitada al alumnado, el semestre de impartición, el número de alumnos de cada grupo, el tipo de asignatura (libre elección) y el tipo de alumno matriculado en la asignatura (incluyéndose un porcentaje de alrededor de un 20% de alumnos Erasmus en cada caso).

Los datos de la prueba escrita indican mejores resultados de los alumnos en la asignatura a distancia y también una mayor correlación entre la nota final y la prueba escrita. El test de Student de diferencia de media confirma que el último año estas diferencias son significativas y será necesario establecer el origen de la diferencia.

Que el origen de la diferencia se encuentre en la existencia de una clase presencial es poco razonable, resultaría que la clase es perjudicial para el proceso de aprendizaje de los alumnos y esto no parece plausible.

Hay sin embargo dos detalles que pueden ser origen de la diferencia de manera razonable: los ejercicios de autoevaluación y el entorno de trabajo, por

las siguientes razones:

(a) Los ejercicios de autoevaluación favorecen el trabajo individual y paulatino del alumno. Además, al existir una realimentación inmediata del resultado, el alumno puede comprobar enseguida si lo que aprende es correcto.

(b) El entorno de desarrollo que proporciona Netbeans es muy eficiente con multitud de facilidades para la escritura rápida de programas, por ejemplo, el autorelleno o la compilación del código fuente según se escribe. No obstante, esto hace que el alumno confíe más en la herramienta que en su propia memoria o en su comprensión de un determinado error de sintaxis y, por tanto, adapta su proceso de aprendizaje hacia otros objetivos muy distintos de los que se evalúan con una prueba escrita tipo examen tradicional.

6.4. Diferencias entre las dos ediciones de la asignatura

Hay otra diferencia entre el curso 2010-2011 y 2011-2012: en la asignatura presencial se ha pasado de trabajos individuales a trabajos en grupo. Para los trabajos del último curso los alumnos han elegido ellos mismos la composición de los grupos y se observa muy claramente una diferencia de comportamiento respecto de los grupos de la otra asignatura elegidos por el profesor. En la asignatura presencial no ha habido quejas explícitas respecto del trabajo de los compañeros que sí existen en la otra asignatura. Esto parece indicar que los alumnos dan cobertura a sus compañeros que participan menos en el trabajo cuando existen relaciones de amistad y a su vez esto puede ser un problema para el aprendizaje individual de la materia.

La evidencia de que los resultados de las calificaciones obtenidas de la rúbrica y del examen no tengan correlación parece indicar que la evaluación no es equivalente y, por tanto, evalúa cosas diferentes. El profesor de la asignatura tiene esta misma opinión a partir de la observación de las capacidades que los alumnos muestran en el desarrollo de la clase durante todo el cuatrimestre y durante la presentación de los hitos del trabajo.

Desde el punto de vista del profesor la diferencia fundamental no está en la rúbrica en si sino sobre lo que evalúa la rúbrica: un trabajo en equipo realizado por los alumnos en su tiempo de trabajo personal.

Respecto de otros años donde la evaluación del trabajo no se ha realizado a través de la rúbrica, la introducción de la misma ha servido para racionalizar y fijar los criterios de evaluación para el trabajo, pero esto no significa que la mejora en la evaluación de un trabajo acerque los resultados de la evaluación de un trabajo a los resultados de un examen tradicional.

Hay dos factores observados por el profesor en las presentaciones de los trabajos y en la corrección del examen tradicional que han influido significativamente en la diferencia entre uno y otro método de evaluación:

(a) El trabajo es en equipo no individual. Mientras que el examen es individual. El profesor ha observado que el equipo ha compensado las carencias de algunos de sus integrantes. El resultado es que las capacidades individuales que se miden en un examen tradicional son inferiores.

(b) La herramienta de trabajo. En clase y en el trabajo se ha utilizado lo que en informática se conoce como un IDE, un entorno de desarrollo integrado, este tipo de herramientas se caracterizan por hacer muy efectivo el trabajo del programador por los múltiples sistemas de ayuda y soporte al desarrollo. Estas ayudas hacen aparentemente innecesario el adquirir una serie de conocimientos de base del lenguaje de programación porque la herramienta arregla sola los errores. Y sin embargo, esta herramienta no está disponible en el examen tradicional de manera que alumnos que han sabido desarrollar aplicaciones más o menos complejas no son capaces de hacer ejercicios sencillos sin el apoyo de la herramienta.

7. CONCLUSIONES

En este trabajo se han analizado diferentes actividades, tecnologías y herramientas docentes que pueden emplearse tanto como apoyo en la impartición de asignaturas presenciales como de soporte básico de asignaturas a distancia. Estas actividades facilitan al alumno la comprensión de los conceptos fundamentales de la asignatura así como la práctica en su aplicación. Por otro lado proporcionan al profesor una herramienta muy valiosa para el seguimiento del proceso de aprendizaje de los alumnos y de evaluación de los mismos.

Utilizar un único mecanismo de evaluación, constante en el tiempo, y aplicado a grupos de alumnos que han seguido distintas herramientas y métodos de aprendizaje resulta ser un medio muy conveniente de realizar comparación de resultados del aprendizaje en una materia o asignatura.

Los resultados son aún más interesantes cuando las metodologías son muy distintas pues, como se ha podido comprobar, se llegan a obtener diferencias significativas.

Se puede deducir una conclusión muy general y no completamente justificada, pero razonable a la vista de los resultados: es mucho más importante el trabajo del alumno que el del profesor. Como se ha indicado en la discusión, los aspectos que influyen directamente en la manera en que el alumno trabaja la materia: ejercicios (individuales y en grupo) y entorno de desarrollo.

8. REFERENCIAS BIBLIOGRÁFICAS

Wankat, P. C. y Oreovicz, F. S. (1993). *Teaching Engineering*, Knovel. URL http://www.knovel.com/web/portal/browse/display?_EXT_KNOVEL_DISPLAY_bookid=1287. Visita 2012, marzo, 20.

García-Beltrán, A. y Martínez, R. (2001). AulaWeb: un sistema para la gestión, evaluación y seguimiento de asignaturas, *Industria XXI*, 2, 11-16.

García-Beltrán, A. y Arranz, J.M. (2004). *Introducción a la Programación con Java*, Madrid: Sección de Publicaciones de la ETSII-UPM.

García-Beltrán, A. y Arranz, J.M. (2009). Programación en Java I, OpenCourseWare de la UPM. URL <http://ocw.upm.es/lenguajes-y-sistemas-informaticos/programacion-en-java-i>. Visita: 30 de marzo de 2012.

Martínez, R. y García-Beltrán, A. (2003). Manual de AulaWeb, Madrid: Sección de Publicaciones de la ETSII-UPM.

[6] C. Fernández, D. Díez, J. Torres y T. Zorraonandía, T. (2009). "The cost of learning and teaching Java in the Bologna process" en Proc. of the 2nd Workshop on Methods and Cases in Computing Education, Barcelona, España, 2009, pp. 41-45.

Java SDK URL
<http://www.oracle.com/technetwork/java/javase/downloads/index.html>. Fecha de última consulta: 30 de marzo de 2012

[8] ADL, Advanced Distributed Learning (2010), About Sharable Content Object Reference Model (SCORM), URL
<http://www.adlnet.gov/Technologies/scorm/default.aspx>. Fecha de última consulta: 30 de marzo de 2012

[9] García-Beltrán, A., Martínez, R., Muñoz, D. J. y Muñoz-Guijosa, J. A. (2007). Implementación de un Módulo de Gestión de Contenidos SCORM en la Plataforma AulaWeb, In Proc. of SPDECE 2007, Diseño, Evaluación y Desarrollo de Contenidos Educativos Reutilizables, Bilbao, España.

[10] A. García-Beltrán y J.M. Arranz, *Introducción a la Programación con Java*, Sección de Publicaciones de la ETSII-UPM, Madrid (2007).

[11] A. García-Beltrán, S. Tapia, R. Martínez, y J. A. Jaén, Simulator for a Multi Programming Environment for Computer Science Learning and Teaching, *International Journal of Engineering Education*, 25(2), (2009) 221-227.