

Resolución de ejercicios de programación en la web

Sergio Luján-Mora, Fernando Llopis
Departamento de Lenguajes y Sistemas Informáticos
Universidad de Alicante
E-03080 Alicante
e-mail: {slujan,llopis}@dlsi.ua.es

Resumen

En el presente artículo se presenta una aplicación web que permite a los alumnos resolver ejercicios planteados en la asignatura *Fundamentos de la Programación* de las titulaciones de Informática de la Universidad de Alicante. Esta aplicación se ha planteado a modo de “competición” o juego, con el fin de motivar al alumno a que realice una gran cantidad de ejercicios sobre la asignatura. Las dos principales ventajas que se logran a la hora de emplear esta aplicación son: por un lado, cada alumno puede verificar cuando desee y desde donde desee los conocimientos adquiridos sobre la asignatura y, por otro lado, los profesores obtienen información que les permite conocer el grado de aprendizaje de sus alumnos y los aspectos que mejor o peor han asimilado.

1. Motivación

En la enseñanza de programación en el primer curso de las titulaciones de Informática confluyen una serie de circunstancias que la diferencian de otras asignaturas. En primer lugar, para la mayoría de los alumnos es la primera vez que se enfrentan a una asignatura de este tipo, aunque es cierto que actualmente el conocimiento sobre informática en general, y el uso de ordenadores en particular, es mucho mayor que hace 4 o 5 años. En segundo lugar, el esfuerzo que deben realizar los alumnos para su aprendizaje debe ser mucho más deductivo que memorístico. En último lugar, también cabe citar la importancia que suele tener esta asignatura dentro de los planes de estudio a nivel de incompatibilidades o recomendaciones sobre otras asignaturas. Todo esto influye en que el alumno se sienta muchas veces “presionado” a aprobar la asignatura, lo que conduce a que el alumno intente aprobar la asignatura mediante la memorización

en vez del aprendizaje correcto de los conocimientos que recibe.

La solución que recomendamos a los alumnos para que el aprendizaje de la asignatura sea lo menos costoso y a su vez lo más provechoso posible es la realización del mayor número posible de ejercicios prácticos. Con el objetivo de facilitar esta tarea a los alumnos, se han propuesto a lo largo de los seis últimos años numerosos ejercicios, que han sido recopilados junto con ejercicios de exámenes en un libro [4], donde aparecen la mayoría de los ejercicios con su correspondiente solución.

Parece claro que el disponer de ejercicios con enunciado y solución facilita el aprendizaje de la asignatura, pero en ocasiones puede producir el efecto contrario si el alumno sólo se limita a leer el enunciado y la solución, sin llegar a pasar por el proceso deductivo que teóricamente tiene que realizar para obtener la solución.

Por otro lado, la resolución por parte de los alumnos de los ejercicios propuestos en clase de teoría, formando grupos y realizando pequeñas competiciones entre ellos se ha comprobado que motiva a los alumnos y sí que logra que intenten resolver los ejercicios. No obstante, la masificación de las aulas y la cantidad de tiempo necesaria para llevar a cabo estos ejercicios, impide que se dediquen muchas clases a este método de estudio. Estos problemas no deben impedir que este tipo de clases se sigan impartiendo de dicha forma, pero sí que limita considerablemente su número.

2. Propuesta

Por todos los motivos anteriores, hemos desarrollado una aplicación web, a la que hemos denominado *Aprendizaje en la Web AutoMotivado* (AWAM) [1], que nos permite poner a

disposición de los alumnos una gran cantidad de ejercicios que pueden resolver cuando ellos quieran (la aplicación está disponible las 24 horas del día) y desde donde ellos quieran (se puede acceder a la aplicación desde cualquier ordenador con acceso a Internet). Con el fin de motivar al alumno, hemos dotado a la aplicación de una apariencia de “competición” o juego que les anime a superarse. El funcionamiento de la aplicación es sencillo: se plantea al alumno una serie de ejercicios que tienen que resolver en un tiempo determinado y el juego finaliza cuando el alumno comete un número determinado de errores o contesta todos los ejercicios. Según el número de ejercicios contestados correctamente, el alumno obtiene una puntuación, que se traduce en una posición con respecto a los resultados de sus compañeros.

Los ejercicios planteados son de tipo test, agrupados por los diferentes temas que conforman el programa de la asignatura. Estos ejercicios inciden en los errores más comunes que cometen los alumnos a la hora de programar. Cada ejercicio tiene asignada una dificultad; si el profesor lo desea, se puede emplear este valor para presentar al alumno los ejercicios de menor a mayor dificultad. Además, también existe la opción de que esta dificultad se actualice en base a las contestaciones de los alumnos (cuantas más veces sea mal contestada una pregunta, más aumentará su dificultad). Por último, también existe la posibilidad de mostrar al alumno un comentario que explique la respuesta correcta de una pregunta cuando falle.

El resto del artículo se ha dividido de la siguiente forma: en la sección 3 se comentan algunos trabajos previos relacionados con nuestra propuesta; en la sección 4 se describen las características principales de la aplicación AWAM; en la sección 5 se detalla la arquitectura de la aplicación; en la sección 6 se presentan las primeras impresiones que hemos obtenido; por último, en la sección 7 comentamos los trabajos futuros que pretendemos realizar.

3. Trabajos relacionados

En los últimos tres años, el uso de Internet como herramienta docente en la universidad ha aumentado espectacularmente. Prueba de ello son las numerosas ponencias presentadas en JENUI en

los últimos años que emplean Internet, y en especial la Web, como una herramienta de apoyo a la docencia. Simplemente como un ejemplo, incluimos las siguientes referencias del año 2001: [2], [3], [5], [6] y [7].

De las anteriores referencias nos interesa destacar dos por su semejanza con nuestra propuesta. Por un lado, en [2] se presenta un sistema de apoyo a la enseñanza presencial basado en Internet. El sistema permite publicar materiales docentes y, además, los alumnos pueden verificar los conocimientos adquiridos en cada tema, mediante la realización de exámenes de tipo test generados por el sistema. Sin embargo, el sistema presenta varias carencias como son: no existe un control del tiempo que se puede emplear en contestar una pregunta, no se puede indicar la dificultad de una pregunta, todas las preguntas tienen que tener cuatro respuestas, etc.

Por otro lado, en [6] se presenta otro sistema basado en la web, que también posee ciertas similitudes con nuestra propuesta. En este sistema, el alumno puede acceder a través de Internet a listas de problemas generadas de forma aleatoria a partir de una base de datos. Sin embargo, esta propuesta carece de la interactividad e inmediatez de nuestro sistema (no dispone de un módulo para preguntas-respuestas de tipo test con corrección automática), y tampoco presenta el aspecto de juego que creemos motiva la participación del alumno.

4. Descripción de la aplicación

La aplicación AWAM se divide en dos módulos: el módulo del alumno y el módulo del profesor. Para acceder a cualquiera de los módulos, el usuario (ya sea el alumno o el profesor) tiene que disponer de un *nombre de usuario* y una *clave*. Los alumnos además poseen la posibilidad de definirse un alias que será el que utilicen en la realización de los ejercicios. De este modo, pueden realizar los ejercicios de forma anónima, sin tener que preocuparse porque el profesor vaya a tener en cuenta sus posibles fallos.

Toda la información que emplea la aplicación (usuarios, profesores, asignaturas, temas, preguntas y respuestas, etc.) reside en una base de datos. La Figura 1 representa un modelo en UML de la base de datos que emplea nuestro sistema. Cada clase de UML se convierte en una tabla en el

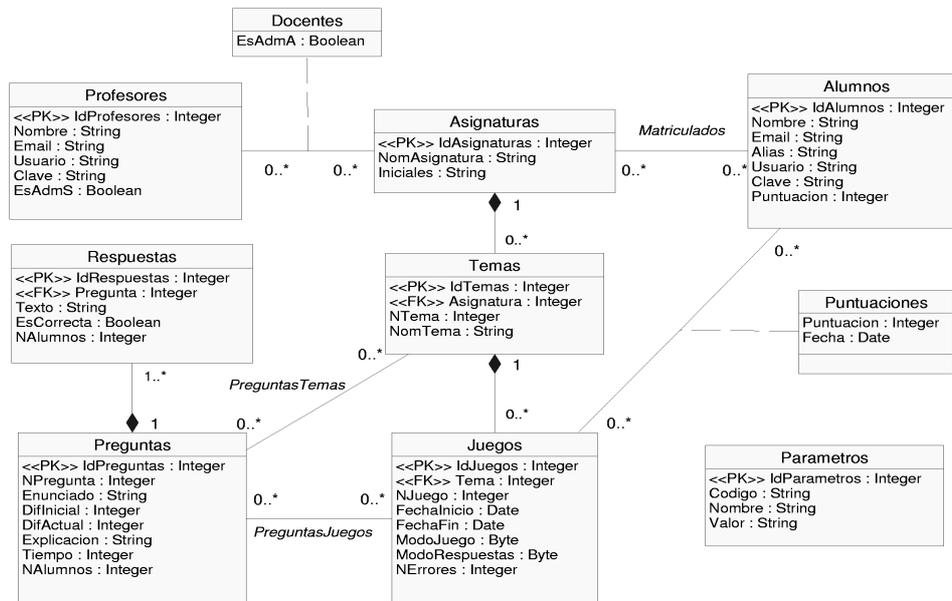


Figura 1. Base de datos del sistema modelada mediante UML

modelo relacional. Las asociaciones que poseen un nombre (PreguntasTemas, Preguntas-Juegos y Matriculados) y las clases asociación (Docentes y Puntuaciones) se convierten en tablas que permiten representar las relaciones muchos a muchos en el modelo relacional. Por último, se han marcado con <<PK>> y <<FK>> las claves primarias y las claves ajenas respectivamente.

Como se puede apreciar en la Figura 1, una asignatura se compone de muchos temas, un tema de muchos juegos, un juego de muchas preguntas, y una pregunta de muchas respuestas (todas las preguntas no tienen por que tener el mismo número de respuestas). Como una pregunta puede pertenecer a varios temas a la vez, se pueden crear temas de repaso que incluyan preguntas de otros temas.

Por último, se ha intentado que la aplicación sea lo más flexible posible. Por eso existe la tabla **Parámetros** que almacena distintos valores que configuran el funcionamiento de la aplicación. Estos parámetros se pueden configurar a través del

menú de configuración de parámetros del sistema (MCPS) que existe en el módulo del profesor.

4.1. Módulo del alumno

A través de este módulo los alumnos pueden contestar las preguntas planteadas por el profesor. Para comenzar a “jugar”, el alumno debe elegir primero la asignatura, después el tema y por último el juego en el que quiere jugar. A continuación, la aplicación muestra una a una las preguntas que contiene el juego. El orden en que se muestran las preguntas depende de cómo lo haya configurado el profesor en el momento de crear el juego (de forma aleatoria, de menor a mayor dificultad o según un orden creado por el profesor).

En la Figura 2 podemos ver la página que se muestra al alumno para que conteste una pregunta. En ella podemos observar la siguiente información:

- Enunciado de la pregunta.

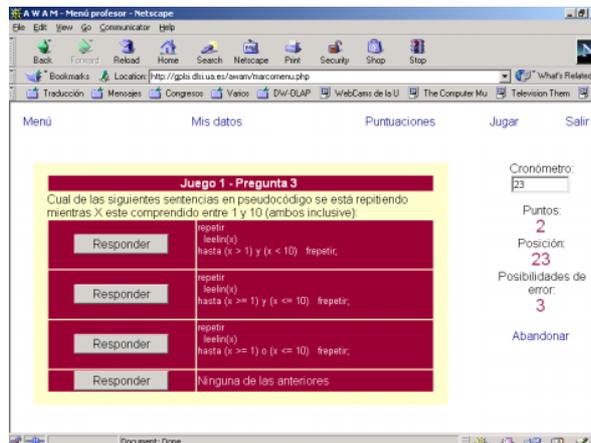


Figura 2. Página con una pregunta y cuatro respuestas

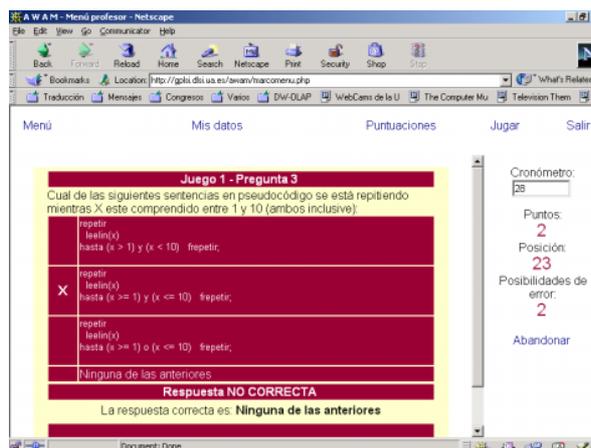


Figura 3. Página con el resultado de la contestación de un alumno

- Posibles respuestas junto con el botón que permite al alumno seleccionar cada una.
- Cronómetro: tiempo restante para contestar la pregunta. Si se cumple el tiempo y el alumno no ha contestado, se considera un fallo.
- Puntos: puntos que ha logrado el alumno hasta el momento en el juego actual.
- Posición: indica la posición del alumno con respecto a las puntuaciones del resto de sus compañeros.
- Posibilidades de error: número máximo de fallos permitidos. El profesor puede fijar este valor a través del MCPS.

Si el alumno acierta la pregunta, sigue el juego; en caso de que falle aparece un mensaje explicando la contestación correcta. Por ejemplo, en la Figura 3 vemos como el alumno ha contestado incorrectamente a la pregunta de la

Puntuaciones Globales			
Posición	Puntuación	Máximo nivel	Alumno
1	1024	12	1234
2	762	9	1234
3	512	9	4452
4	512	8	2348
5	396	8	1267
6	256	7	3334
7	256	6	1165
8	128	4	3712
9	64	3	3455
10	64	3	2453

Figura 4. Tabla de puntuaciones globales

Figura 2. Si ya ha cometido todos los errores permitidos por el profesor, el juego finaliza.

Por otro lado, el alumno puede consultar las tablas de clasificación, ya sea la de puntuaciones globales (Figura 4) o la de puntuaciones por temas. Además, también puede consultar en que posición aparece dentro de cada tabla.

4.2. Módulo del profesor

El profesor o profesores, ya que el sistema contempla la existencia de distintas asignaturas, cada una de ellas con sus profesores correspondientes, dispone en su módulo de todas las opciones necesarias para mantener la base de datos del sistema. El profesor puede añadir (borrar, consultar, ...) nuevos temas, nuevos juegos, nuevas preguntas, etc. Como pueden existir distintos profesores, no todos tienen acceso a todas las opciones. Así, por ejemplo, existen profesores administradores que pueden crear asignaturas, crear otros profesores, etc.; profesores que pueden crear temas, juegos o preguntas en las asignaturas a las que pertenecen, y profesores que sólo pueden introducir preguntas en las asignaturas a las que pertenecen.

En la Figura 5 se puede ver la página principal del módulo del profesor. Se puede observar que se encuentra dividida en dos secciones (marcos): en el marco superior se muestran una serie de listas

desplegables con acceso a las opciones más usuales; en el marco inferior se muestran distintos contenidos según el profesor va eligiendo opciones (la primera vez se muestra el menú completo del módulo del profesor).

Todo el acceso a la base de datos se realiza mediante formularios HTML. Por ejemplo, en la Figura 6 se muestra el formulario que permite añadir una pregunta nueva. Para cada una de las preguntas se introduce la siguiente información:

- Juego al que pertenece la pregunta. Cuando se añade una pregunta nueva sólo se puede elegir un juego, pero una vez añadida se puede asociar a otros juegos.
- Número de pregunta. Si el profesor lo desea, este número se emplea para mostrar las preguntas ordenadas según este valor.
- Enunciado de la pregunta. En el enunciado de la pregunta el profesor puede introducir etiquetas HTML.
- Dificultad inicial de la pregunta. Si el profesor lo desea, este valor se emplea para mostrar las preguntas ordenadas según la dificultad. Además, el profesor también puede indicar que este valor se actualice automáticamente a medida que los alumnos responden a la pregunta de forma correcta o incorrecta.
- Explicación. Permite incluir un breve comentario que explica al alumno, en caso de

que no haya respondido correctamente, el motivo de su fallo. A través de esta opción se puede establecer una conexión con los contenidos teóricos necesarios para contestar correctamente la pregunta.

- Tiempo máximo para contestar la pregunta. Si no se introduce ningún valor, se emplea el tiempo máximo por defecto fijado en el MCPS.
- Posibles respuestas. El número de respuestas es configurable a través del MCPS.

□ Respuesta correcta.

Una opción muy interesante de AWAM es el menú de resultados. A través de él, se pueden obtener distintos parámetros estadísticos como pregunta mejor o peor contestada, porcentaje de acierto medio por pregunta, juego o tema, etc.

Por ejemplo, en la Figura 7 se puede ver la página que muestra al profesor los porcentajes de contestación por preguntas dentro de un juego de un tema. Otras opciones disponibles son: ordenar las preguntas por porcentaje de fallo (acierto),

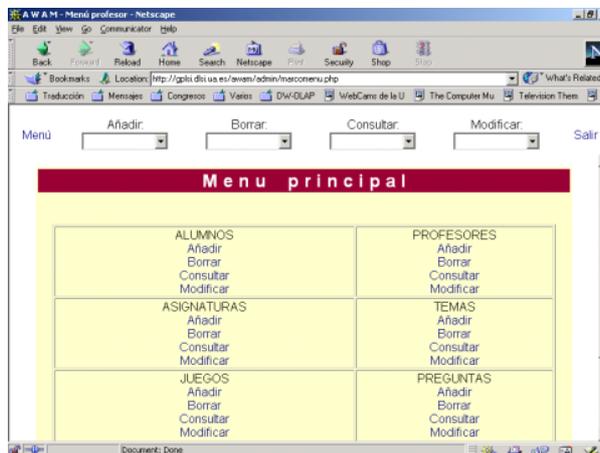


Figura 5. Menú principal del módulo del profesor

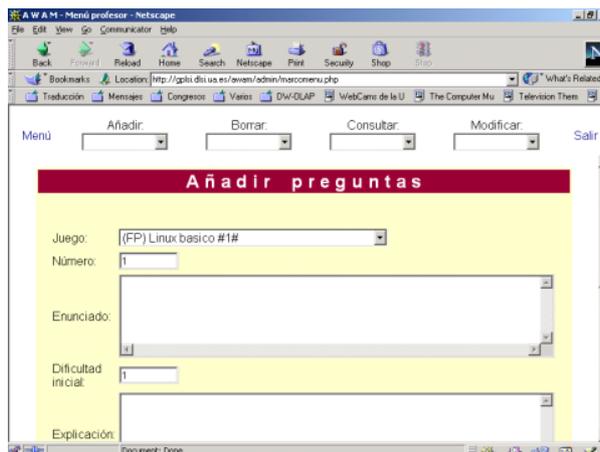


Figura 6. Formulario para añadir una pregunta nueva

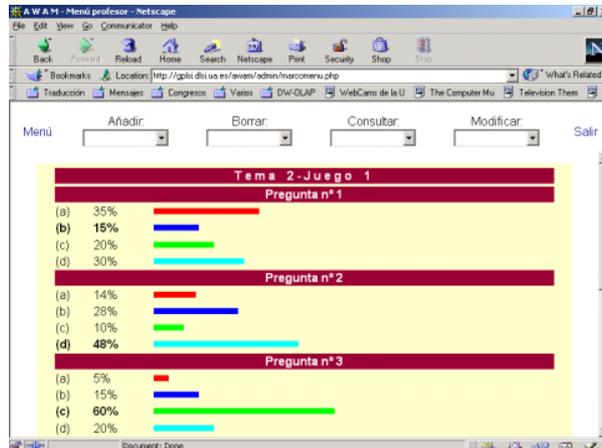


Figura 7. Resultados estadísticos

obtener las preguntas cuyo nivel de dificultad más ha cambiado, ordenar los temas por porcentaje de fallo (acierto), etc.

El profesor también dispone del menú de configuración de parámetros del sistema (MCPS) que permite fijar una serie de parámetros que definen el funcionamiento por defecto del sistema:

- Tiempo máximo para contestar las preguntas.
- Número de preguntas en un juego.
- Número de respuestas por pregunta.
- Posibilidades de error.
- Activar actualización de la dificultad de las preguntas.
- Si se muestra o no la respuesta (y la explicación si existe) cuando un alumno falla una pregunta.
- Modo de juego por defecto: preguntas seleccionadas de forma aleatoria, preguntas ordenadas según la dificultad inicial, según la dificultad actual o según el número asignado por el profesor.

Todos estos valores son por defecto, lo que significa que se pueden variar cuando se desee. Por ejemplo, se puede fijar un tiempo máximo para contestar las preguntas de 30 segundos, pero para algunas preguntas el profesor puede decidir usar otra cantidad.

5. Arquitectura de la aplicación

La aplicación AWAM se puede dividir en tres niveles: lógica de presentación, lógica de negocio (aplicación) y lógica de datos.

La lógica de presentación, encargada de interactuar con el usuario, se ha programado mediante los lenguajes HTML y JavaScript. Esta parte de la aplicación se ejecuta en el cliente (navegador web) y se ha intentado hacer lo más compatible posible. Hemos comprobado su correcto funcionamiento en los navegadores más conocidos: Microsoft Internet Explorer, Netscape Communicator y Opera.

La lógica de negocio, encargada de seleccionar las preguntas, verificar las respuestas de un usuario, etc., se ha programado mediante PHP en un servidor web Apache bajo el sistema operativo Linux.

Por último, para la lógica de datos, encargada de gestionar los datos, se ha empleado un servidor de bases de datos MySQL bajo el sistema operativo Linux.

Actualmente, tanto el servidor web como el servidor de bases de datos residen en la misma máquina, pero nada impide que residan en máquinas diferentes.

6. Conclusiones

Las primeras pruebas realizadas parecen indicar que la aplicación incentiva la realización de ejercicios y el aprendizaje de la asignatura. No

obstante, hay que conseguir que el alumno trabaje con la aplicación y que realice los ejercicios de forma conjunta a como va recibiendo los conocimientos en clase de teoría.

Actualmente, el alumno resuelve de forma anónima los ejercicios, aunque estamos planteando la posibilidad de contemplar en la nota de la asignatura la resolución de los ejercicios. No obstante, a pesar de que esto incentivaría a que el alumno entrara en la aplicación a resolver ejercicios, podría ocurrir que los alumnos tuvieran como único objetivo responder correctamente y no aprender.

También consideramos que la aplicación puede suponer una ayuda inestimable en el proceso de aprendizaje/enseñanza. Por un lado, permite que los alumnos puedan darse cuenta de los temas en los que mayores fallos cometen, para poder remediarlo antes de los exámenes. Por otro lado, los profesores pueden hacer mayor hincapié en los temas en los que los alumnos cometen más errores. En cualquier caso, pensamos que esta herramienta no sustituye a las clases presenciales teóricas y/o prácticas ni a los libros de texto, sino que simplemente los complementa. El objetivo que nos hemos marcado con el desarrollo de este sistema es lograr que el proceso de resolución de ejercicios sea lo más ameno posible.

7. Trabajos futuros

Al final de curso deseamos confirmar si los resultados estadísticos obtenidos por el sistema (temas y preguntas con mayor porcentaje de error, etc.) se corresponden con los resultados obtenidos en los exámenes, lo cual permitirá valorar el carácter “predictivo” de la aplicación.

Otro aspecto en el que vamos a trabajar es en ampliar el tipo de preguntas para que no sean únicamente de tipo test sino que incluyan la posibilidad de que el alumno escriba pequeños programas y la aplicación sea capaz de validar su corrección.

Otras mejoras que pensamos llevar a cabo tienen como objetivo aumentar las prestaciones del sistema y su facilidad de uso. Por ejemplo, pensamos incorporar un método adicional de puntuación en el que se tenga en cuenta el tiempo que ha necesitado un alumno para contestar una pregunta (cuanto menos tiempo, más puntuación) y un sistema adicional que permita importar las

preguntas y respuestas desde un fichero en formato texto plano.

Aunque esta aplicación ha sido pensada inicialmente para la asignatura *Fundamentos de la Programación*, como la hemos desarrollado con la capacidad de incluir distintas asignaturas y profesores simultáneamente, en el futuro pensamos también aplicarla a otras asignaturas.

Referencias

- [1] *Aprendizaje en la Web AutoMotivado (AWAM)*. Disponible en Internet (URL): <http://gplsi.dlsi.ua.es/awam/>. Usuario y clave de prueba para acceder como alumno: jenui/jenui.
- [2] Barchino, Roberto; Gutiérrez, J.M.; García, Elena; Hilera, J. Ramón. *EDVI: Un sistema de apoyo a la enseñanza presencial basado en Internet*. Actas de JENUI 2001, páginas 451-453.
- [3] Gayo, Daniel; López, Benjamín; Labra, José E. *Desarrollo del portal web de la E.U. de Ingeniería Técnica en Informática de Oviedo*. Actas de JENUI 2001, páginas 39-44.
- [4] Llopis, Fernando.; Pérez, Ernesto; Ortuño, Fernando. *Introducción a la programación. Algoritmos y C/C++*. Publicaciones de la Universidad de Alicante, 2000.
- [5] Martel, Ernestina A.; Ojeda, Carmen N.; Hernández, Pablo; Macías, Elsa M.; Monagas, Vidina. *Sistema de Gestión de Asignaturas en Entorno Web*. Actas de JENUI 2001, páginas 33-38.
- [6] Más, Ramón; Lacosta, Ignacio. *Aplicaciones de Internet a la Enseñanza: Un Sistema de Autoevaluación*. Actas de JENUI 2001, páginas 500-503.
- [7] Merelo, Juan Julián; Castillo, Pedro Ángel; Prieto, Alberto. *Integración de una asignatura en Internet: el caso de Diseño y Evaluación de Configuraciones*. Actas de JENUI 2001, páginas 51-56.