

Aplicaciones de Internet a la Enseñanza: Un Sistema de Autoevaluación

Ramon Mas, Ignacio Lacosta

Dept. de Matemàtiques i Informàtica
Universitat de les Illes Balears
07071-Palma de Mallorca
e-mail: Ramon.Mas@uib.es

Resumen

Esta aplicación pretende aprovechar las ventajas inherentes de internet para la creación de un sistema complementario de autoevaluación para los alumnos. El sistema se basa en el acceso remoto a listas de problemas generadas aleatoriamente a partir de una base de datos creada por el profesor de la asignatura. Para la selección de los problemas se utilizan criterios coherentes de dificultad. Cada alumno recibirá, a petición propia, un conjunto aleatorio de problemas junto con el intervalo de tiempo esperado para su resolución. Este intervalo temporal es fijado por el profesor y una vez extinguido podrán consultarse las respuestas correctas.

Introducción

Internet y los lenguajes para la transferencia de hipertexto (HTML, ASP, XTML...) posibilitan la consulta remota de volúmenes inmensos de información. Las fronteras geográficas y políticas se diluyen en el espacio de internet.

En un principio, internet se utilizó para difundir información a modo de publicación electrónica o como un catálogo o un periódico. La presentación comercial de una empresa u organización y sus productos o servicios continua siendo, hoy en día, el contenido más frecuente. A pesar de ello, los portales ofrecen ya muy variados servicios complementarios: páginas que ponen en contacto empresas demandantes de empleo con posibles candidatos, agencias que ponen a disposición del viajero la propia configuración de vuelos y

estancias, bancos electrónicos, agencias de valores...

Mediante esta aplicación proponemos la creación de un sistema complementario de autoevaluación para los alumnos. El profesor de la asignatura tendrá acceso a una colección de problemas para su gestión. La idea se basa en la asignación, de forma aleatoria, de una cantidad determinada de problemas que cumplan los criterios de dificultad establecidos. Cada alumno recibirá un juego aleatorio, y por tanto en mayor o menor medida diferente al del resto de los alumnos, de problemas que podrán consultarse (y responderse) a través de un navegador.

A pesar de la aleatoriedad del proceso de selección de las preguntas, se garantizan desviaciones mínimas en la dificultad total para cada alumno.

El alumno dispondrá de un plazo de tiempo, fijado por el profesor, pasado el cual, y hasta que se disponga una nueva lista, podrá consultar y auto evaluar la validez de sus respuestas.

Las páginas dinámicas: Personalización de contenidos

Las páginas dinámicas se caracterizan por continuos cambios en sus contenidos dependiendo de factores externos como pueden ser la identidad del visitante o el lugar desde el que es visitada. El usuario percibe un trato diferenciado, especial y, por tanto, más directo.

Muchos de los portales que ofrecen productos o servicios utilizan en mayor o menor grado técnicas de personalización del contenido y su presentación. Estas aplicaciones pueden

configurar la presentación al visitante según los gustos de éste, deducidos a partir de sesiones anteriores. Un ejemplo muy claro es el de una hipotética web con contenido publicitario: el usuario ha ido dejando constancia de sus gustos y preferencias por medio de encuestas, formularios, peticiones o guardando información acerca de los detalles de la navegación en sus servidor. A medida que el tiempo avanza y el conocimiento del usuario se hace más completo, la web puede personalizar sus páginas para especializar o concretar al máximo los artículos o servicios que ofrece, siempre según los gustos del consumidor.

Para la generación de las páginas dinámicas suelen utilizarse los CGI [1] (*Common Gateway Interface*) que facilitan el proceso de formularios, el acceso a bases de datos y otros servicios de valor añadido sobre HTML¹.

En la aplicación que se presenta, la generación dinámica de las páginas se ha realizado mediante programas CGI creados con el lenguaje de programación PERL [2].

La personalización en el aspecto y en la funcionalidad de las páginas se consigue en la medida en que un alumno sólo puede acceder a las opciones admitidas en su perfil. No verá las operaciones propias del profesor ni del administrador de la aplicación a pesar de que en todos los casos las páginas se generan a través del mismo CGI.

Perfiles de Usuarios

El entorno trata a los usuarios clasificándolos en tres categorías: administradores de la aplicación, administradores de asignatura y alumnos.

Cada uno de los perfiles tiene un conjunto de operaciones propias de sus funciones. Veamos a continuación una descripción de los diferentes grupos.

Administrador de aplicación

Es la única figura que no está ligada a ninguna asignatura. Puede gestionar las operaciones relativas a usuarios y asignaturas, tales como su creación, destrucción o mantenimiento. Véase la figura 1 para el detalle de las funcionalidades disponibles en esta configuración.



Figura 1. Administración de la aplicación

Administrador de asignatura

Generalmente coincide con la figura del profesor. Tiene la facultad de gestionar las listas de problemas de cada asignatura y es el único que puede lanzar el proceso de asignación de problemas a los alumnos y que tiene a su disposición los informes y la capacidad de realizar un seguimiento personalizado de cada usuario. Puede gestionar varias asignaturas. La figura 2 muestra la página de acceso a la opciones de gestión.



Figura 2. Administración de asignaturas

¹ No podemos olvidar tampoco la utilización de los *scripts* o de JAVA que, a diferencia de los CGI, se ejecutan sobre el ordenador cliente y quedan condicionados por las características del ordenador del usuario.

Alumnos

Puede ver los problemas y enviar las soluciones. Puede cursar diferentes asignaturas a la vez. Con el mismo código de acceso puede consultar las listas de problemas de todas las asignaturas en las que se ha matriculado (fig. 3.)



Figura 3. Acceso de los alumnos

Funcionalidades

En los siguientes apartados procederemos a una breve enumeración de las principales funcionalidades del sistema que se presenta.

Administración de listas de problemas

Cuando el administrador de la asignatura inicia el proceso de generación de listas los alumnos pierden la posibilidad de acceder a la colección antigua. Se supone que el plazo para la consulta y solución se da por concluido para todos los usuarios.

El profesor selecciona la cantidad de problemas para el sorteo y fija el intervalo de tiempo en días, horas y minutos. También especifica el desvío permitido en la media de la dificultad. Cada problema tiene una dificultad entre 1 y 5, asignada de forma arbitraria. En el proceso de sorteo de problemas se busca conseguir que los alumnos reciban todos una carga de trabajo similar. Por tanto se calcula el promedio de dificultad de todos los problemas disponibles y el promedio de los problemas que ha recibido un alumno. Se busca un subconjunto de problemas cuyo promedio se encuentre dentro del margen de desvío fijado.

Mecanismo de control temporal

Cuando el alumno accede por primera vez a su colección personal de problemas pone en marcha el mecanismo de control temporal. A partir de ese momento dispondrá de un tiempo fijado por el profesor para redactar todas las soluciones. Durante ese lapso de tiempo el alumno puede establecer tantas sesiones como desee. En todo momento conocerá cuándo expira el plazo para el envío de soluciones. Cuando se cumpla ese tiempo sólo podrá acceder en modo consulta a los enunciados y a las soluciones por él propuestas. Los mecanismos de medición de tiempo funcionan con el reloj interno del servidor donde se ejecutan los *scripts*.

Formato de los problemas

Un problema, necesariamente, se especifica a través de un número identificativo (único para cada asignatura), un enunciado, una solución y una medida de su dificultad (valorada de 1 a 5).

Opcionalmente, los problemas pueden ir acompañados de imágenes explicativas (en los ya tradicionales formatos *gif* o *jpg*).

Seguimiento de los alumnos

El administrador de la asignatura puede, en todo momento, consultar si un alumno ha accedido a la lista de problemas, y, si lo ha hecho, cuándo fue la primera vez y cuándo termina el plazo. También es capaz de ver las soluciones enviadas a cada problema. Todo ello desde listados “sensibles”, dotados con enlaces para navegar hasta la información deseada. Es decir, que partiendo de un listado general de alumnos podemos ir consultando características de uno en uno y llegar hasta las soluciones de los problemas en curso.

Correo electrónico

La aplicación incluye la posibilidad de que todo usuario pueda utilizar el correo electrónico para comunicarse con otros alumnos y con el mismo profesor (aunque la implementación de los clientes de correo permiten las funcionalidades POP3 y SMTP).

Balanceo de carga y distribución de la aplicación

El diseño modular de la aplicación permite su distribución entre diferentes servidores: el motor de bases de datos puede residir en un servidor mientras que el servidor web se encuentra en otra plataforma y el servidor de correo en otra distinta. Esta posibilidad puede utilizarse para optimizar los recursos hardware o para repartir la carga de proceso.

Conclusiones y trabajo futuro

La demostración que se presenta ofrece un mecanismo complementario para la evaluación del alumnado. No pretende ser un mecanismo de control exhaustivo por parte del profesor sino que intenta desplazar hacia el alumnado el control escalonado de sus progresos, dándole la posibilidad de auto-evaluarse.

Hasta este momento, la aplicación se ha mostrado muy estable para la gestión de grupos moderados de alumnos (del orden de 60) aunque no se descarta su aplicación a grupos más numerosos. En caso de verse mermado el rendimiento global de la aplicación puede recurrirse a una repartición de carga del proceso distribuyendo los módulos de la aplicación entre diferentes servidores.

A pesar de que el acceso a las listas de problemas se hace mediante clave y contraseña personalizadas, la cuestión de seguridad puede mejorarse mediante la utilización de un servidor seguro. También queda pendiente la implementación de un módulo para preguntas-respuestas tipo test, con lo cual se podría realizar una corrección automática.

Referencias

[1] Gundavaram S., CGI Programming on the World Wide Web, 2nd Edition, O'Reilly & Associates, ISBN 1565924193, 2000

[2] Deep J., Developing CGI Applications with Perl, Wiley Computer Publishing, ISBN:0471141585, 1996