

## Sistema de Evaluación Automática Vía Web en Asignaturas Prácticas de Ingeniería

F. Gómez-Estern, M. López-Martínez y D. Muñoz de la Peña

*Dpto. Ingeniería de Sistemas y Automática.  
Escuela Superior de Ingenieros. Universidad de Sevilla.  
Camino de los Descubrimientos s/n. 41092-Sevilla.  
e-mail: fabio@esi.us.es, mlm@esi.us.es, dmunoz@us.es*

Resumen: En este trabajo se presenta una nueva herramienta web para educación que permite automatizar la recogida y evaluación de ejercicios prácticos de diferentes disciplinas de ingeniería con la complejidad típica que la formación técnica requiere. Una de las principales características de la herramienta propuesta es la posibilidad de personalizar los ejercicios para cada alumno. El sistema realiza una comparación funcional de las soluciones propuestas por los alumnos con las soluciones correctas proporcionadas por el profesorado y en función de los resultados obtenidos asigna una calificación al alumno de forma automática. La plataforma permite al profesor implementar técnicas innovadoras de docencia que fomentan el autoaprendizaje en cursos numerosos. Se presentan los resultados de uso del sistema de evaluación en dos cursos, uno de Teoría de Sistemas y otro de Fundamentos de Informática. *Copyright © 2010 CEA.*

Palabras Clave: E-learning, LCMS, Evaluación automática, Ingeniería de Sistemas y Automática, TIC en la educación.

### 1. INTRODUCCIÓN

En los últimos años, Internet ha sido el medio por excelencia que ha permitido hacer realidad un paradigma de trabajo en el que diferentes personas de diversas partes del mundo pueden colaborar en un proyecto común. Este paradigma de trabajo ha sido posible no sólo por la capacidad de compartir información de forma eficiente y fácil, sino también por las posibilidades de colaboración que ofrecen nuevas herramientas como las redes sociales, foros, blogs y chats. Todas estas herramientas están englobadas en un mismo concepto genérico denominado sistema de gestión de contenidos o CMS (*content management system*).

Atraídos por las posibilidades que ofrecen, las instituciones de enseñanza comienzan a usar estos CMS como herramientas de aprendizaje a través de la web (e-learning). De esta forma, surgen los llamados sistemas de gestión de contenidos para el aprendizaje (LCMS) o simplemente sistemas de gestión del aprendizaje (LMS). Un LMS genérico permite registrar usuarios, organizar cursos, almacenar información de los usuarios y generar informes de gestión para el profesor. Suelen incluir también herramientas para la comunicación entre participantes de un mismo curso. Las mejoras obtenidas en los últimos años en el manejo y la accesibilidad de dichos sistemas, hacen que se hayan ampliado las posibilidades de formación a todos los niveles, superándose la barrera tecnológica que en un principio era el principal inconveniente del e-learning.

El empleo de sistemas de enseñanza on-line permite una reducción de los requisitos de movilidad de los alumnos, con las consiguientes ventajas para las empresas con empleados en procesos de formación, una interacción profesor-alumno más estrecha y con más recursos (foros, e-mail, FAQ, disponibilidad

inmediata de apuntes, avisos, notas, planificación, etc) y por último, la introducción de nuevos elementos de evaluación (herramientas de test automático, sistemas de entrega y seguimiento de trabajos de curso, etc).

Actualmente existe un gran número de plataformas que aportan mejoras a la experiencia de profesores y alumnos en el proceso educativo. Entre ellas cabe mencionar Moodle y WebCT por lo extendido de su uso, aunque hay muchas otras como por ejemplo Blackboard, eCollege, Desire2Learn, Skillfactory, Dokeos, SakaiProject, Atutor, Ilias, Claroline, Saba-Learning, etc. Respecto a las posibilidades que se abren a la hora de hacer uso de estos sistemas, podemos citar los trabajos de (Kapur and Stillman, 1997; Poindexter and Heck, 1999; Heck *et al.*, 2000), donde los autores dan una visión general de lo que se puede hacer en un curso académico empleando la web.

En el seno de las escuelas de Ingeniería, tales herramientas pueden ser de especial utilidad, como se indica en (Dormido and Torres, 2005). En ellas, un gran número de profesores trabaja en el desarrollo de herramientas con nuevas características adaptadas a este campo. En particular, se han desarrollado un gran número de trabajos basados en laboratorios remotos que permiten a los alumnos realizar prácticas con equipos reales a través de Internet. Cabe mencionar los trabajos (Sanchez *et al.*, 2004, 2006), donde los autores usan programación en Java (Easy Java Simulations) para crear laboratorios de control virtuales y remotos. Otra plataforma para la creación de laboratorios virtuales se presenta en (Cedazo *et al.*, 2007), bautizada con el nombre *Ciclope*. Otros tele-laboratorios de control automático conocidos son los de (Casini *et al.*, 2003, 2004; Torres *et al.*, 2006) a los que cabe añadir los presentados en la revista RIAI (Zuluaga *et al.*, 2005; Guzmán *et al.*, 2005). Por otro lado, la tendencia actual es la de hacer una extensión del concepto de

laboratorio virtual al de tele-laboratorios distribuidos para su uso conjunto por alumnos pertenecientes a diferentes universidades, como se hace en (Kerer *et al.*, 2005; Domínguez *et al.*, 2005; Jiménez *et al.*, 2005).

Los sistemas LMS son muy eficientes para recoger y almacenar el trabajo del alumno, sin embargo, en general ofrecen una capacidad de evaluación automática limitada a la evaluación de cuestionarios de respuesta múltiple (en diferentes modalidades). En particular, los mecanismos de evaluación ofrecidos no permiten la evaluación personalizada y además no están diseñados para disciplinas de ingeniería con una fuerte carga matemática. En el campo de la algoritmia y la programación, hay incluso menos aplicaciones y herramientas disponibles aunque cabe destacar que se han presentado diversos laboratorios virtuales específicos que sí disponen de herramientas de autoevaluación en (Rodríguez *et al.*, 2007a,b,c; García *et al.*, 2009), los tres primeros para programar periféricos y el cuarto para programar un microprocesador virtual basado en el MC88110. Sin embargo, en estos casos, las herramientas son específicas y están completamente cerradas, no disponiendo de la flexibilidad necesaria para ser fácilmente modificadas y empleadas en otras asignaturas.

Teniendo en cuenta las limitaciones de los sistemas actuales, se ha desarrollado un sistema de e-learning específico para ingeniería que permite automatizar la recogida y evaluación de ejercicios prácticos de diferentes disciplinas con la complejidad típica que la formación técnica requiere y con la posibilidad de personalizarlos para cada alumno. El uso de la herramienta presentada permite, entre otras cosas, incrementar la periodicidad de los exámenes y pruebas, ya que en la Universidad, la masificación y la escasez de recursos evaluadores (profesores) suelen ser los principales obstáculos para aumentar dicha frecuencia. Del mismo modo, también permite establecer un criterio único y homogéneo a la hora de evaluar al alumnado que se puede mantener invariable con el paso de los años y los cambios de profesorado.

La aplicación fue inicialmente diseñada para asignaturas de Teoría de Sistemas y Control Automático (en (Muñoz de la Peña and Gómez-Estern, 2009) se explica su uso en tales asignaturas y se ofrece una descripción detallada del motor interno). Posteriormente se extendió a asignaturas de programación en C y Matlab, ampliando drásticamente el abanico de potenciales usuarios. A continuación, se describirá la nueva herramienta web de evaluación y se presentarán los resultados de dos casos de estudio de la Escuela de Ingenieros de la Universidad de Sevilla.

## 2. METODOLOGÍA

La herramienta presentada en este trabajo es un sistema de gestión de contenidos para el aprendizaje que además ofrece la funcionalidad de evaluar automáticamente ejercicios prácticos de diferente naturaleza. Para ello, la herramienta dispone de las siguientes características:

- Existe una metodología sencilla y flexible para la creación y almacenado de nuevos ejercicios.
- Existe un mecanismo simple y robusto para la entrega de los trabajos de los alumnos, así como de gestión de los mismos.
- Existe un método automático para compilar, ejecutar y evaluar los trabajos de los alumnos (calcular la nota).

Los dos primeros puntos (recogida y almacenamiento de información) son comunes a los sistemas de LMS existentes. Aun así, se ha desarrollado una plataforma particular utilizando tecnología estándar con características que la hacen atractiva desde el punto de vista de la gestión administrativa, tales como:

- Creación de grupos.
- Gestión de usuarios: identificación segura de alumnos y profesores.
- Creación, coordinación y almacenamiento de bibliotecas de ejercicios para posteriores cursos académicos, que pueden ser usados en diferentes asignaturas y por distintos profesores.
- Asignación de ejercicios a grupos.
- Control automático de las fechas límite de entregas de los ejercicios.
- Acceso a las soluciones y calificaciones de cada alumno para una revisión manual posterior, en caso de que el alumno lo requiriera.

La principal aportación del sistema desarrollado se encuentra en el campo de la automatización de la evaluación. Se ha desarrollado un sistema con dos objetivos fundamentales: personalizar los ejercicios para cada alumno introduciendo variabilidad en las preguntas a partir de un identificador personal (como el número de matrícula), y evaluar ejercicios basándose en cálculos matemáticos complejos, simulaciones numéricas y comprobación de algoritmos. El proceso de evaluación, lo más importante de este sistema, emplea el principio de evaluación funcional, o de “caja negra”, consistente en coordinar los siguientes elementos:

- La solución del alumno, enviada en formato de texto a través de Internet. Dicho texto debe redactarse conforme a unas normas sintácticas que permitan ser tratadas *como instrucciones de ordenador* escritas en algún lenguaje de programación determinado (C, Matlab, Ensamblador...). El alumno no tiene por qué ser consciente de este hecho, ni conocer el lenguaje en su totalidad, sólo las reglas básicas que se explicarán en el enunciado.
- El algoritmo del profesor, escrito en el mismo lenguaje de programación, que es capaz de producir una entrada de información para el código (el conjunto de instrucciones) del alumno, ejecutar éste, y observar la información de salida tras el procesado del alumno. Dicha información permitirá calcular si la solución del alumno es correcta funcionalmente, es decir, si como objeto de proceso de información funciona correctamente.

En resumen, el sistema presentado es capaz de recoger código desarrollado por el alumno para después compilarlo y ejecutarlo junto con código desarrollado por el profesor. De esta forma, el profesor diseña el código de evaluación de la práctica, el alumno inserta su solución, y el sistema compila y ejecuta el programa resultante, obteniendo y almacenando la nota del alumno. Esta operación se realiza para cada una de las soluciones propuestas por los alumnos de un grupo determinado que está realizando una práctica concreta de una asignatura definida y empleando un lenguaje de programación específico.

### 2.1 Ventajas del sistema de evaluación

La posibilidad de diseñar un código que se compila junto con el código del alumno ofrece la flexibilidad necesaria para personalizar prácticas y para poder evaluar el trabajo de cualquier

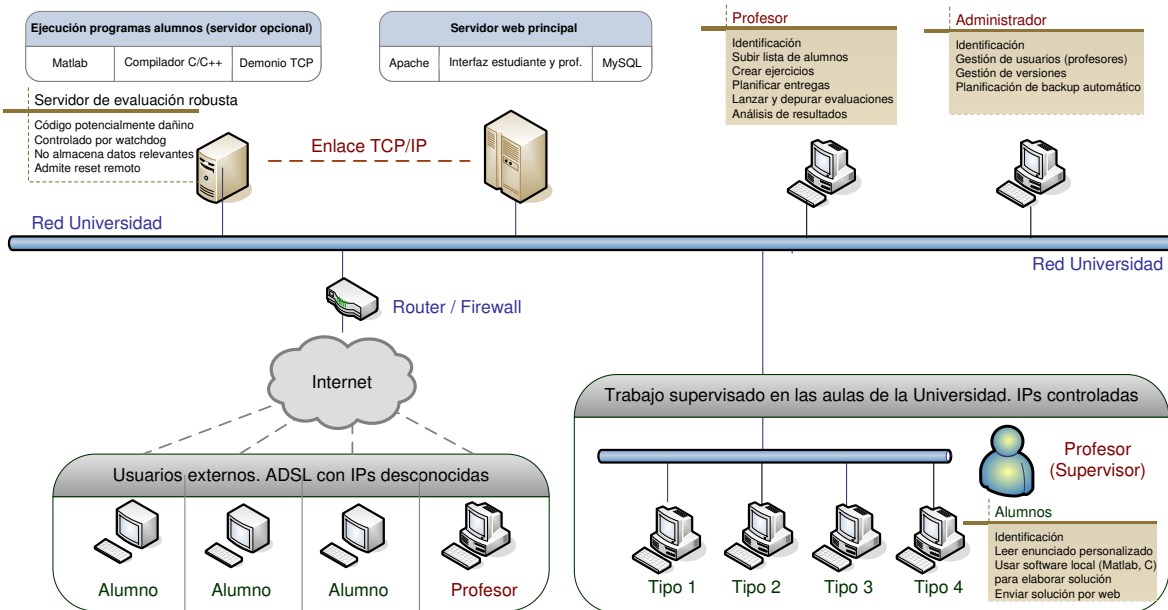


Figura 2. Arquitectura del sistema

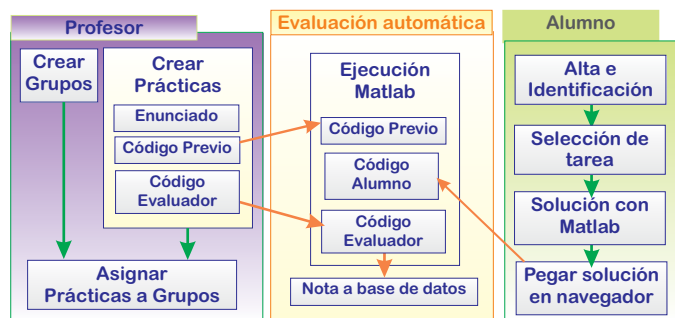


Figura 1. Funcionamiento del sistema de evaluación automática

disciplina con una base matemática o algorítmica. Aunque a primera vista cabe pensar que este sistema se restringe a asignaturas relacionadas con la programación y la informática, nada más lejos de la realidad. En el caso de un problema de análisis, la respuesta del alumno puede consistir en la definición de unas variables que almacenan los resultados de un determinado problema. El programa desarrollado por el profesor puede calcular los resultados que deberían obtenerse para el conjunto de parámetros personalizados del alumno y compararlo con la solución entregada. En el caso de problemas de diseño, es posible incluso comprobar si la solución presentada por el alumno cumple o no un conjunto de restricciones mediante simulación (por ejemplo en problemas de diseño de controladores para sistemas no lineales). En los casos de estudio se ilustrará cómo se ha empleado intensivamente la herramienta en asignaturas de fuerte componente matemático y teórico, como la Teoría de Sistemas. En la figura 1 se muestra la estructura lógica del sistema de evaluación en el caso en el que se emplee Matlab (lenguaje matemático, de cálculo numérico y simulación empleado en numerosos campos de ingeniería).

El proceso de corrección es instantáneo y se puede repetir cuantas veces se desee. Es particularmente útil el hecho de que se puede cambiar el código de corrección en línea para determinar por ejemplo la sensibilidad de las notas con un margen de error  $\alpha$ . Este hecho también facilita el desarrollo del código de

corrección. Además, se ha diseñado un procedimiento robusto de evaluación, por un lado prohibiendo el uso de funciones potencialmente peligrosas para la integridad del servidor, y por otro lado, detectando posibles errores y bucles infinitos en el código del alumno.

Respecto a las dificultades encontradas, cabe reseñar que el servidor desarrollado proporciona una nota numérica que no tiene en cuenta posibles errores tipográficos, confusión en los parámetros u otros factores que puedan adular los resultados. Esto significa que una cierta información se pierde en el proceso de evaluación. Sin embargo es una herramienta muy adecuada para evaluar a una gran cantidad de alumnos bajo un único criterio. Esto es muy difícil de conseguir con un método tradicional de evaluación.

### 2.2 Evaluación competitiva

Existe la posibilidad de calificar el trabajo del alumno no por su calidad individual, sino tras compararlo con el resto. Este paradigma se ilustra en la figura 3 y funciona del siguiente modo: se ejecuta el código del alumno y el código evaluador del profesor, obteniéndose una figura de mérito que se almacena en la base de datos como nota provisional. En un problema de diseño de controladores, esta figura podría ser el tiempo de subida en bucle cerrado. Una vez calculada la nota provisional para todos los alumnos, el sistema filtra los resultados inválidos, ordena las calificaciones y calcula la nota final para cada alumno según su posición en la lista ordenada, dentro de un intervalo de notas configurado por el profesor. Esta funcionalidad selecciona como una opción en el interfaz de creación de prácticas.

### 2.3 Protección frente al fraude

Una dificultad del sistema es la posibilidad de ayuda no autorizada de alumnos más aventajados al resto. Los riesgos son mayores cuando el trabajo se realiza en casa del estudiante y sin control del profesorado. Se trata de una limitación inherente al

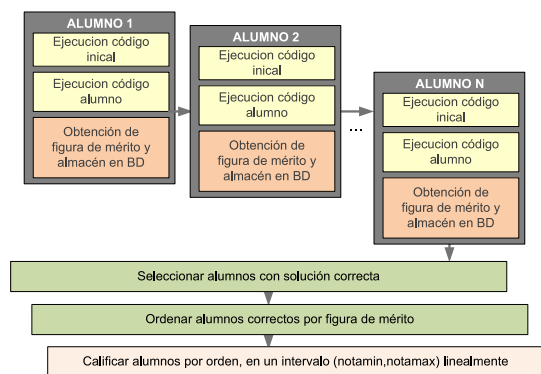


Figura 3. Funcionamiento del sistema de evaluación automática

aprendizaje basado en proyectos, y es común a las herramientas de e-learning mencionadas. Nuestra aplicación, mitiga este riesgo mediante la personalización de ejercicios, como es el caso de la parametrización individual de las funciones de transferencia  $G(s)$  del apartado 4.2. Dicha parametrización es posible gracias al acceso al número de matrícula o ID del alumno desde el Código Previo y el Código Evaluador del Profesor (ver figura 1). A este mecanismo se añade la posibilidad de supervisar el trabajo en el aula, según se ilustra en la figura 2. En este escenario el sistema permite filtrar los envíos en función de la IP de origen, garantizando la presencia de los alumnos en las aulas, mientras que el profesor se encarga de verificar la identidad del alumno, que se muestra en pantalla en todo momento.

En las siguientes secciones se describe la arquitectura del sistema junto con una breve descripción de los interfaces de usuario.

### 3. ARQUITECTURA DEL SISTEMA

La herramienta consta de dos procesos independientes, un programa de ordenador, que se ejecuta en un servidor web dedicado situado en la Escuela Superior de Ingenieros, y un servidor independiente no visible desde la red, de apoyo a la evaluación automática (aunque, a costa de una ligera pérdida de seguridad, ambos procesos se pueden unificar en una única máquina). El hardware, sistema operativo y aplicaciones empleadas son:

- Servidor de rack HP Proliant DL-1400, 2 GB de memoria RAM, 160Gb de disco duro.
- Sistema operativo Windows 2003 Server R2.
- Apache 2.2.6, PHP 5 y MySQL Server 5.0.
- Matlab 7.3.
- Dev-C++.

Los usuarios (profesores y alumnos) acceden mediante un navegador convencional de Internet. En la figura 2 se ilustra la arquitectura de red del sistema. Los alumnos se encuentran en las aulas de prácticas de la escuela o facultad (en el caso en que se desee controlar la identidad del alumno), o en el exterior.

Respecto de las tecnologías software empleadas, destacamos el empleo de Matlab, una herramienta de cálculo numérico matricial programable sobradamente conocida en el área de Automática. Como apoyo a la docencia, se emplea principalmente en el análisis de sistemas, la síntesis de controladores (con el soporte del *Control System Toolbox*), y dentro de las asignaturas de primeros cursos de Informática y Programación. Como herramienta de realización de las prácticas su efectividad

Figura 4. Ventana de entrada al sistema

es innegable y a menudo se emplea el propio Matlab para evaluar la validez de las soluciones de los alumnos.

Además de Matlab, en el servidor se ofrece la posibilidad de realizar prácticas basadas en el lenguaje de programación C, usando el compilador de libre distribución Dev-C++. Cabe destacar, que es posible desarrollar módulos que permitan la realización de prácticas para cualquier otro lenguaje de programación.

Para desarrollar el servidor se ha utilizado ActiveX. ActiveX emplea el modelo de objetos distribuidos COM de Microsoft para la automatización de aplicaciones. En el presente proyecto se utiliza esta tecnología de manera similar a (Zuluaga *et al.*, 2005) con el fin de que el servidor web pueda realizar operaciones en el espacio de trabajo de Matlab y compilar código C a través de llamadas a Dev-C++.

La implementación software del servidor se basa en las siguientes tecnologías

- Script de servidor PHP. Herramienta de software libre que permite la ejecución de programas en un servidor web en respuesta a peticiones HTTP emitidas por un navegador.
- Servidor de base de datos MySQL. Servidor en red de base de datos relacional para la gestión de los alumnos, las bibliotecas de prácticas y las calificaciones.
- Apache. Servidor web que atiende a las consultas HTTP de los navegadores y genera las respuestas a través de la ejecución de scripts PHP.

Los servicios Apache–PHP–MySQL forman la arquitectura de servidores web dinámicos más empleada actualmente en el mundo (65 % de penetración frente al 35 % de ASP de Microsoft) y es reconocido por su estabilidad y portabilidad a distintos sistemas operativos.

#### 3.1 Interfaz del alumno.

El flujo de trabajo del alumno aparece esquematizado en la parte derecha de la figura 1. Inicialmente, el alumno debe identificarse a través de la página web del servidor de docencia, <http://bono.us.es/sdocencia>, mediante nombre de usuario y clave (figura 4). A continuación se muestra una lista con los trabajos pendientes de entrega (que pueden pertenecer a distintas asignaturas). La lista depende de los grupos en los que aparece dicho alumno y las asignaciones (asociaciones) de prácticas que estén activas en cada uno de estos grupos (figura 5).

El alumno seleccionará uno de estos trabajos, y a continuación se le mostrará una página con un enunciado en PDF, y un



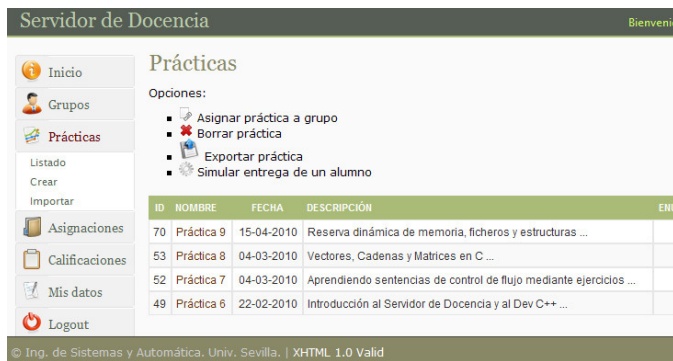


Figura 5. Ventana de selección de trabajos



Figura 6. Ventana de envío de soluciones

campo de texto sobre HTML para incorporar su solución al problema (figura 6). Dicha solución debe estar escrita usando una sintaxis compatible con el lenguaje que se emplee para corregir las prácticas (C, Matlab, Cesium,...) y cumplir unas normas unívocamente definidas en el enunciado de la práctica.

Un ejemplo práctico sería la realización de una función en lenguaje C que determine si una matriz de enteros dada es un Sudoku. La solución presentada por el alumno se muestra en la figura 6.

### 3.2 Interfaz del profesor.

El flujo de trabajo del profesor se muestra en la figura 1 (izquierda). El interfaz consiste en un menú que permite introducir todos los parámetros para que se lleven a cabo las prácticas, y una serie de opciones para calcular la evaluación y mostrar los resultados. Las opciones principales son:

- Crear nuevo grupo. Para dar de alta nuevos alumnos, grupos y asignaturas.
- Listar alumnos y grupos.
- Crear prácticas, definiendo el programa evaluador y una serie de opciones (ver figura 7).
- Listar y modificar prácticas.
- Lanzar evaluación fuera de línea. Realiza la conexión con Matlab o Dev-C++ y evalúa todos los trabajos entregados correspondientes a una asignatura seleccionada.
- Visualizar calificaciones. En distintos formatos, permite la descarga de un fichero Excel que puede ser complementado con otro tipo de calificaciones. Además, permite el acceso al código entregado por cada alumno y realizar

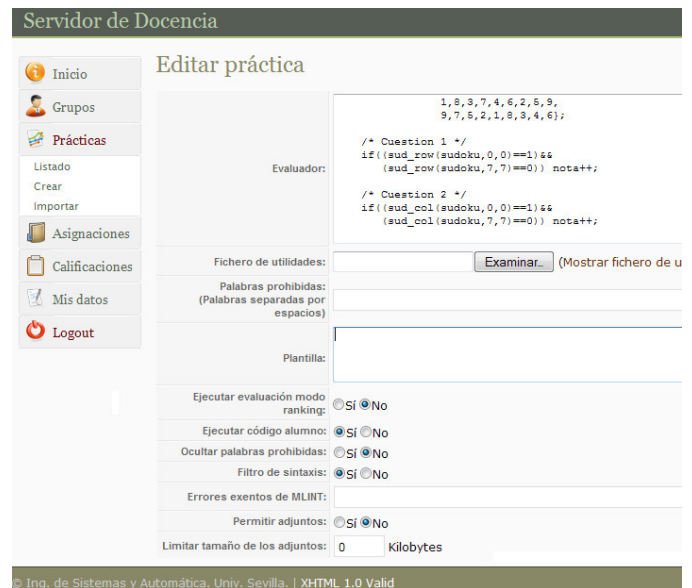


Figura 7. Ventana de creación y configuración de prácticas

modificaciones sobre él (por ejemplo, para recuperar errores de sintaxis).

En todo momento, el profesor puede volver a lanzar evaluaciones de un grupo. Si se observa que las calificaciones no son coherentes o se desea mejorar el proceso, se podrá optar por refinar el programa que calcula la nota o revisar y en su caso corregir a mano los errores particulares de los alumnos, para finalmente volver a lanzar la evaluación.

El sistema es también capaz de rechazar aquellos envíos con errores de sintaxis, o que usan funciones potencialmente peligrosas para la integridad del servidor. La arquitectura del sistema ha sido desarrollada para permitir que se asignen diversos ejercicios en cada sesión con el fin de evitar el fraude.

## 4. CASOS PRÁCTICOS

En esta sección se presentan los resultados de utilizar el sistema de evaluación en dos cursos, uno de teoría de sistemas y otro de fundamentos de informática. Para los casos prácticos presentados, el procedimiento de evaluación se realizó en menos de tres minutos.

### 4.1 Caso práctico 1: Fundamentos de Informática.

Una situación donde la automatización de la evaluación resulta efectiva es en las asignaturas de introducción a la informática y la programación, que en numerosas escuelas se adscriben al área de Automática. En este contexto nos centraremos en las asignaturas en las que se imparte programación en entorno Matlab (como es el caso de la asignatura Informática de Ingeniería Aeronáutica en la Universidad de Sevilla) y en lenguaje C (Fundamentos de Informática de Ingeniería Industrial y de Ingeniería Química). El esquema de evaluación de un trabajo de programación se basa en la comprobación de que una función programada por un alumno realiza el proceso especificado por el profesor en forma de “caja negra”. Para ello se plantea el siguiente esquema:

- El profesor propone una función a programar por el alumno, especificando sus parámetros de entrada y salida, y el

procesado que ha de hacerse con los mismos. Ejemplo: cálculo de la media de un vector que se ha pasado como parámetro.

- El profesor implementa la función o algoritmo que resuelve el problema, que denominaremos S1.
- El alumno implementa su propia versión de su solución, S2. Esta función debe contener un encabezado estándar definido por el profesor con el fin de que la llamada y el paso de parámetros sean inequívocos.
- El profesor ejecuta un programa que genera aleatoriamente unos datos de entrada (en el ejemplo, el vector cuya media se desea calcular). El programa llama a S1 y S2 con dichos datos y compara los resultados.
- Se genera una calificación en función de la coincidencia de resultados.

A continuación se presenta el enunciado de la octava práctica que fue realizada en el curso académico 2008-2009, en la asignatura de Fundamentos de Informática de 1º de Ingeniería Industrial en la Universidad de Sevilla. Dicha asignatura es troncal y dicho curso constaba de 615 alumnos divididos en 12 grupos diferentes. En esta asignatura se realizaron 5 prácticas diferentes gestionadas a través del servidor. Las prácticas constan de una serie de funciones que el alumno ha de diseñar y codificar en lenguaje C. Al final de la clase de laboratorio el alumno envía el código fuente diseñado a través de la aplicación.

### Enunciado de ejemplo.

Determine si una matriz de enteros 9x9 dada, es un Sudoku. Sudoku es un pasatiempo matemático, donde hay que rellenar una matriz 9x9 con enteros del 1 al 9. La matriz está dividida a su vez en 9 submatrices 3x3. Dada una matriz, se dice que ésta es un Sudoku si para cada posición de la matriz, el número correspondiente no se repite ni en la misma fila, ni en la misma columna ni en la misma submatriz 3x3.

- Cuestión 1:** Realice una función, tal que dada una matriz y una posición específica (fila y columna), compruebe si el número en esa posición de la matriz se repite en la misma fila. La función devuelve 0 si hay repetición. Si no, devuelve 1.  
`int sud_row(int * M, int i, int j);`
- Cuestión 2:** Realice una función, tal que dada una matriz y una posición específica (fila y columna), compruebe si el número en esa posición de la matriz se repite en la misma columna. La función devuelve 0 si hay repetición. Si no, devuelve 1.  
`int sud_col(int * M, int i, int j);`
- Cuestión 3:** Realice una función, tal que dada una matriz y una posición específica (fila y columna), compruebe si el número en esa posición de la matriz se repite en la submatriz 3x3 a la que pertenece. La función devuelve 0 si hay repetición. Si no, devuelve 1.  
`int sud_sub(int * M, int i, int j);`
- Cuestión 4:** Realice una función, tal que dada una matriz y una posición específica (fila y columna), compruebe si el número en esa posición de la matriz verifica las tres propiedades anteriores. La función devuelve 1 si las tres propiedades se cumplen. Si no, devuelve 0.  
`int sud_cas(int * M, int i, int j);`
- Cuestión 5:** Realice una función, tal que compruebe si una matriz dada es un Sudoku. La función devuelve 1 si la matriz es un Sudoku. Si no, devuelve 0.

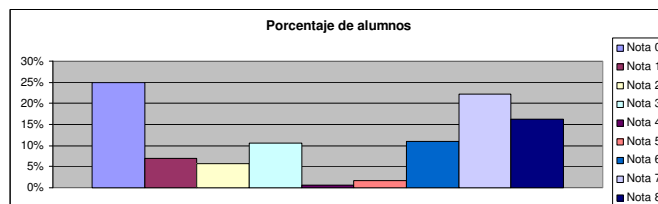


Figura 8. Distribución porcentual de las notas obtenidas.

```
int sud(int * M);
```

El código de evaluación del profesor para este ejercicio práctico, viene principalmente dado por la siguiente función.

```
int evaluacion(void) {
    int nota=0;
    int sudoku[81] =
        {1,2,3,4,5,6,7,8,9,
         4,5,6,3,2,7,8,9,6,
         7,8,9,3,2,7,8,9,6,
         2,4,5,3,2,7,8,9,6,
         3,4,5,3,2,7,8,9,6,
         5,4,5,3,2,7,8,9,6,
         6,4,5,3,2,7,1,4,3,
         8,4,5,3,2,7,4,4,6,
         9,4,5,3,2,7,7,8,9};

    int sudokul[81]=
        {7,9,8,6,3,2,5,1,4,
         5,3,1,8,7,4,6,9,2,
         4,6,2,1,9,5,7,8,3,
         2,4,6,5,8,1,9,3,7,
         8,1,7,9,6,3,4,2,5,
         3,5,9,4,2,7,8,6,1,
         6,2,4,3,5,9,1,7,8,
         1,8,3,7,4,6,2,5,9,
         9,7,5,2,1,8,3,4,6};

    /* Cuestion 1 */
    if((sud_row(sudoku,0,0)==1)&&
        (sud_row(sudoku,7,7)==0)) nota++;

    /* Cuestion 2 */
    if((sud_col(sudoku,0,0)==1)&&
        (sud_col(sudoku,7,7)==0)) nota++;

    /* Cuestion 3 */
    if((sud_sub(sudoku,0,0)==1)&&
        (sud_sub(sudoku,7,7)==0)) nota++;

    /* Cuestion 4 */
    if((sud_cas(sudoku,0,0)==1)&&
        (sud_cas(sudoku,7,7)==0)) nota++;

    /* Cuestion 5 */
    if((sud(sudoku)==0)&&
        (sud(sudokul)==1)) nota++;

    return nota;
}
```

Esta función de evaluación simplemente verifica si cada una de las cinco funciones propuestas al alumno, funciona correctamente con dos matrices diferentes. En caso de que una función sea correcta, la nota se incrementa en un punto.

**Calificaciones de los alumnos.**

En la figura 8 se presenta la distribución porcentual de las notas obtenidas, evaluadas entre 0 a 8. En el ejercicio práctico había tres cuestiones más, que se han omitido por motivos de claridad en la exposición. Al analizar los resultados obtenidos, se puede observar que hay una clara separación entre los ejercicios suspensos (nota de 0 a 3) y los aprobados (nota de 6 a 8). Particularmente, el 48 % de las calificaciones corresponde a ejercicios suspensos. Sin embargo, este porcentaje no es realista en absoluto, ya que el evaluador asigna 0 al ejercicio del alumno cuando el proceso de su evaluación dura más de cinco segundos, lo cual implica que el código de alguna función del alumno es incorrecto (probablemente un bucle infinito). La imposibilidad de continuar la evaluación hace que no se tenga en cuenta la nota de las restantes funciones. Esto se ha solventado sacando notas parciales para cada función <sup>1</sup>.

**4.2 Caso práctico 2: Teoría de Sistemas**

Otro escenario de docencia donde resulta muy conveniente el empleo de Matlab sería en la calificación de prácticas de diseño de controladores. En este contexto, se propone el siguiente esquema:

- El sistema propone una función de transferencia G(s) (puede ser individualizada para cada alumno) y una estructura de controlador, por ejemplo PID o red de avance.
- El sistema propone al alumno unas especificaciones a cumplir por el controlador en bucle cerrado: error en régimen permanente, márgenes de fase y ganancia, etc.
- El alumno diseña el controlador empleando los métodos explicados en clase, obteniendo las constantes del controlador (Kp, Td, Ti en el caso de un controlador PID).
- El alumno verifica que el controlador cumple las especificaciones, mediante comandos de simulación (“lsim”) o con instrucciones específicas del Control System Toolbox, (p. ej. “margin”). A continuación, entrega el trabajo.
- El sistema evalúa automáticamente si se cumplen las especificaciones empleando las mismas herramientas de Matlab (principalmente simulaciones).

A continuación se presenta el enunciado de una de las prácticas que fueron realizadas en el curso académico 2007-2008, en la asignatura de Teoría de Sistemas de 2º de Ingeniería Industrial en la Universidad de Sevilla. Dicha asignatura es troncal y dicho curso constaba de 366 alumnos divididos en cuatro grupos diferentes.

La práctica consta de una serie de ejercicios que el alumno ha de resolver usando Simulink durante la clase de laboratorio. Al final de la clase de laboratorio el alumno entrega una memoria con los resultados. Los parámetros que definen el ejercicio que debe realizar el alumno dependen de su número de DNI. La mayor parte de las respuestas de las prácticas son numéricas (como por ejemplo el tiempo de subida de un sistema o la amplitud de la salida en régimen permanente frente a una entrada escalón). La corrección manual de un gran número de prácticas con resultados numéricos diferentes para cada una de ellas conllevaba una gran carga de trabajo. Además, tan sólo se distinguían cuatro posibles notas, *suspenseo*, *mal*, *regular* y

<sup>1</sup> Concretamente se programaron varios evaluadores que hacían llamadas selectivas a una u otra función, de manera que el bloqueo de una de ellas no imposibilitara la ejecución de las demás.

**Tabla 1: Resultados a entregar de la práctica**

$\omega$ (rad/s)	$Y_0$ (dB)	$\Delta$ (s)	$\phi$ (rad)
<b>p1</b> $0.01/\tau=$			
<b>p2</b> $0.1/\tau=$			
<b>p3</b> $0.3/\tau=$			
<b>p4</b> $1/\tau=$			
<b>p5</b> $3/\tau=$			
<b>p6</b> $10/\tau=$			
<b>p7</b> $100/\tau=$			

bien. El uso de la aplicación en esta asignatura proporciona una valoración más precisa del trabajo del alumno.

**Enunciado de la práctica.**

Considere el siguiente sistema lineal:

$$Y(s) = \frac{k}{1 + \tau s} U(s)$$

dónde los parámetros  $k$  y  $\tau$  toman los siguientes valores para cada alumno en función de sus dos últimos dígitos del DNI (siendo D7 el penúltimo y D8 el último por la derecha):

- $k = (D7 + 1) \times 10$
- $\tau = D8 + 1$

Evalúe utilizando Simulink la respuesta en régimen permanente del sistema frente a una serie de señales de entrada senoidales de diferente amplitud y frecuencia, y rellene la tabla 1.

Cada respuesta correspondiente a una cuestión de trabajo en el centro de cálculo tiene asignado un nombre. En este módulo, se han agrupado las respuestas en vectores.

*Ejemplo:* La respuesta **p1** corresponde a la frecuencia de entrada, amplitud de salida, desfase temporal y ángulo de desfase de la señal de salida correspondiente a una frecuencia de entrada de  $0.01/\tau$ .

Para las respuestas **p1...p7** se pide escribir una línea con el siguiente formato:

`nombre = [frec amp des_temp des_ang];`

*Ejemplo:* Para responder que para una señal de entrada de frecuencia de 0.1 rad/s (correspondiente a  $0.3/\tau$ ) la señal de salida tiene una amplitud de 3dB, un desfase temporal de 2 segundos y un desfase angular de 0.3 radianes tenemos que escribir las siguiente líneas:

`p3 = [0.1 3 2 0.3];`

**Ejemplo de formulario relleno.**

`p1 = [0.15 10 2 0.3]; p2 = [1.5 10 0.2 0.3];`

`p3 = [4.5 5.7 0.01 0.3]; p4 = [15 -5 0.01 0.3];`

`p5 = [45 -10 0.01 0.3]; p6 = [150 -30 0.010.3];`

`p7 = [1500 -50 0.001 0.3];`

Para evaluar cada una de las prácticas se ha escrito un código en Matlab que realiza las siguientes operaciones:

- Genera los parámetros  $k$  y  $\tau$  a partir de el DNI del alumno.
- Genera un conjunto de vectores solución **s1...s7** usando la fórmula explícita de la respuesta frecuencial en régimen permanente de un sistema lineal.
- De cada uno de los elementos los vectores de respuesta se evalúa el error normalizado como

Tabla 2: Resultados de la evaluación de la práctica.

Nota	1	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1
Alumnos	9	2	2	4	3	11	43	118	60	7

$$ei(j) = \left| \frac{pi(j) - si(j)}{si(j)} \right|$$

- Fijado un porcentaje de error  $\alpha$ , la nota entre 0 y 1 de una práctica se evalúa como el número de respuestas con un margen de error menor de  $\alpha$  partido por el número total de respuestas.

### Calificaciones de los alumnos.

La práctica se corrigió con un margen de error del 10%. En la tabla 2 se muestra la distribución de notas obtenida. Se puede observar que la práctica presenta unas notas bastante bajas. Esto se debe a que, al tratar la práctica del estudio de la respuesta frecuencial de un sistema de primer orden, las medidas tomadas por los alumnos del retraso y la amplitud de señales de alta frecuencia eran poco precisas.

## 5. ENCUESTAS DE SATISFACCIÓN DE USUARIOS

Se ha empleado la plataforma con más de 4000 alumnos con resultados diversos en cuanto a calificaciones y mejoras con respecto a años anteriores. En ((Muñoz de la Peña and Gómez-Estern, 2009)) se mostró el resultado de una encuesta al alumnado para una versión inicial del servidor destinada a la asignatura Teoría de Sistemas. En la figura 9 se ilustra una encuesta masiva realizada entre los alumnos de Fundamentos de Informática.

Por último, en la tabla 3 se muestra el resultado de una encuesta realizada entre los profesores involucrados en asignaturas de la Escuela Superior de Ingenieros donde se ha empleado la herramienta. La muestra es pequeña (7 profesores), ya que se restringe al profesorado de 5 asignaturas, aunque de alumnado numeroso. Sin embargo, se puede apreciar un consenso en cuanto a la conveniencia de uso de la herramienta para profesores y alumnos, mientras se atisba cierta desconfianza a la hora de depositar completamente la responsabilidad de la evaluación en el sistema.

## 6. CONCLUSIONES

Se ha presentado una nueva aplicación basada en web para la recogida y evaluación automática de trabajos prácticos dentro de los estudios de Ingeniería. Esta herramienta proporciona un marco general de trabajo en el que se pueden definir fácilmente muy diversas clases de ejercicios. La aplicación ha sido empleada en varias asignaturas de programación en la Escuela Superior de Ingenieros de la Universidad de Sevilla. En los casos de estudio presentados, los alumnos resolvieron problemas complejos de programación, así como de análisis de sistemas dinámicos, enviando sus trabajos a través de Internet para ser posteriormente evaluados automáticamente. La plataforma permite al profesor implementar técnicas innovadoras de docencia que fomentan el autoaprendizaje en cursos con un gran número matriculados, y ya ha sido probada con más de 4000 alumnos (las tecnologías involucradas permiten una escalabilidad virtualmente ilimitada). Actualmente se están explorando otras áreas de aplicación de la herramienta: cálculo numérico, química, laboratorios virtuales, matemáticas, etc.

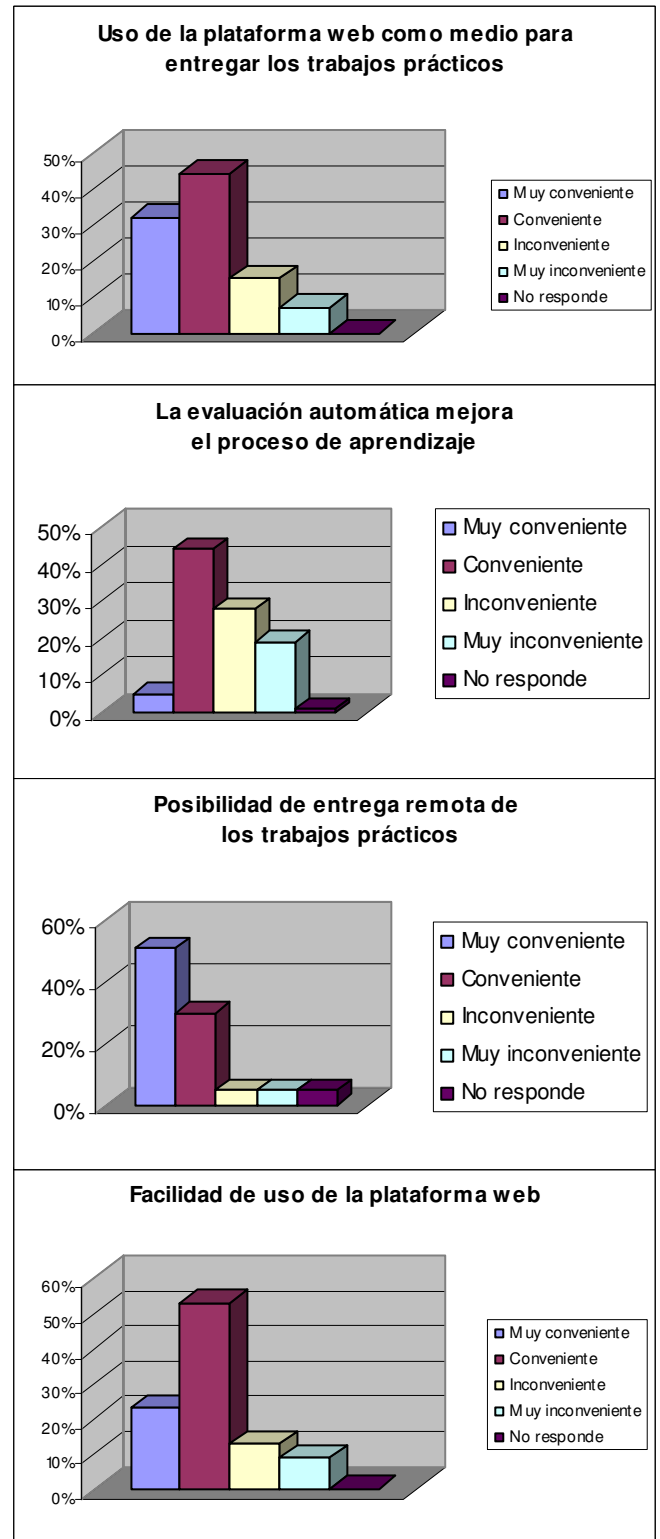


Figura 9. Resultado de la encuesta realizada en Fundamentos de Informática.

## AGRADECIMIENTOS

El presente trabajo se ha desarrollado gracias a las ayudas para la realización de Proyectos de Innovación y Mejora Docente, incluidas en el I Plan Propio de Docencia de la Universidad



Tabla 3. Resultados de la encuesta al profesorado. (Votos totales. 1=Desfavorable, 5=Muy favorable)

Cuestión	1	2	3	4	5	NS/NC
Utilidad de la herramienta para tu trabajo como profesor (economía de esfuerzo, organización, rapidez, información disponible)	0	0	0	1	6	0
Utilidad de la herramienta para el aprendizaje del alumno	0	0	0	4	3	0
¿Se logra mayor implicación del alumno al emplear la herramienta y su metodología de trabajo?	0	0	1	2	4	0
Calidad técnica de la aplicación (robustez, opciones disponibles)	0	1	1	3	2	0
Calidad del interfaz gráfico (uso amigable, estética, orden en la presentación de datos)	0	1	0	4	2	0
Confianza que le inspira la nota calculada automáticamente	0	1	1	4	1	0
Facilidad de diseño de nuevas prácticas (creación de evaluadores, selección de opciones y filtros)	0	1	1	3	1	1
Facilidad de gestionar el desarrollo de la práctica en el aula del Centro de Cálculo cuando se emplea la herramienta	0	0	1	2	2	2
Facilidad para ejecutar la evaluación automática para calificar un grupo (gestión de bucles infinitos, errores de problemas sintaxis de las entregas)	0	1	1	3	2	0
¿Utilizarías la herramienta, con todos sus filtros y supervisión del profesor, como único medio para calificar su asignatura?	2	1	1	1	2	0
Satisfacción general del alumno con la herramienta (tanto por su comodidad de uso como por la metodología en general)	0	0	2	4	0	1

de Sevilla. Los autores agradecen igualmente el esfuerzo y las valiosas sugerencias de los revisores anónimos de este artículo.

### REFERENCIAS

- Casini, M., Prattichizzo D. and Vicino A. (2003). The automatic control telelab: A user-friendly interface for distance learning. *IEEE Transactions on Education* **46**, 252–257.
- Casini, M., Prattichizzo D. and Vicino A. (2004). The automatic control telelab - a web-based technology for distance learning. *IEEE Control Systems Magazine* **24**, 36–44.
- Cedazo, R., D. Lopez, F.M. Sanchez and J.M. Sebastian (2007). Ciclope: Foss for developing and managing educational web laboratories. *IEEE Transactions on Education* **50**(4), 352 – 359.
- Domínguez, M., P. Reguera and J.J. Fuertes (2005). Laboratorio remoto para la enseñanza de la automática en la universidad de león (españa). *Revista Iberoamericana de Automática e Informática Industrial* **2**(2), 36–45.
- Dormido, S. and F. Torres (2005). Aplicación de las tic's a la educación en automática. *Revista Iberoamericana de Automática e Informática Industrial* **2**(2), 3–7.
- García, M. I., S. Rodríguez, A. Perez and A. Garcia (2009). p88110: A graphical simulator for computer architecture and organization courses. *IEEE Transactions on Education* **52**(2), 248 – 256.
- Guzmán, J.L., F. Rodríguez, M. Berenguel and S. Dormido (2005). Laboratorio virtual para la enseñanza de control climático de invernaderos. *Revista Iberoamericana de Automática e Informática Industrial* **2**(2), 82–92.
- Heck, B.S., S.E. Poindexter and García R. (2000). Integrating the web into traditional teaching methods.. *Proceedings of the American Control Conference* **5**, 3444–3448.
- Jiménez, L.M., R. Puerto, O. Reinoso, C. Fernández and R. Neco (2005). Recolab: Laboratorio remoto de control utilizando matlab y simulink. *Revista Iberoamericana de Automática e Informática Industrial* **2**(2), 64–72.
- Kapur, S. and G. Stillman (1997). Teaching and learning using the world wide web: A case study. *Innovations in Education and Training International* **34**, 316–322.
- Kerer, C., G. Reif, T. Gschwind, E. Kirda, R. Kurmanowitsch and M. Paralic (2005). Shareme: Running a distributed systems lab for 600 students with three faculty members. *IEEE Transactions on Education* **48**(3), 430 – 437.
- Muñoz de la Peña, D. and F. Gómez-Estern (2009). A new web-based tool for education and automatic evaluation in control systems engineering. *Advances in Control Education ACE'2009. Kumamoto, Japan*.
- Poindexter, S. E. and B. S. Heck (1999). Using the web in your courses: What can you do? What should you do?. *IEEE Control Systems Magazine* **19**, 83–92.
- Rodríguez, S., J. Zamorano, F. Rosales, A.G. Dopico and J.L. Pedraza (2007a). A framework for lab work management in mass courses. application to low level input/output without hardware. *Computers and Education* **48**(2), 153 – 170.
- Rodríguez, S., J.L. Pedraza, A. Garcia, F. Rosales and R. Mendez (2007b). Computer-assisted assembly language programming laboratory. *International Journal of Electrical Engineering Education* **44**(3), 216 – 229.
- Rodríguez, S., J.L. Pedraza, A.G. Dopico, F. Rosales and R. Mendez (2007c). Computer-based management environment for an assembly language programming laboratory. *Computer Applications in Engineering Education* **15**(1), 41 – 54.
- Sanchez, J., F. Morilla, S. Dormido, J. Aranda and P. Ruiperez (2006). Virtual and remote control labs using java: a qualitative approach. *IEEE Control Systems Magazine* **22**, 8–20.
- Sanchez, J., S. Dormido, R. Pastor and F. Morilla (2004). A java/matlab-based environment for remote control system laboratories: Illustrated with an inverted pendulum. *IEEE Transactions on Education* **47**, 321–329.
- Torres, F., F. A. Candelas, S. T. Puente, J. Pomares, P. Gil and F. G. Ortiz (2006). Experiences with virtual environment and remote laboratory for teaching and learning robotics at the University of Alicante. *International Journal of Engineering Education* **22**, 766–776.
- Zuluaga, C.A., C.G. Sánchez and E.A. Rodríguez (2005). Laboratorio de automática vía internet (lavi). *Revista Iberoamericana de Automática e Informática Industrial* **2**(2), 30–35.