

# PREGUNTAS AUTOEVALUABLES DE PROGRAMACIÓN EN C/C++ Y JAVA EN EL SISTEMA DE E-LEARNING AULAWEB

M. GONZÁLEZ, A. GARCÍA-BELTRÁN, R. MARTÍNEZ Y S. TAPIA  
*Departamento de Automática, Ingeniería Electrónica e Informática Industrial.  
Universidad Politécnica de Madrid. España*

*En este trabajo se describe el desarrollo de un sistema de autoevaluación basado en el servicio WWW que permite la creación de preguntas de codificación multilenguaje. El sistema se ha implementado en una primera fase para preguntas de código en C/C++ y Java y se ha integrado dentro del módulo de autoevaluación del entorno de eLearning AulaWeb desarrollado y empleado como sistema de apoyo a la enseñanza en la Universidad Politécnica de Madrid y en otros centros de formación.*

## 1. Introducción

La realización de ejercicios prácticos es una actividad fundamental dentro del objetivo de aprender un lenguaje de programación. Uno de los principales problemas en esta actividad del proceso de aprendizaje estriba en la necesidad de poder comprobar la corrección de lo aprendido mediante alguna forma de evaluación. Si el estudiante realiza esta labor por sí sólo, sin necesidad de otra persona, se está tratando de una autoevaluación. En este caso, el propio estudiante debe poder, de algún modo, acceder a la respuesta correcta para poder compararla con la dada. Este procedimiento puede plantear algunas dificultades en el caso de ejercicios cuya respuesta es un código de programación: el acceso a la respuesta correcta no siempre es posible o bien la comparación directa de la respuesta correcta con la dada no permite inferir un juicio claro. El otro método, clásico, de evaluación requiere de la existencia de un profesor que corrija las respuestas de los alumnos a los ejercicios planteados. En este caso los inconvenientes pueden focalizarse en la imposibilidad de la presencia física del profesor en el momento de realizar el ejercicio, en la carga de trabajo del profesor si el grupo de estudiantes es relativamente numeroso o en el intervalo de tiempo que puede transcurrir entre el instante en el que el alumno concreta la respuesta y el momento en el que el profesor hace llegar al alumno el resultado de la evaluación.

La solución a estos obstáculos planteada en este trabajo se centra en un sistema de preguntas autocorregibles o autoevaluables que permitan al alumno la realización de ejercicios de programación, en las que las respuestas a las preguntas son trozos de código escritos en lenguaje de alto nivel, y que puedan ser corregidos de forma automática e inmediata por parte del propio sistema, sin necesidad de intervención de un profesor.

Siguiendo esta idea, en esta comunicación se describe como se ha desarrollado un sistema de autoevaluación basado en el servicio web que permite la implementación de preguntas de código en lenguajes de programación de alto nivel, en concreto en Java y C/C++ [1]. Este desarrollo es específico y original de la plataforma de *e-learning* AulaWeb empleada desde 1999 en la Escuela Técnica Superior de Ingenieros Industriales y en otros centros de la Universidad Politécnica de Madrid [2, 3]. Se pretende que el sistema pueda utilizarse como apoyo en la enseñanza y aprendizaje de cualquier lenguaje de programación aunque, por ahora, sólo se ha implementado para C/C++ y Java y se ha utilizado en varias asignaturas de las titulaciones de Ingeniería Industrial, Ingeniería Química, Ingeniería de Organización e Ingeniería en Automática y Electrónica Industrial en la ETSII-UPM.

## 2. La plataforma AulaWeb

AulaWeb es una plataforma de tele-enseñanza desarrollada en el Departamento de Automática, Ingeniería Electrónica e Informática Industrial de la ETSII-UPM que se ha utilizado fundamentalmente como herramienta de apoyo para la impartición y el seguimiento de asignaturas presenciales de las carreras impartidas en sus centros (*b-learning*) pero también como herramienta de *e-learning* de cursos a distancia. La aplicación fue inicialmente concebida para facilitar el aprendizaje a los alumnos que por primera vez recibían conocimientos de informática como asignatura presencial en la ETSII-UPM y ha sido desarrollada empleando diferentes tecnologías de software: HTML [4], ASP (VBScript) [5], Java [6] y MS-SQL [7].

El entorno de la aplicación permite la realización de varias actividades docentes mediante el empleo de un ordenador con conexión a Internet y un navegador web independientemente de su sistema operativo. La interfaz de AulaWeb está basada en una estructura mixta de iconos y menús, que permite al usuario saber en todo momento como acceder a una determinada zona o módulo del sistema con el objeto de facilitar al máximo la navegación del usuario. La aplicación incluye un sistema de seguridad a nivel de usuario para el acceso de alumnos, profesores y administrador general que se realiza a través de la red y tres interfaces distintas según el tipo de usuario: alumno, profesor y administrador general. Mediante un entorno amigable y sencillo de utilizar, se trata de facilitar tanto al alumnado como al profesorado, tareas como la publicación y recogida de información y recursos formativos, la realización y entrega de trabajos y prácticas, la ejecución de actividades de autoevaluación o el establecimiento de tutorías telemáticas en tiempo real.

## 3. El sistema de autoevaluación de AulaWeb

El módulo de autoevaluación de AulaWeb incluye varias funcionalidades:

- 1) un gestor de preguntas que facilita al profesor la introducción de preguntas en la base de datos de la aplicación. Las preguntas pueden ser distintos tipos: verdadero-falso, test de selección simple, test de selección múltiple, de respuesta numérica entera o real, de respuesta de tipo cadena de caracteres, de enunciado variable y de respuesta tipo código de programación en TurboPascal. Además, el profesor puede asignar a cada pregunta un capítulo del temario y un nivel de dificultad a priori (muy fácil, fácil, medio, difícil y muy difícil).
- 2) un sistema de configuración que permite indicar todos sus parámetros de los ejercicios: título del ejercicio, grupo de alumnos al que va dirigido, número de preguntas, tipo de preguntas (fijas o aleatorias y de test, numéricas, de programación...), capítulo y nivel de dificultad de las preguntas, tiempo límite (duración en pruebas cronometradas), forma de corrección, fecha límite de finalización y máscara de red de los ordenadores conectados a internet desde los cuales se puede realizar el ejercicio.
- 3) el generador y administrador de ejercicios que, en base al contenido de la base de datos de preguntas y a los parámetros de configuración del ejercicio correspondiente, compone su contenido, presenta las preguntas y almacena las respuestas y los resultados en la base de datos y muestra dichos resultados a alumnos y profesores de la asignatura correspondiente.

Hasta hace poco, además de los tipos clásicos de preguntas de test y de respuesta *corta* (numérica o de cadena), sólo existía el tipo de pregunta de código de programación en TurboPascal en la plataforma AulaWeb. Este tipo de pregunta, muy interesante en la asignatura de Informática, permite la simulación de un entorno de programación y el planteamiento de preguntas en las que el alumno debe completar un programa o código en TurboPascal de forma que realice una tarea determinada. La finalidad fundamental de este tipo de pregunta es fijar los conceptos y técnicas descritas en las clases de teoría y acercarse a los aspectos prácticos aprendiendo los métodos de la programación en TurboPascal. La mayor ventaja era la corrección automática (sin la necesidad de la actuación de un corrector *humano*) e inmediata de las respuestas de código, pero con el

inconveniente de la restricción a código de lenguaje de TurboPascal. Para poder hacer extensible el sistema de autoevaluación al aprendizaje de otros lenguajes de programación se hizo necesario un nuevo desarrollo en el módulo de autoevaluación.

#### **4. Preguntas de autoevaluación en Java y C/C++**

Las preguntas de código de programación en Java y C/C++ se han implementado mediante un conjunto de páginas desarrolladas Microsoft *Active Server Pages* (ASP) [5], un *applet* de Java [6] y una base de datos gestionada por la aplicación Microsoft *SQL Server 2000* [7]. Además ha sido necesario emplear compiladores de Java y de C/C++. En concreto, se han escogido el J2SE Development Kit 5.0 [8] y el DJGPP (versión DOS del compilador GNU GCC) [9]. Las nuevas funcionalidades añadidas a la plataforma AulaWeb para la implementación de este tipo de preguntas han sido fundamentalmente las dos siguientes:

- 1) La edición (creación y modificación) de la pregunta en la interfaz del profesor.
- 2) La presentación de la pregunta en la interfaz del alumno, así como la gestión y corrección de la respuesta.

#### **5. Edición de preguntas en la interfaz del profesor**

Esta funcionalidad incluye la posibilidad tanto de creación de una nueva pregunta como de modificación de una pregunta ya existente en la base de datos. El sistema facilita una secuencia de formularios con elementos de texto para llevar a cabo las dos tareas anteriormente indicadas. Para construir una nueva pregunta es necesario introducir los siguientes datos:

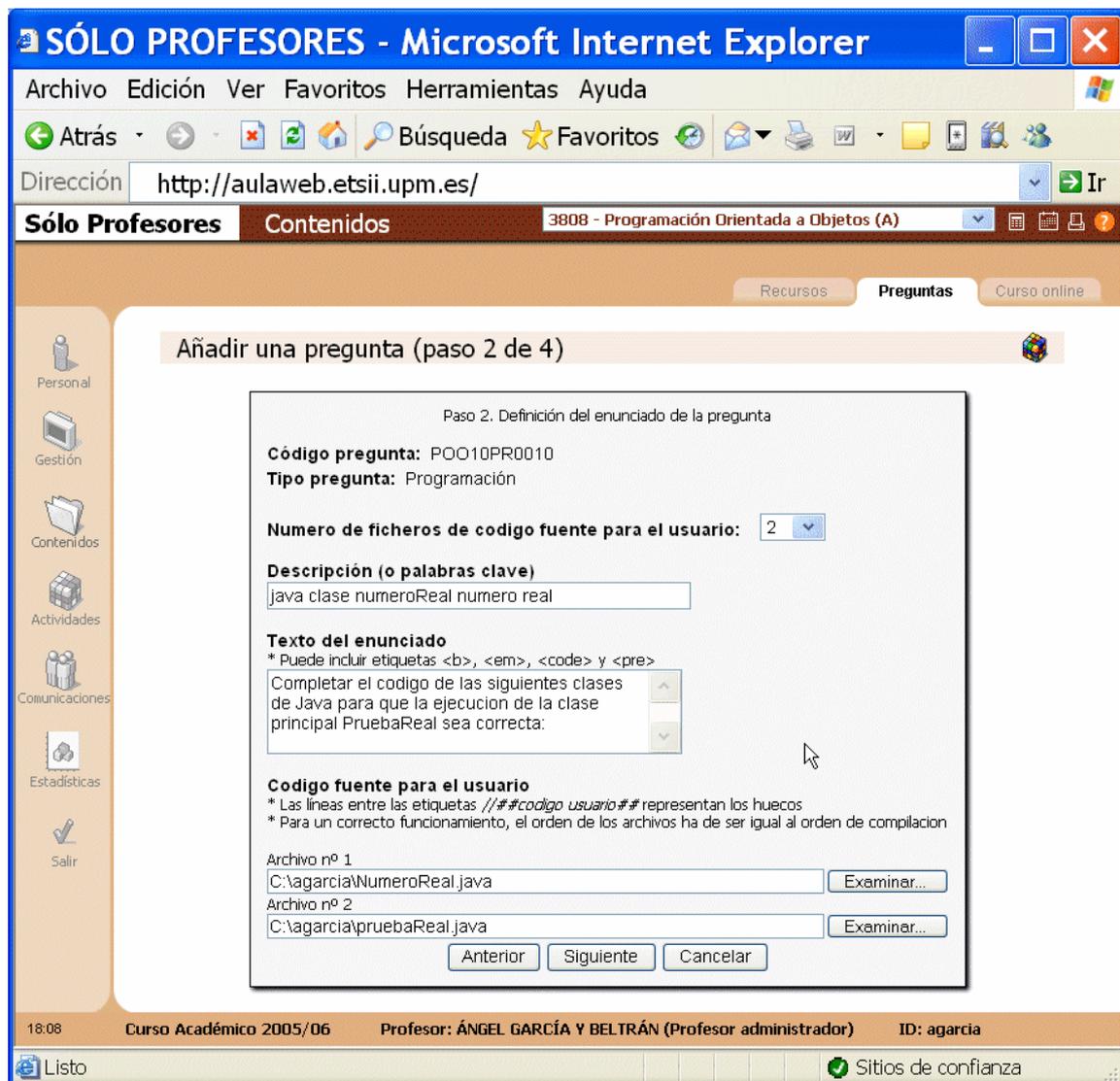
- 1) La descripción o palabras-clave asociadas a la pregunta que pueden servir para posteriores procesos de búsqueda en la base de datos.
- 2) El capítulo del temario al que pertenece la pregunta.
- 3) El nivel de dificultad teórico (muy fácil, fácil, medio, difícil y muy difícil) de la pregunta.
- 4) enunciado de la pregunta (si es necesario se pueden incluir etiquetas del lenguaje HTML para dar formato al texto, a los párrafos, introducir tablas...).
- 5) El código de los archivos fuente a completar adjunto al enunciado (pueden ser más de uno). Cada uno de estos puede incluir cero, uno o más huecos de código a rellenar por el alumno. Estos huecos se indican situándolos entre dos etiquetas con el código: `//##codigo usuario##`. Normalmente para facilitar este trabajo se le indica al asistente para la introducción de preguntas el nombre de uno o más archivos fuente de un directorio local.
- 6) Un ejemplo de código fuente solución (puede ser más de uno).
- 7) El código de los archivos fuente que componen el programa corrector (pueden ser más de uno). Como en el punto 5 anterior se le indica al asistente para la introducción de preguntas los nombres de los archivos fuente correspondientes de un directorio local.

De forma automática, el sistema asigna un código identificador a la nueva pregunta introducida. Además en la base de datos se reserva espacio para almacenar otra información adicional para cada pregunta:

- 1) El número de veces que es seleccionada la pregunta para componer un ejercicio (R).
- 2) El número de veces que ha sido respondida correctamente (A).
- 3) El número de veces que se ha quedado sin contestar (SC).
- 4) El nivel de dificultad experimental tanto a nivel absoluto o *global* (en relación a todas las preguntas de autoevaluación de la asignatura, DG) como a nivel *relativo* (en relación a las preguntas del capítulo correspondiente, DR).
- 5) El tipo de pregunta (en este caso, es de tipo *Programación*).
- 6) La fecha de creación o de la última modificación de la pregunta.
- 7) El identificador del profesor responsable o autor de la pregunta.

- 8) Si está activa o no (si el estado de una pregunta es *inactiva* entonces no puede ser seleccionada para conformar un ejercicio).
- 9) El identificador del lenguaje de programación correspondiente

En la Figura 1 se muestra un ejemplo de ventana de la interfaz del profesor en la que aparece uno de los pasos del asistente para introducir una nueva pregunta de programación en la base de datos de la plataforma

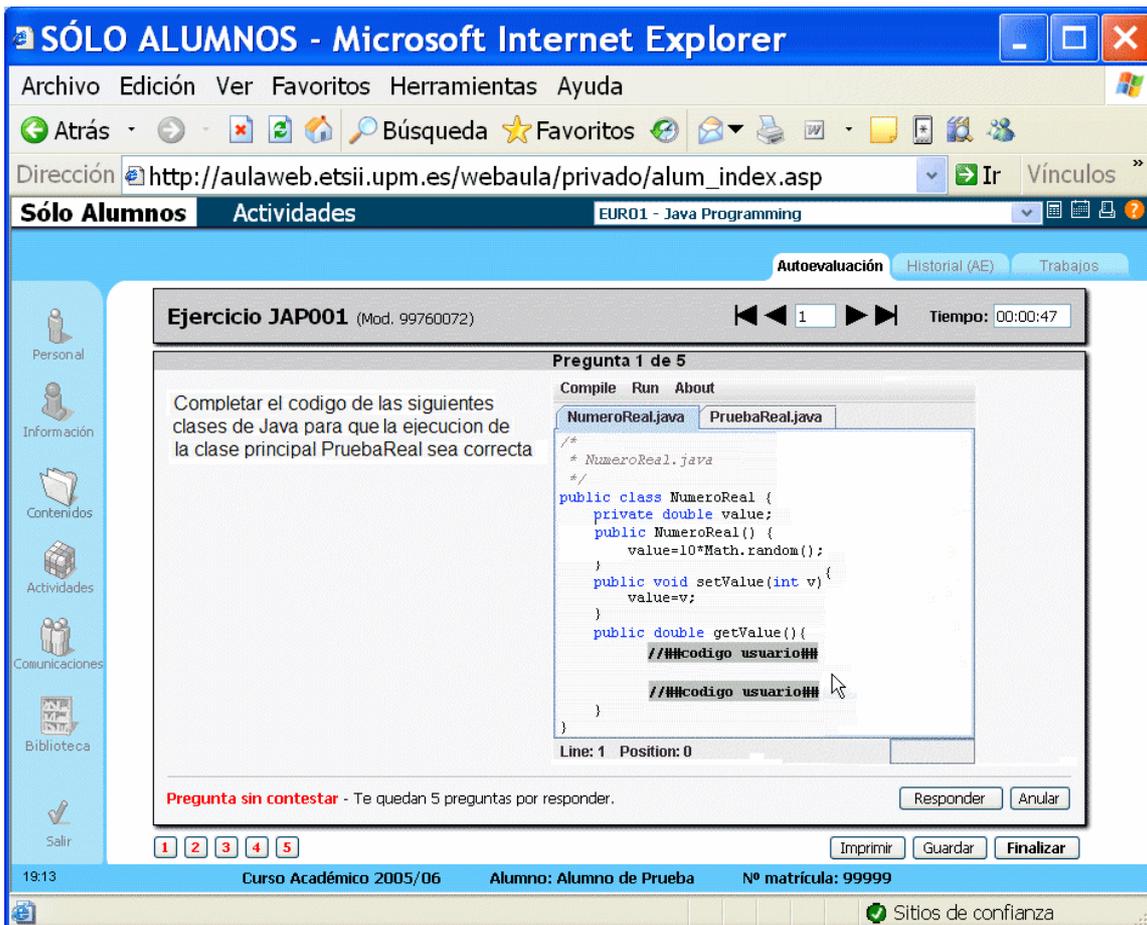


**Figura 1.** Interfaz del profesor en el módulo de contenidos para introducir una nueva pregunta de programación.

Para modificar una pregunta existente el sistema recoge estos datos de la base de datos y mediante otro asistente se los presenta al profesor para que pueda cambiar lo que considere oportuno. La plataforma también facilita al profesor otras operaciones como el listado completo o selectivo de preguntas almacenadas en la base de datos, la impresión de los elementos componentes de una pregunta, la activación/desactivación y eliminación de preguntas, la actualización de la dificultad teórica y la búsqueda por palabras clave de preguntas.

## 6. Presentación de la pregunta en la interfaz del alumno

En la interfaz del alumno (Figura 2) se muestra el enunciado de la pregunta que especifica lo que se plantea en la pregunta y lo que se pide que haga la respuesta de código de programación a introducir por el alumno. El *applet* de Java, que acompaña al enunciado de la pregunta, emula un entorno virtual de desarrollo de software que muestra el contenido de cada uno de los archivos fuente de la aplicación a completar y que acompañan al enunciado. Las solapas incluidas en el entorno permiten seleccionar el acceso a cada uno de los archivos fuente.



**Figura 2.** Interfaz del alumno en el módulo de autoevaluación con preguntas de programación de código de Java.

Los alumnos no necesitan la instalación de ninguna herramienta de software en su ordenador salvo el propio navegador web, ya que pueden emplear la ventana de edición del entorno virtual para introducir y editar el código respuesta. El entorno incorpora además la identificación semántica de las palabras del lenguaje de programación (C/C++ o Java) escogido para cada pregunta en cuestión. Así, los identificadores o palabras *reservadas* del lenguaje aparecen destacados en color azul, las cadenas literales de caracteres (entrecomilladas) se presentan en rojo, los comentarios de código se muestran en gris y el resto de palabras y caracteres de texto en negro. Además aparecen resaltadas en negrita las cadenas "**///**codigo usuario**##**" que delimitan los huecos o espacios reservados al alumno para que escriba la respuesta o respuestas a la pregunta. En la barra de estado situada en la parte inferior se indican las coordenadas del cursor en el archivo fuente que se está editando y el estado del proceso que se está ejecutando.

También se pueden utilizar las opciones de la barra de menú para compilar (**Compile**) o ejecutar (**Run**) el código. Una vez introducido, el alumno puede *compilar* en cualquier momento el código respuesta dentro del archivo fuente del enunciado. Al pulsar **Compile**, el código respuesta se envía al servidor para su compilación. Si la compilación tiene éxito, aparece una ventana con el mensaje correspondiente enviado por el servidor. Si al usar la opción **Compile** se genera un error de compilación en el servidor, aparece una nueva ventana con el mensaje de error correspondiente, el nombre del archivo fuente y la posición del error encontrado. Por su parte, la opción **Run** permite ejecutar el programa corrector introducido previamente por el profesor y que incluye el código necesario para detectar errores lógicos o de ejecución, en cuyo caso se indica también con el mensaje correspondiente. Al usar esta opción (y también cuando el alumno finaliza el ejercicio), las respuestas a este tipo de preguntas son enviadas al servidor de AulaWeb para su corrección automática, siendo la contestación inmediata y mostrándose las ventanas de aviso correspondientes tanto si se ha tenido éxito como si se ha generado un error de ejecución.

Para poder llevar a cabo estos dos tipos de correcciones *on-line* (tanto de compilación como de ejecución) de las preguntas de código de programación, se ha instalado en dicho servidor un compilador de los lenguajes correspondientes que realiza las tareas correspondientes de forma transparente al usuario. En la parte inferior derecha aparecen otros botones de utilidad para la realización del ejercicio. El botón **Responder** permite almacenar las respuestas introducidas por el alumno y pasar a la siguiente pregunta del ejercicio. El botón **Anular** borra las respuestas introducidas para poder responder a la pregunta desde cero. Por otro lado, el alumno también puede imprimir la pregunta en cualquier momento, guardar un ejercicio para poder ser reanudado en una sesión posterior, finalizar un ejercicio para que sea corregido inmediatamente y obtener todos los datos (enunciados completos, respuestas y resultados) de un ejercicio una vez finalizado.

Todas estas funcionalidades están también accesibles al profesor para poder probar el correcto funcionamiento del sistema y de las preguntas de código de programación introducidas en la base de datos del sistema. Asimismo desde la interfaz del profesor pueden obtenerse datos experimentales tanto de los resultados de los ejercicios realizados por un alumno en particular como de los resultados obtenidos por una pregunta en concreto. Esto permite conseguir información muy valiosa acerca de los progresos en el proceso de aprendizaje de un alumno en particular y de un grupo de alumnos en general.

## **7. Utilización del sistema**

Este tipo de preguntas es especialmente interesante para uno de los objetivos fundamentales de las asignaturas relacionadas el aprendizaje de este tipo de lenguajes de programación en la educación superior o universitaria. En principio, la aplicación se ha utilizado de forma experimental en tres asignaturas impartidas por el personal docente del Departamento de Automática, Ingeniería Electrónica e Informática Industrial en las titulaciones de Ingeniería Industrial, Ingeniería Química e Ingeniería en Automática y Electrónica Industrial impartidas en la ETSII-UPM en el curso 2005-06. Las metodologías empleadas han sido distintas según la asignatura:

- 1) En *Programación Orientada a Objetos* (asignatura opcional del 1º semestre del 2º curso del título de segundo ciclo de Ingeniería en Automática y Electrónica Industrial) se ha utilizado como procedimiento de seguimiento de los alumnos que asisten a clases presenciales. Tras dichas clases los alumnos realizan los ejercicios de autoevaluación desde sus domicilios particulares a través del servicio web [10]. Los resultados de los ejercicios tienen una aportación en la calificación final de la asignatura. Esta circunstancia junto con la obtención de resultados inmediatos incentiva aún más la realización de los ejercicios por parte de los alumnos.
- 2) En *Programación en C* (asignatura de libre elección), las preguntas de código se emplean

durante las clases presenciales impartidas en un aula de ordenadores para la realización de prácticas de programación. El profesor puede explicar personalmente los ejercicios planteados, resolver dudas y ver el progreso de los alumnos in situ.

- 3) En la asignatura *Java Programming* (de libre elección impartida a distancia y en inglés), no hay clases presenciales y el profesor va facilitando a los alumnos los contenidos conforme avanza el curso. De forma paralela emplea un régimen continuo de autoevaluación para, por un lado, motivar el trabajo de los alumnos desde el principio al final del curso y por otro lado, poder verificar los avances en el aprendizaje de los alumnos de forma telemática.

A modo de ejemplo, en la Figura 3 se muestra el listado de preguntas de programación almacenadas en la base de datos para la asignatura de Programación Orientada a Objetos. Entre otros datos para cada pregunta de programación aparece el código asignado, el capítulo correspondiente, el número de veces que ha sido seleccionada para formar parte de un ejercicio (R), el número de veces que ha sido contestada correctamente (A) y el número de veces que ha quedado sin contestar (SC). Estos resultados corresponden a los ejercicios realizados en este curso 2005-06.

**SÓLO PROFESORES - Microsoft Internet Explorer**

Archivo Edición Ver Favoritos Herramientas Ayuda

Dirección <http://aulaweb.etsii.upm.es/>

**Sólo Profesores**    **Contenidos**    3808 - Programación Orientada a Objetos (A)

Recursos    Preguntas    Curso online

**Resultados de la búsqueda**

A continuación se muestran las preguntas de la base de datos que coinciden con su criterio de búsqueda. Pulse sobre el código de cada una de ellas para ver un borrador de su enunciado. También puede marcar varias preguntas para imprimir, eliminar, activar, desactivar, inicializar estadísticas, actualizar dificultad o cambiar de capítulo.

DR: dificultad relativa, DG: dificultad global, DE: dificultad estadística, R: nº veces realizada, A: nº veces acertada, SC: nº de veces sin contestar.

CÓDIGO	Capítulo	Tipo	DR	DG	DE	R	A	SC	Revisión	Estado
<input type="checkbox"/> P0002PR0480	02	PR	0,1	0,1	0,11	9	8	1	agarcia (08/11/2005)	Modificar ✓
<input type="checkbox"/> P0002PR0490	02	PR	0,1	0,1	0,19	16	13	1	agarcia (08/11/2005)	Modificar ✓
<input type="checkbox"/> P0002PR0500	02	PR	0,1	0,1	0,23	13	10	2	agarcia (08/11/2005)	Modificar ✓
<input type="checkbox"/> P0002PR0510	02	PR	0,3	0,3	0,14	7	6	0	agarcia (08/11/2005)	Modificar ✓
<input type="checkbox"/> P0003PR0250	03	PR	0,3	0,3	0,38	8	5	0	agarcia (20/11/2005)	Modificar ✓
<input type="checkbox"/> P0003PR0260	03	PR	0,5	0,5	0,5	6	3	1	agarcia (20/11/2005)	Modificar ✓

Seleccionar todas / ninguna

Nº de preguntas encontradas: 6

Activar    Borrar    Actualizar dificultad

Desactivar    Imprimir    Inicializar realizadas/acertadas

Imprimir descripción

Volver

DR: dificultad relativa, DG: dificultad global, DE: dificultad estadística, R: nº veces realizada, A: nº veces acertada.

17:58    Curso Académico 2005/06    Profesor: ÁNGEL GARCÍA Y BELTRÁN (Profesor administrador)    ID: agarcia

Listo    Sitios de confianza

**Figura 3.** Interfaz del profesor en el módulo de contenidos en el que se muestra un listado de preguntas de programación de la base de datos

Los datos mostrados en la pantalla anterior le sirven al profesor para verificar el correcto funcionamiento de las preguntas, contrastar los coeficientes asociados a las dificultades teóricas y experimentales de cada una de ellas y analizar la asimilación de los conceptos teóricos asociados a éstas por el grupo de alumnos en general. A nivel individual el sistema AulaWeb permite al profesor acceder al historial de ejercicios de autoevaluación de un alumno para ver su evolución particular.

Al finalizar el primer semestre los alumnos en los que se utilizó por primera vez este nuevo sistema de autoevaluación han sido encuestados para evaluarlo: la mayoría de ellos destacan la utilidad del sistema para motivar el esfuerzo continuado y la flexibilidad temporal y espacial que otorga para la realización de ejercicios de programación.

## **8. Conclusiones**

El módulo de autoevaluación con preguntas de programación puede ser una herramienta muy útil desde el punto de vista docente para el aprendizaje de lenguajes de programación en asignaturas con un número elevado de alumnos y en el que sea deseable un sistema flexible y eficiente para la realización de prácticas. Esta actividad se considera imprescindible para la asimilación tanto de las técnicas básicas de programación como de aspecto más avanzados, sin que suponga un esfuerzo de corrección para el profesor. La plataforma permite además la implantación de un sistema regular de ejercicios que estimula el trabajo continuado del alumno durante todo el periodo académico.

En concreto, el *applet* desarrollado permite realizar a los alumnos ejercicios de programación con el navegador sin necesidad de instalar un entorno real de programación y conocer de forma inmediata y automática si la respuesta introducida (código de Java o de C/C++) es correcta o no. Esta es una de las características que los alumnos, los principales beneficiarios del sistema, destacan de la aplicación.

El principal inconveniente puede ser el esfuerzo que conlleva al profesor la preparación de una pregunta de este tipo, aunque puede quedar compensado con el tiempo ahorrado si tuviera que corregir personalmente dicha pregunta un número muy elevado de veces. En cualquier caso, los profesores que lo han utilizado este curso por primera vez de forma experimental están muy contentos con el uso y rendimiento del sistema y de su eficacia como sistema de apoyo docente.

Como líneas de futuro desarrollo se destacan las siguientes:

- 1) La explotación general del sistema de autoevaluación con preguntas de programación en otras asignaturas y cursos impartidos en la ETSII-UPM y en otros centros en dónde está instalada la plataforma AulaWeb.
- 2) La extensión de las preguntas de programación a otros lenguajes como MatLab, FORTRAN o Maple y ampliación para el uso de librerías de software especializadas como Open GL, Active X o tecnología COM.
- 3) La posibilidad de incluir imágenes y elementos multimedia en el enunciado de las preguntas.
- 4) El desarrollo de una aplicación que facilite la creación y depuración de este tipo de preguntas de programación.

## **Agradecimientos**

Este proyecto se ha realizado con el apoyo del proyecto de investigación SEJ2004-08004-C02-02 del Ministerio de Ciencia y Tecnología. Asimismo, los autores desean agradecer a todas las personas que han colaborado en el proyecto de desarrollo de las distintas versiones del sistema AulaWeb dentro del Departamento de Automática, Ingeniería Electrónica e Informática Industrial

de la ETSII-UPM: Aurora Alonso, Pablo Avendaño, Marcos Aza, Juan Antonio Criado, Francisco de Ory, Carlos Engels, Miguel Fernández, Pilar García, Jorge Granado, José Alberto Jaén, Teresa Hernández, Isadora Iglesias, Antonio R. López, Diego López, Julio A. Martín, Marcos Martín, Francisco José Mascato, David Molina, Carlos Moreno, Luis Miguel Pabón, Juan Carlos Pérez, Alberto Rodelgo, Alberto Valero, Eduardo Villalar y Carlos Zoido.

## Referencias

- [1] M. González, *Sistema de Autoevaluación con Preguntas de Programación Multilenguaje Integrado en la Plataforma de eLearning AulaWeb*, Proyecto Fin de Carrera nº 04199199, ETSII-UPM (2005).
- [2] A. García-Beltrán, R. Martínez, *AulaWeb: un sistema para la gestión, evaluación y seguimiento de asignaturas*, *Industria XXI*, 2, 11-16 (2001).
- [3] R. Martínez, A. García-Beltrán, *AulaWeb: Una plataforma e/b-Learning para la UPM*, VI Congreso de Tecnologías Aplicadas a la Enseñanza de la Electrónica, TAEE 2004 (2004).
- [4] HyperText Markup Language (HTML) Home Page. Disponible en la red en la dirección: <http://www.w3.org/MarkUp/>. Última visita: 16 de junio de 2006.
- [5] ASP.NET Web: The Official Microsoft ASP.NET 2.0 Site. Disponible en la red en la dirección: <http://www.asp.net>. Última visita: 16 de junio de 2006.
- [6] Java Sun Developer Network. Disponible en la red en la dirección: <http://java.sun.com/>. Última visita: 16 de junio de 2006.
- [7] Microsoft SQL Server Home. Disponible en la red en la dirección: <http://www.microsoft.com/sql>. Última visita: 16 de junio de 2006.
- [8] Download Java 2 Platform Standard Edition 5.0. Disponible en la red en la dirección: <http://java.sun.com/j2se/1.5.0/download.jsp>. Última visita: 16 de junio de 2006.
- [9] DGJPP. Disponible en la red en la dirección: <http://www.delorie.com/djgpp/>. Última visita: 16 de junio de 2006.
- [10] A. García-Beltrán, J.A. Jaén, R. Martínez y S. Tapia, *Programación Orientada a Objetos para Alumnos de Ingeniería en Automática y Electrónica Industrial*, Comunicación enviada y aceptada para su presentación en el Congreso TAEE 2006.