

Automatización de prácticas en entornos masificados

Antonio

García Dopico

Dpto. de Arquitectura y Tecnología
de Sistemas Informáticos
Universidad Politécnica de Madrid
28660 Boadilla del Monte
e-mail: dopico@fi.upm.es

Santiago

Rodríguez de la Fuente

Dpto. de Arquitectura y Tecnología
de Sistemas Informáticos
Universidad Politécnica de Madrid
28660 Boadilla del Monte
e-mail: srodri@fi.upm.es

Francisco Javier

Rosales García

Dpto. de Arquitectura y Tecnología de
Sistemas Informáticos
Universidad Politécnica de Madrid
28660 Boadilla del Monte
e-mail: frosal@fi.upm.es

Resumen

La docencia en el área de Arquitectura de Computadores tiene una fuerte componente práctica, que se aborda a través de la realización de múltiples prácticas por parte de los alumnos. En la Facultad de Informática de la Universidad Politécnica de Madrid (UPM) estas prácticas se tienen que realizar para un elevado número de alumnos, que oscila de 500 a 700 por asignatura. El afán de realizar prácticas de calidad para todos estos alumnos ha obligado a desarrollar un entorno automatizado de prácticas cuyos componentes principales se presentan en este artículo.

1. Introducción

En general, la docencia se compone de una parte teórica y de otra práctica. La parte práctica es imprescindible para afianzar los conocimientos explicados en la teoría, esto se consigue enfrentado al alumno a un problema concreto que debe resolver, lo que le obliga a aplicar los conocimientos explicados a lo largo de la asignatura. Con frecuencia, a los alumnos le surgen dudas y problemas, en cuyo caso acuden a los profesores de prácticas para que les ayuden a resolverlos. Este trato directo con los profesores es positivo y hace más humana la docencia.

Desafortunadamente hay situaciones en las que no es posible el funcionamiento antes expuesto. Por ejemplo, en entornos de educación a distancia o allí donde el número de alumnos es muy elevado. En ambos casos, la posibilidad de un trato directo se ve mermada por el contexto en que se desarrollan las prácticas.

En el Departamento de Arquitectura y Tecnología de Sistemas Informáticos (DATSI) de la Facultad de Informática de la UPM, se tiene una media de 500 a 700 alumnos por asignatura troncal u obligatoria. Como se desean realizar varias prácticas a lo largo de las distintas asignaturas, la masificación representa un problema importante que hay que abordar.

Las prácticas propuestas no son de resolución trivial, dado que tratan en profundidad temas de cierta complejidad para los alumnos. Cada una es un pequeño proyecto, con una especificación, que se entrega al alumno y cuyo contenido se explica en clase. Esto, unido a que se exige la correcta realización de todas las prácticas, comprobando para cada una de ellas que se obtienen los resultados correctos, da una idea de la cantidad de trabajo que esto representa. La simple comprobación de todos los resultados sería una tarea casi imposible de abordar manualmente, ya que cada alumno dispone de varias oportunidades para su correcta realización. Esto ha obligado a automatizar una parte importante de la gestión y seguimiento de las prácticas.

La experiencia docente del departamento en la impartición de las prácticas permitió observar que los trabajos que los alumnos entregaban como prácticas completas adolecían de numerosas deficiencias:

- El alumno no había comprendido la especificación de la práctica.
- El alumno ni siquiera se había planteado el establecimiento de una batería de prueba que permitiera comprobar su trabajo de forma sistemática.
- El alumno no aprendía a depurar su código.

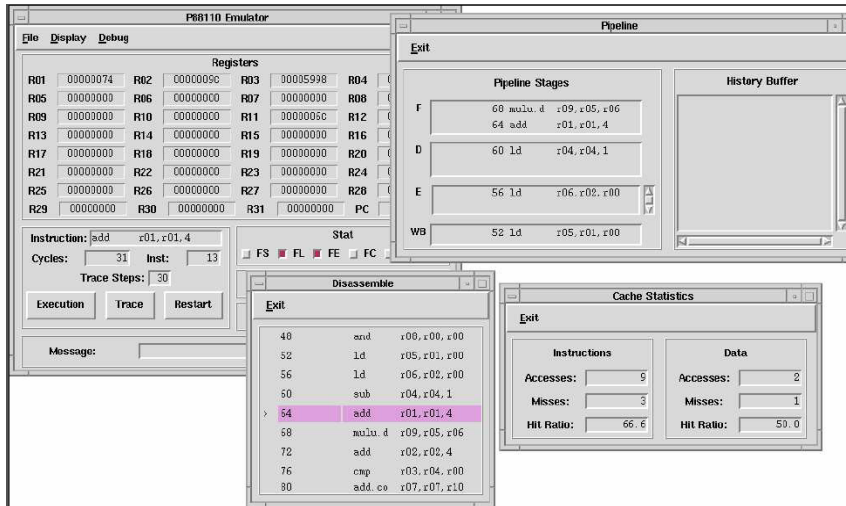


Figura 1. Interfaz gráfica del mc88110

Aunque la automatización reduce el contacto entre alumnos y profesores, en su momento se consideró que no había alternativas factibles a la misma. Las alternativas que se tenían: hacer menos prácticas, que fuesen mucho más sencillas o no comprobar su correcto funcionamiento, se consideraron poco didácticas, ya que todas ellas repercuten en un menor aprendizaje por parte de los alumnos.

Por tanto, se ha preferido mantener un alto número de prácticas, de complejidad media e incluso alta, y comprobar que todos los alumnos obtenían los resultados deseados en los tres apartados anteriores.

2. Prácticas en el DATSI

Como muestra del conjunto de prácticas de que se está hablando se enumeran a continuación algunas de ellas.

En el grupo de arquitectura se proponen las siguientes prácticas para su realización en grupos de dos alumnos:

- En Estructura de Computadores una práctica de microprogramación, en el tema de la

unidad de control. Emplea un simulador basado en el i8080, el p8080e.

- En Laboratorio de Estructura de Computadores una práctica de ensamblador, dentro del tema de programación en ensamblador. Se emplea un simulador del mc88110 [4], cuya interfaz puede verse en la figura 1.
- En Arquitectura de Computadores hay dos prácticas, una de entrada/salida, en su tema correspondiente, y otra de memorias caché, en el tema de jerarquía de memoria. En la primera práctica se emplea un simulador del mc68000, el BSVC [3], cuya interfaz puede verse en la figura 2. En la segunda se emplea el simulador del mc88110 ya mencionado [5].

En el grupo de sistemas operativos las prácticas son individuales y se proponen las siguientes:

- En Fundamentos de la Programación de los Sistemas Operativos varias prácticas de programación en C.
- En Sistemas Operativos una práctica de *shell scripts* y otra de implementación de un pequeño intérprete de mandatos.

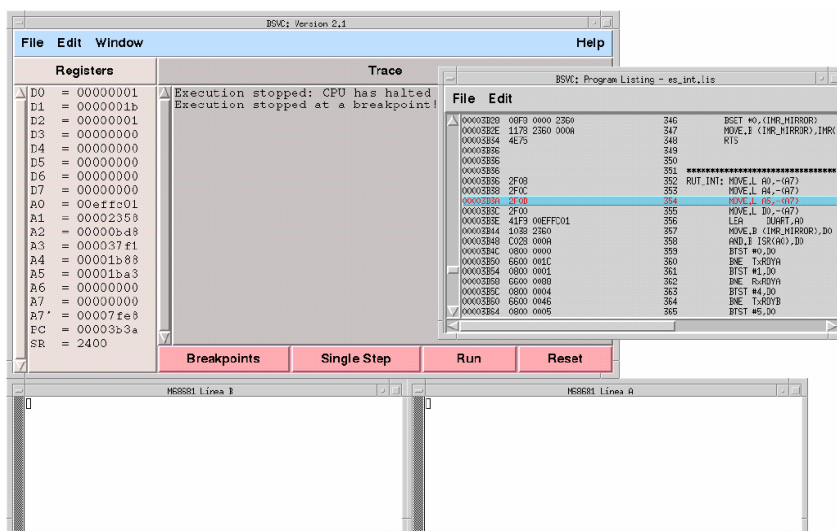


Figura 2. Interfaz gráfica del BSVC

- En Diseño de los Sistemas Operativos hay una práctica de análisis del comportamiento del gestor de memoria del S.O. y otra en la que se incorpora funcionalidad a un pequeño núcleo de sistema operativo.

3. Entorno de prácticas

Para posibilitar la realización de estas prácticas en el contexto descrito ha sido preciso el desarrollo de un entorno de prácticas, compuesto por un conjunto de herramientas y métodos para la automatización de la gestión de las mismas. Básicamente, este entorno de prácticas se basa en:

- Enunciados y documentación en la Web, así como en formato papel. Esto le permite a los alumnos acceder a la documentación de forma remota.
- Tutorías electrónicas vía e-mail, aunque, por supuesto, siguen atendándose tutorías presenciales. Así, aunque el alumno esté trabajando a distancia, se le resuelven las dudas que le surjan, ahorrándole tiempo. Esta vía de comunicación tiene la ventaja de que

permite crear un archivo con las preguntas más frecuentes (FAQ). Este archivo ahorra muchas preguntas similares.

- Entregador de prácticas remoto. Permite a los alumnos enviar su práctica desde cualquier sitio, sin necesidad de desplazarse, así como consultar también de forma remota los resultados obtenidos tras la corrección.
- Corrector automático. Ejecuta una batería de pruebas sobre cada práctica de cada alumno o grupo. El objetivo del corrector es producir un resultado que proporcione al alumno información suficiente para determinar el origen del error detectado y sugerirle alternativas para su solución. De este modo los alumnos pueden ir corrigiendo su práctica sin acudir sistemáticamente al profesor.
- Detector de copia de prácticas. Asiste al profesor en una labor que manualmente sería imposible: detectar sobre el total de prácticas entregadas qué parejas contienen código plagiado una de la otra.

A continuación se describen en mayor profundidad el entregador, el corrector y el

detector de copias, que son la parte central de este entorno de prácticas.

4. Sistema de entrega remota

Permite que el alumno se conecte desde cualquier sitio a las máquinas del departamento. Para dicha conexión debe identificarse mediante un *login* y un *password*.

El entregador dispone de una lista con todos los alumnos que pueden realizar una práctica concreta. Para evitar accesos indeseados, sólo dichos alumnos podrán darse de alta en el sistema de entregas.

Una vez conectado, se puede enviar la práctica, consultar los resultados obtenidos en correcciones anteriores, ver las noticias más recientes relativas a esa práctica o anular alguna entrega realizada incorrectamente. A continuación se muestra el menú del entregador:

```
Practica de Lab. Estr. de Computadores
Septiembre 2002
Servidor de Practicas. Version 1.9.5
Introduzca su login (matricula): k0267
Password:
```

```
Sesion iniciada correctamente
```

```
Practica de Lab. Estr. de
Computadores (88110).
Septiembre 2002
```

```
OPCIONES:
```

1. Mandar Ficheros.
2. Consultar Resultados.
3. Cancelar Entregas.
4. Bloquear la Entrega.
5. Ayuda !!!!
6. Noticias. (11/03/02 18:17)
- q Abandonar

```
>>
```

Figura 3. Menú del entregador

De cara al profesor encargado de la administración de la práctica, se guarda un histórico con todas las conexiones realizadas, detallando desde dónde, cuándo y quién la hizo, a modo de auditoría. También se almacena todo lo que los alumnos hacen una vez conectados. Dicho histórico se muestra en la figura 4.

Además, por cada alumno, se guardan todas sus entregas, incluso las que luego anula. De cada entrega realizada correctamente se guarda el resultado obtenido, en qué falla y por qué.

El hecho de guardar tanta información sobre lo que hace cada alumno se ha mostrado muy útil, ya que al ser un número tan alto de alumnos siempre hay algunos que se quejan por causas dispares. Por ejemplo, alumnos que afirman que le no se le ha corregido la práctica, cuando en realidad anuló su entrega, o que dicen disponer de correcciones cuando realmente las han agotado ya. La información recogida en este archivo histórico permite determinar qué sucedió en realidad.

5. Corrector automático

Recibe una batería de pruebas que ejecuta sobre todas las prácticas y para cada una de ellas almacena en un fichero qué casos pasan correctamente y cuáles fallan. Posteriormente, el alumno puede consultar, a través del entregador, dicho fichero para recoger los resultados de la corrección.

Cada prueba consiste en ejecutar la práctica del alumno con unos datos de entrada esperando obtener unos resultados concretos. En el grupo de sistemas operativos las prácticas se ejecutan directamente sobre máquinas Unix como un proceso. Pero en el caso del grupo de arquitectura, debido a los problemas que pueden surgir si se ejecutan las prácticas directamente sobre una máquina real, se emplean simuladores.

En cualquier caso el corrector automático es suficientemente flexible para poder tratar con diversos lenguajes de programación, ya sea ejecutando la práctica directamente sobre la máquina o sobre un simulador.

Todo lo que necesita es que la ejecución sea en modo *batch*. Es decir, en caso de ejecutar directamente en Unix será:

```
Practica < datos > resultados
```

```
En caso de utilizar un simulador será:
```

```
simulador < practica+datos > result
```

El alumno tiene un número limitado de correcciones a su disposición, que establece el profesor de esa práctica. Esto es así para evitar que el alumno use el corrector como un depurador, ya que se deben alcanzar los siguientes objetivos:

- Comprensión de la especificación del trabajo que debe realizar el alumno.
- Desarrollo de aplicaciones en grupo.
- Diseñar baterías de pruebas para probar su propio trabajo.

```

04 Nov 2002 12:33 f980769 ALTA(ERROR) Cancelada [batman.fi.upm.es]
04 Nov 2002 12:33 f980769 ALTA(OK) [batman.fi.upm.es]
04 Nov 2002 12:33 f980769 LOGIN(OK) [batman.fi.upm.es]
04 Nov 2002 12:34 f980769 NEWS(OK)
04 Nov 2002 12:34 f980769 LOGOUT(OK)
05 Nov 2002 09:32 a920157 LOGIN(ERROR) Password erróneo [batman.fi.upm.es]
05 Nov 2002 09:32 a920157 LOGIN(OK) [batman.fi.upm.es]
05 Nov 2002 09:32 a920157 ENTREGA(ERROR). Fichero practica.s Error en formato
05 Nov 2002 09:32 a920157 CANCELACION(OK) Error en entrega
05 Nov 2002 14:13 960016 LOGOUT(OK)
05 Nov 2002 14:14 960016 LOGIN(OK) [batman.fi.upm.es]
05 Nov 2002 14:14 960016 ENTREGA(OK)
05 Nov 2002 14:14 960016 LOGOUT(OK)

```

Figura 4. Fichero histórico

- Aprender los métodos para depurar la práctica.

Cada vez que un alumno quiera corregir su práctica, para conocer qué fallos tiene aún, bastará con que envíe los ficheros de la misma mediante el entregador. Quedará registrado que dicho alumno ha hecho una entrega y cuando de nuevo se ejecute el corrector se corregirá su práctica.

Si el alumno detecta algún fallo antes de la corrección siempre puede anular la entrega, para evitar que le cuente dicha corrección.

El corrector ejecuta un número fijo de veces al día, a unas horas concretas, que establece el profesor y son conocidas por los alumnos.

El profesor puede lanzar cuando quiera alguna corrección extra. Incluso puede decidir si quiere que cuente o no sobre el número total de correcciones que tiene cada alumno.

Cada práctica tiene una batería de casos de prueba. Cada caso de prueba consta de cuatro ficheros:

- Prueba.cf: Configuración del caso de prueba. Indica en qué fijarse tras la ejecución, dónde estarán los resultados y en cuáles se debe centrar la atención.
- Prueba.ens: Fichero fuente, que contiene los datos con los que se invocará la práctica del alumno.
- Prueba.ofl: Resultado correcto, lo que debería dar la práctica del alumno tras la ejecución.
- Prueba.res: Mensaje de error que recibe el alumno. Le indica qué ha fallado y le proporciona información para corregir su error.

Una vez que el alumno ve sus fallos, intenta corregirlos con la información que le proporciona el corrector. Si no es capaz puede preguntar al

profesor, ya sea personalmente o por correo electrónico. Muchas veces, cuando sus problemas se deben a que no entienden conceptos importantes es preferible el trato directo y acuden a las tutorías.

El conjunto de prácticas se pueden englobar en dos grupos:

- La mayor parte de ellas son prácticas funcionales en las que la calidad de los resultados se mide en función de si son conformes a la especificación dada sin hacer especial hincapié en las prestaciones proporcionadas por el trabajo del alumno.
- En el segundo tipo, al que pertenece la práctica de jerarquía de memoria, se exige que el alumno modifique un programa para que proporcione mejores prestaciones (menor número de accesos a memoria, menor número de fallos de cache, etc.). En este caso las herramientas (corrector de prácticas) deben ser capaces de comparar el resultado obtenido por el alumno con el peor de los resultados admisibles.

6. Detección de copias

El alto número de alumnos que realizan cada práctica tiene otra consecuencia inevitable aparte de las ya vistas. Aumenta mucho la probabilidad de que los alumnos se dejen las prácticas unos a otros, confiados en que el alto número de prácticas hará casi imposible detectar los casos de copia.

Para evitar que algunos alumnos aprueben las prácticas gracias al esfuerzo de otros, se ha incorporado a este entorno un detector de copias.

Esta herramienta hace una búsqueda exhaustiva, es decir, compara cada práctica con

todas las demás. Para un alto número de alumnos pueden generarse miles de comparaciones, pero los recursos actuales permiten realizar toda esta tarea en pocas horas.

El detector de copias sirve para diferentes lenguajes de programación, incluso para ensamblador o prácticas de microprogramación. Para ello, en una fase inicial procesa cada fichero de entrada identificando ciertos patrones predefinidos (por ejemplo, palabras reservadas del lenguaje) componiendo con ellos una firma o resumen de cada práctica. De esta forma, se obvian cambios elementales, como renombrado de variables o funciones, y se centra el estudio principalmente en la estructura lógica del programa, que en definitiva es aquello que quien copia no sabe cambiar.

A continuación se compara cada una de las firmas con todas las demás, evaluándose cuatro criterios distintos de similitud por cada pareja. Estos criterios cuantifican secuencias de patrones repetidos independientemente de su posición relativa. Esto permite la detección de casos en los que sólo se copia una rutina o un fragmento de la

misma, incluso si se ha movido el código respecto del original.

El resultado producido es un archivo tal que, ordenado según cada uno de los criterios de similitud, permite determinar qué parejas de prácticas son más parecidas total o parcialmente.

Debido a la gravedad del tema, esta herramienta debe considerarse sólo un asistente a la detección de copias. La decisión final la han de tomar los profesores responsables, inspeccionado detenidamente cada posible caso de copia.

7. Evaluación remota con Aulaweb

Como extensión al entorno de prácticas visto, y siguiendo una filosofía similar, se ha empleado en la asignatura de Sistemas Operativos una herramienta de evaluación remota [6] a través de Internet denominada Aulaweb, desarrollada en la Escuela de Ingenieros Industriales de la UPM [1].

Esta herramienta permite programar exámenes periódicos que los alumnos deben resolver, conectándose a la misma vía Internet. A pesar de

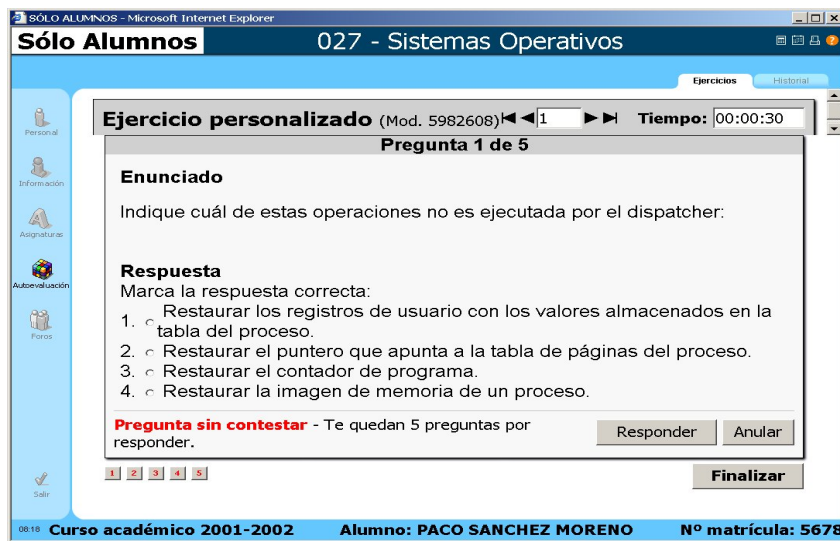


Figura 5. Examen de Aulaweb

poder tener los apuntes y libros delante, este método les obliga a llevar la materia al día y a estudiar de forma continuada.

Dado que a la hora de hacer el examen no se tiene ningún control presencial, no se puede asegurar la identidad del alumno ni evitar que compañeros suyos le ayuden. Por ello, estos exámenes tienen poco peso en la nota final, aunque deben realizarse obligatoriamente para poder acceder al examen final de la asignatura.

Cuando un alumno se conecta a través de un navegador a esta herramienta, se construye automáticamente un nuevo examen para dicho alumno. Se obtienen de forma aleatoria "n" preguntas de una gran base de datos de preguntas de la asignatura. El número y tipo de las preguntas es configurable. En la figura 5 se muestra una pregunta tipo test de esta herramienta.

De esta forma cada alumno se enfrenta a preguntas diferentes, por lo que no le sirve de nada mirar como un compañero resuelve otro examen. Se permite que un alumno repita su examen si no está satisfecho con la nota obtenida, ya que el objetivo no es tanto evaluarlo (la nota cuenta muy poco) como obligarle a estudiar. Si repite el examen estará profundizando más en la materia.

8. Conclusiones

Se ha presentado el entorno de prácticas que se emplea en el Departamento de Arquitectura y Tecnología de Sistemas Informáticos de la Facultad de Informática de la Universidad Politécnica de Madrid. Así mismo se han presentado sus componentes principales: un entregador remoto, un corrector automático y un detector de copias.

Este entorno de prácticas se ha mostrado apto para entornos masificados, donde un pequeño grupo de profesores tiene que llevar varias prácticas, cada una de ellas con un elevado número de alumnos.

Gracias a este entorno se ha podido mantener un alto número de prácticas, de complejidad media e incluso alta, y comprobar que todos los alumnos obtenían los resultados deseados. Esto obligó a automatizar parte de la gestión y mantenimiento, para agilizar el proceso y disminuir la carga de trabajo de los profesores.

El entregador remoto se ha mostrado muy útil tanto para los alumnos, que pueden realizar la práctica en cualquier lugar, como para los profesores, ya que permite centralizar y auditar toda la actividad que los alumnos realizan con sus prácticas.

El corrector automático permite diseñar prácticas complejas que pueden ser verificadas en todos sus detalles. Los alumnos no sólo deben realizar la práctica sino que esta debe funcionar correctamente según las especificaciones entregadas. El corrector automatiza esta verificación, generando información sobre los fallos detectados. Esto permite a los alumnos corregir sus errores de modo autónomo, sin precisar una ayuda constante del profesor.

La experiencia nos ha demostrado que siempre hay un porcentaje de alumnos que copia las prácticas. El detector de copias se ha mostrado tan efectivo en la detección de estos casos que incluso ha disuadido a muchos alumnos de su intención. Esto lo demuestra el hecho de que el porcentaje de alumnos que copian se ha reducido considerablemente en los últimos años.

En definitiva, la experiencia nos ha demostrado la utilidad de este entorno, ya que, a pesar de la masificación en que se desarrollan las prácticas, éstas siguen cumpliendo correctamente su cometido original: afianzar los conocimientos teóricos básicos que deben tener los alumnos. Esto lo demuestra el hecho de que un elevado porcentaje de alumnos pasan todas las pruebas satisfactoriamente, asegurándonos que han aprendido los conceptos en los que se apoya la práctica. Para confirmar esta conclusión se muestran unos cuantos datos de dos prácticas concretas:

- Práctica de ensamblador de Laboratorio de Estructura de Computadores en la convocatoria de junio de 2002. El corrector empleaba una batería de 54 pruebas, que debían pasarse en su totalidad.
 - 90% de aprobados entre los grupos que empezaron en fecha (64 grupos aprobados de 71)
 - 50% de aprobados entre los grupos que empezaron tarde (43 aprobados de 86)
 - En total 68% de aprobados (107 de 157).
- Práctica de entrada/salida de Arquitectura de Computadores en la convocatoria de febrero

de 2003. El corrector empleaba una batería de 45 pruebas, que también debían pasarse en su totalidad.

- 79% de aprobados entre los grupos que empezaron en fecha (112 grupos aprobados de 142)
- 49% de aprobados entre los grupos que empezaron tarde (36 aprobados de 73)
- En total 69% de aprobados (148 de 215).

En definitiva, este entorno se ha mostrado muy adecuado para tratar con el problema de la masificación, consiguiendo que los alumnos sean mucho más autónomos de lo que antes eran, a pesar de la complejidad y variedad de las prácticas propuestas.

El hecho de que varias prácticas, muy distintas entre sí, empleen este entorno, muestra su flexibilidad. De la misma forma que se emplea para prácticas con shell scripts, C, ensamblador, entrada/salida o microprogramación, se podría emplear para prácticas con Ada, Java o cualquier otro lenguaje de programación.

Por último cabe mencionar que la mayoría de las herramientas aquí mencionadas son accesibles vía Web (em88110, BSVC, Aulaweb) aunque algunas no lo son ya que no pueden estar accesibles a los alumnos, como puede ser el detector de copias (podrían intentar camuflar sus copias) o el corrector automático, ya no desarrollarían su propia batería de pruebas y usarían la del corrector. Para mayor información de dichas herramientas contacte vía e-mail con los autores del artículo.

Referencias

- [1] García-Beltrán, A.; Martínez, R. *The role of self-assessment in Aulaweb e-learning system*. European Distance Education Network, Annual Conference, Granada, Spain, junio 2002.
- [2] *MC88110: Second Generation RISC Microprocessor. User's Manual*. Motorola Inc. 1991.
- [3] Mott, B.W. *BSVC A microprocessor simulation framework*. <http://www.redlinelabs.com/bsvc>
- [4] Pérez, J.M.; Rodríguez, S.; Méndez, R.; García M.I. *The em88110: Emulating a superscalar processor*. ACM SIGCSE Bulletin, 29(4): 45-50, Septiembre 1997
- [5] Rodríguez de la Fuente, S.; García Clemente, M.I.; Méndez Cavanillas, R. *Teaching Computer Architecture with a New Superscalar Processor Emulator*. Proceedings of the 4th annual SIGCSE/SIGCUE Conference on Innovation and Technology in Computer Science Education, ITiCSE'99, 99-102, Cracow, Poland.
- [6] Sánchez Moreno, F.M.; García Dopico, A. *Experiencia docente con Aulaweb en la asignatura de Sistemas Operativos*. XIII Jornadas de Paralelismo, 87-90 Lleida, 2002.