

PDIWEB: SIMULADOR WEB PARA LA DOCENCIA DE LA PLANIFICACIÓN DINÁMICA DE INSTRUCCIONES

Miguel A. Vega, J. Carlos Burgueño, Juan M. Sánchez, Juan A. Gómez

*Departamento de Informática. Universidad de Extremadura
Escuela Politécnica. Campus Universitario, s/n. 10071-Cáceres. Spain
e-mail: mavega@unex.es Fax: +34-927-257-202*

RESUMEN

Internet es ya, en la actualidad, un recurso docente muy importante, puesto que permite superar las limitaciones de lugar y tiempo, abaratando costes. Además, no debe olvidarse el efecto que la interactividad tiene en el proceso de aprendizaje. Por este motivo, creemos que es importante dedicar esfuerzos en la elaboración de propuestas y prototipos de enseñanza para impartir docencia a través de Internet. En esta ponencia se presenta un sistema multimedia, basado en Internet, que se ha desarrollado y se está aplicando en la docencia de la planificación dinámica de instrucciones (PDI). La PDI es un aspecto fundamental dentro de la segmentación de instrucciones (procesadores segmentados), siendo estudiada en toda asignatura sobre Arquitectura de Computadores. La plataforma aquí presentada recibe el nombre de PDIWeb, y se ha desarrollado gracias a una ayuda concedida para Proyectos de Innovación Docente en la Universidad de Extremadura. La ponencia presenta una descripción general de este simulador web, así como de las herramientas y métodos utilizados para su construcción. En conclusión, se trata de una innovación docente que permite mejorar la enseñanza en Arquitectura de Computadores.

1. INTRODUCCIÓN

Las tecnologías de la información y la comunicación se han convertido en una herramienta fundamental en educación, debido a sus grandes ventajas: cualquier lugar y momento, ahorro de costes, interactividad, etc. Por ello, creemos muy importante dedicar esfuerzos en la elaboración de propuestas y prototipos de enseñanza para impartir docencia a través de Internet. En esta línea, desde 1998 nuestro grupo de investigación ha trabajado en diversos proyectos, como EDONET (Entorno para la DOcencia sobre interNET) [7], TEDA (Tele-Enseñanza para Discapacitados Auditivos) [9] o SD2I (Sistema para la Docencia de Sistemas Digitales a través de Internet) [10].

En la actualidad, nos encontramos inmersos en el proyecto PDIWeb (Simulador Web para la Docencia de la Planificación Dinámica de Instrucciones). Este proyecto surge de la aplicación de nuestra investigación a la docencia que impartimos. El objetivo global es el desarrollo de un sistema para la enseñanza a través de Internet de parte de la materia de la asignatura Arquitectura e Ingeniería de Computadores (AIC). AIC es una asignatura troncal anual de 9 créditos, que se imparte en cuarto curso de la titulación de Ingeniero Informático, dentro de la Universidad de Extremadura (UEX). En particular, este sistema se centra en el temario impartido durante el primer cuatrimestre y dedicado a la planificación dinámica de instrucciones, aspecto fundamental dentro de la segmentación de instrucciones.

El proyecto PDIWeb se inició tras la obtención de una ayuda para Proyectos de Innovación Docente, concedida por el Vicerrectorado de Innovación Educativa y Nuevas Tecnologías, y el Instituto de Ciencias de la Educación (ambos de la Universidad de Extremadura).

El resto de la ponencia se organiza como sigue: en la próxima sección describimos brevemente el contexto en el que se utiliza este simulador. Posteriormente, en la sección 3 se

da una visión general de la plataforma que se ha construido para la enseñanza de la planificación dinámica de instrucciones, indicando sus características fundamentales. La sección 4 detalla las herramientas y métodos que se han seguido para la implementación de esta plataforma. Para finalmente, en la sección 5, presentar las conclusiones de este trabajo.

2. CONTEXTO

La asignatura AIC, impartida en 4º curso de la Ingeniería Informática en la UEX, tiene como objetivo general formar al alumno en los fundamentos arquitectónicos de los computadores actuales. Este objetivo abarca el estudio de muy diversos campos: análisis del rendimiento de arquitecturas, segmentación, multiprocesadores, multicomputadores, etc.

Para obtener información más detallada sobre el programa de la asignatura el lector puede consultar la referencia [11].

3. DESCRIPCIÓN DE PDIWEB

PDIWeb [12] es un simulador web, basado en la arquitectura de una máquina MIPS segmentada con juego de instrucciones de 64 bits, que simula la planificación dinámica de instrucciones utilizando dos técnicas distintas: una centralizada mediante el *método del marcador*, y otra distribuida según el *algoritmo de Tomasulo* [3].

El simulador no requiere ningún tipo de instalación en la máquina por parte del estudiante, ya que se ejecuta a través de Internet en un servidor remoto. Por tanto, para que un alumno pueda ejecutar el simulador, sólo necesita una máquina con acceso a Internet y un navegador que pueda visualizar páginas web diseñadas con marcos (*frames*) HTML [4] y elementos multimedia de tipo *flash*. En concreto, se ha probado con éxito el simulador en los navegadores *Internet Explorer* y *Netscape*.

Los siguientes apartados describen brevemente la interfaz de PDIWeb.

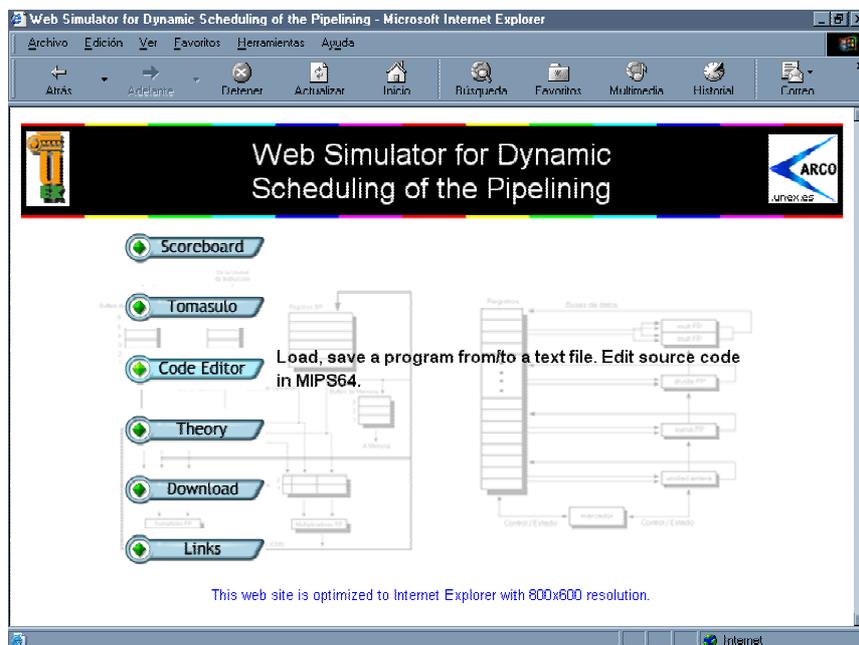


Figura 1. Página principal de PDIWeb.

3.1. Página principal

El alumno parte de la página principal del simulador (ver figura 1), que contiene el menú principal de la plataforma.

En la parte superior de la página, puede verse el título del sitio web y los logotipos de la Universidad de Extremadura y del grupo de investigación ARCO. El resto de la página está ocupado por una serie de botones *flash*. Cada botón contiene el nombre de la función que realiza. Además, puede apreciarse un cambio en el aspecto gráfico de cada botón (animación del mismo) al pasar el ratón sobre él. Este cambio en el aspecto gráfico va acompañado de la aparición de un texto aclarativo con la funcionalidad del botón en cuestión.

3.2. Método del Marcador

Si el alumno selecciona el botón *Scoreboard*, en la página principal del simulador, accederá al método del Marcador. Las figuras 2 y 3 muestran el aspecto de esta página web.

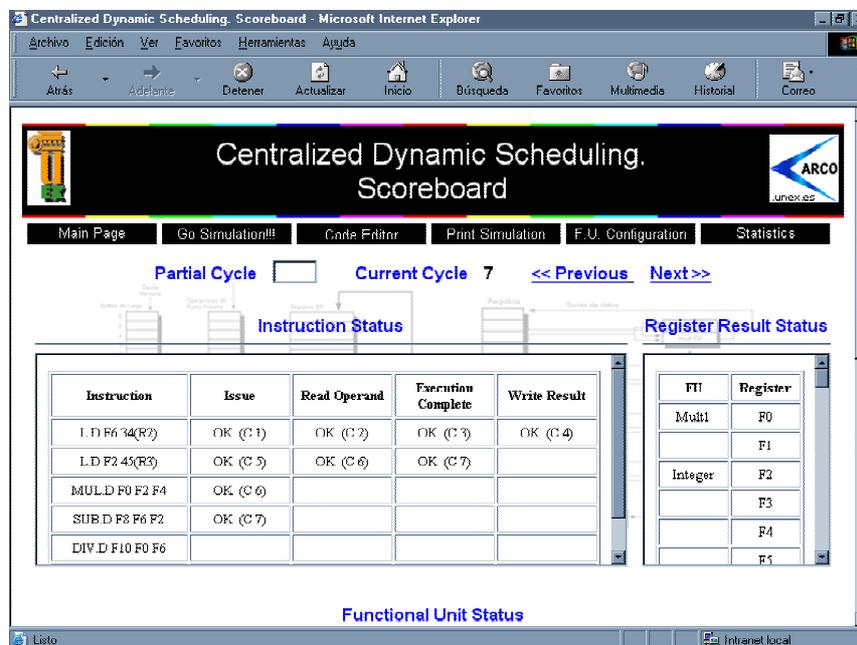


Figura 2. Parte superior de la página web del simulador para el método del Marcador.

En esta página, aparece en primer lugar un menú, seguido de las opciones que controlan la evolución de la simulación: *Partial Cycle*, *Current Cycle*, *Previous* y *Next*. El resto de la página está ocupado por las tablas de estado que utiliza el método del Marcador. La *Tabla de Estado de las Instrucciones (Instruction Status)* muestra el estado de las instrucciones cargadas en el simulador, es decir, en qué etapa se encuentran. La *Tabla de Estado de los Resultados en Registros (Register Result Status)* presenta el estado de los registros destino. En primer lugar aparecen los registros coma flotante, y posteriormente, los registros enteros. La *Tabla de Estado de las Unidades Funcionales (Functional Unit Status)* proporciona información sobre el estado de las unidades funcionales de la máquina: si están ocupadas o no por una instrucción, por qué instrucción, con qué operandos, ... Finalmente, la *Tabla de Estado de las Etapas (Stage Status)* muestra la evolución de todas las instrucciones por las distintas etapas de su ejecución, ciclo a ciclo, indicando las detenciones que se producen, etc.

Para controlar la evolución de la simulación se utilizan las opciones: *Partial Cycle*, *Current Cycle*, *Previous* y *Next*. *Partial Cycle* permite al usuario indicarle al simulador hasta que ciclo

quiere que se simule la ejecución del código, admitiendo números entre 0 y 999. El valor 0 actúa como *reset* del simulador: si se introduce este valor, la información de las tablas será borrada, a excepción del código cargado, por tanto, reiniciándose la simulación. *Current Cycle* indica el número de ciclo por donde se encuentra la simulación. Los enlaces *Previous* y *Next* permiten avanzar o retroceder un ciclo en la simulación.

Por su parte, en la zona superior de la página, se encuentra un menú con las operaciones posibles. *Main Page* es un enlace a la página principal de la plataforma (figura 1). *Go Simulation!!!* es la opción que lanza la simulación del programa que está cargado en el simulador. La simulación durará como máximo el número de ciclos de reloj indicado en *Partial Cycle*. Si este campo está vacío, la simulación se realizará hasta la terminación de todas las instrucciones. *Code Editor* es un enlace hacia la página del editor de código, que se explicará en el apartado 3.4.

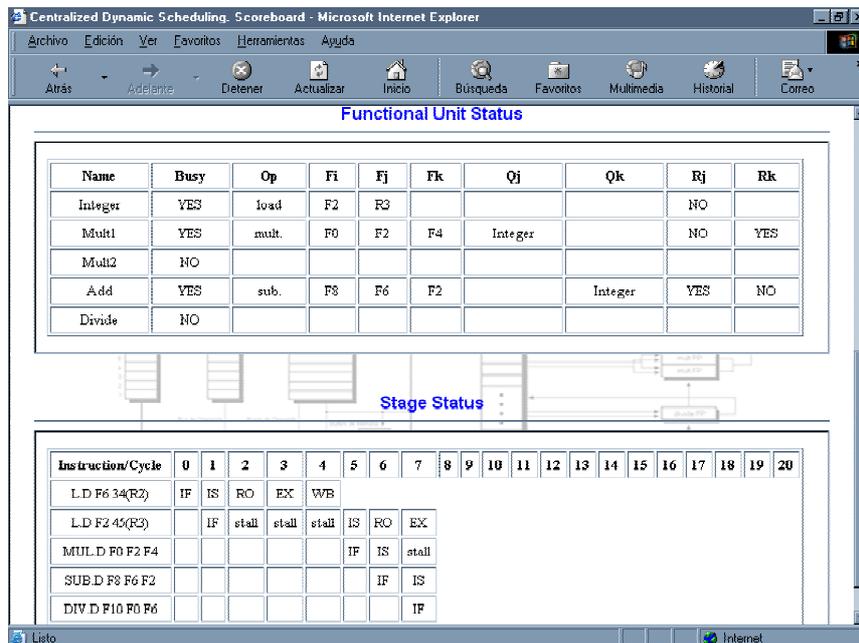


Figura 3. Parte inferior de la página web del simulador para el método del Marcador.

Print Simulation genera una página *HTML* que contiene los resultados de la simulación hasta el ciclo actual, en un formato preparado para ser impreso. La página muestra el contenido de las tablas del Marcador, el ciclo actual de la simulación, los riesgos producidos y la configuración de las unidades funcionales. Al final de la página aparecen dos botones: el botón *Print* imprime esta página utilizando la salida estándar de la impresora, y el botón *Cancel* cancela la operación y retorna a la página del simulador. La figura 4 presenta un ejemplo.

F.U. Configuration (figura 2) accede a la página web en la que se pueden configurar las unidades funcionales de la máquina. Esta página permite al usuario cambiar la duración de la etapa de ejecución (*EX*) de las distintas unidades funcionales. La página ofrece un aspecto distinto dependiendo de si se está utilizando el método del Marcador o el algoritmo de Tomasulo. La figura 5 muestra un ejemplo para el caso del Marcador.

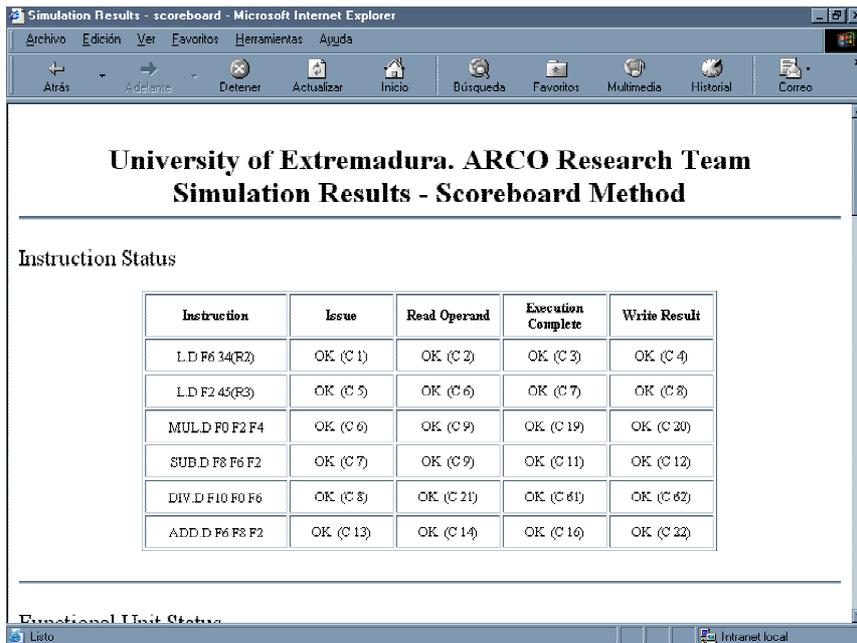


Figura 4. Página para imprimir los resultados de la simulación.

El menú de esta página (figura 5) permite volver a la página principal (figura 1) del simulador (*Main Page*), volver a la simulación con el Marcador o Tomasulo utilizando los nuevos valores configurados (*Load Scoreboard* o *Load Tomasulo*), o establecer los valores por defecto para todas las unidades funcionales (*Default Values*).

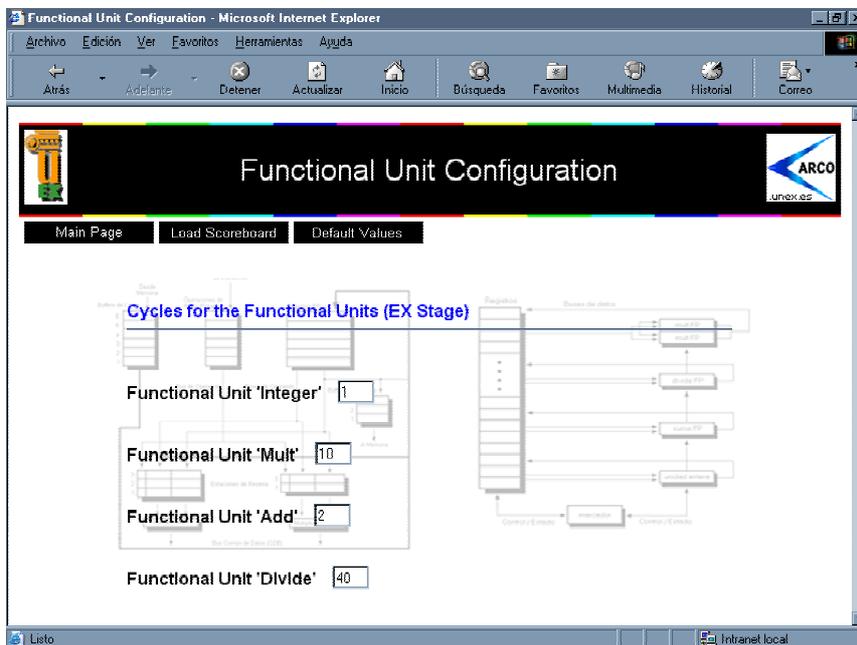


Figura 5. Página de configuración de las unidades funcionales.

Volviendo al método del Marcador, nos queda por explicar la opción de menú *Statistics* (figura 2). Esta opción es un enlace hacia la página que muestra las estadísticas de los riesgos producidos durante la simulación. Los datos se representan en formato gráfico, porcentual y numérico. La figura 6 presenta un ejemplo.

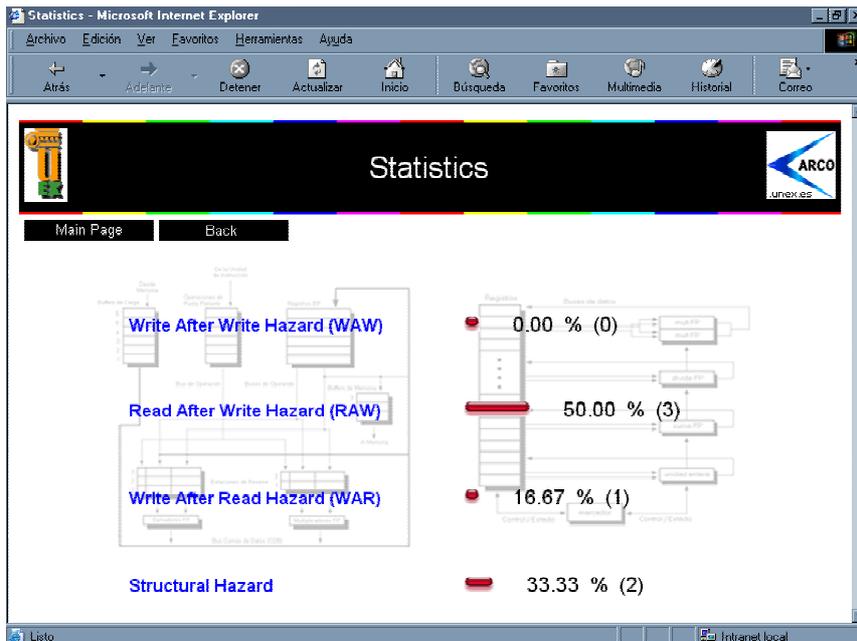


Figura 6. Página web con las estadísticas sobre riesgos tras una simulación.

El menú dinámico de esta página permite al usuario volver a la página principal de la plataforma (*Main Page*) o retornar de vuelta a la simulación (*Back*) mediante el Marcador o Tomasulo, según el caso.

3.3. Algoritmo de Tomasulo

Si el estudiante selecciona el botón *Tomasulo*, en la página principal (figura 1) del simulador, accederá a la simulación mediante el algoritmo de Tomasulo. Las figuras 7 y 8 muestran el aspecto de esta página web.

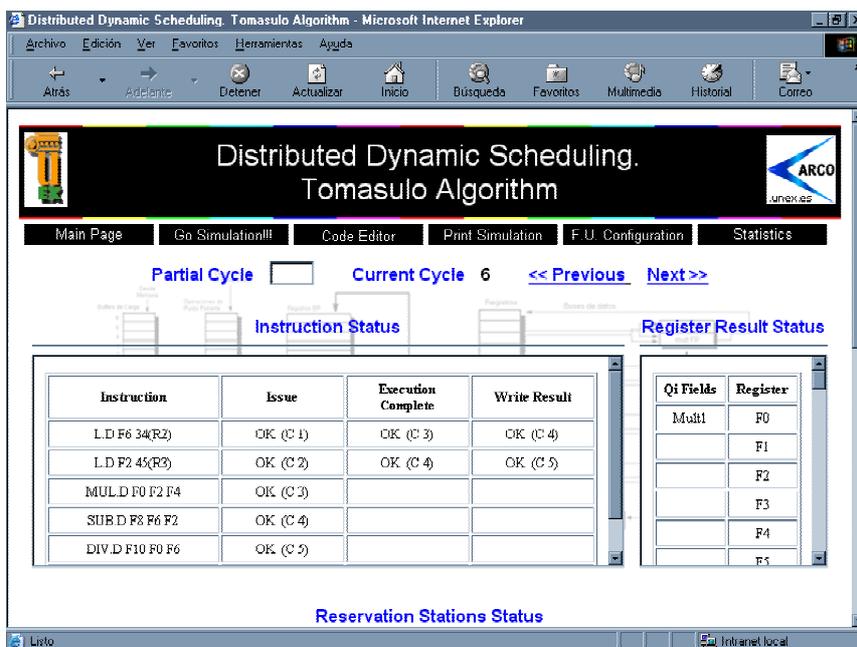


Figura 7. Parte superior de la página web del simulador para el algoritmo de Tomasulo.

Como se puede observar, la zona superior de la página está ocupada por un menú con las operaciones posibles. Estas opciones son muy parecidas a las explicadas para el método del Marcador en el apartado anterior. Posteriormente aparecen las opciones que controlan la evolución de la simulación: *Partial Cycle*, *Current Cycle*, *Previous* y *Next*; cuya funcionalidad es idéntica a la explicada anteriormente para el Marcador.

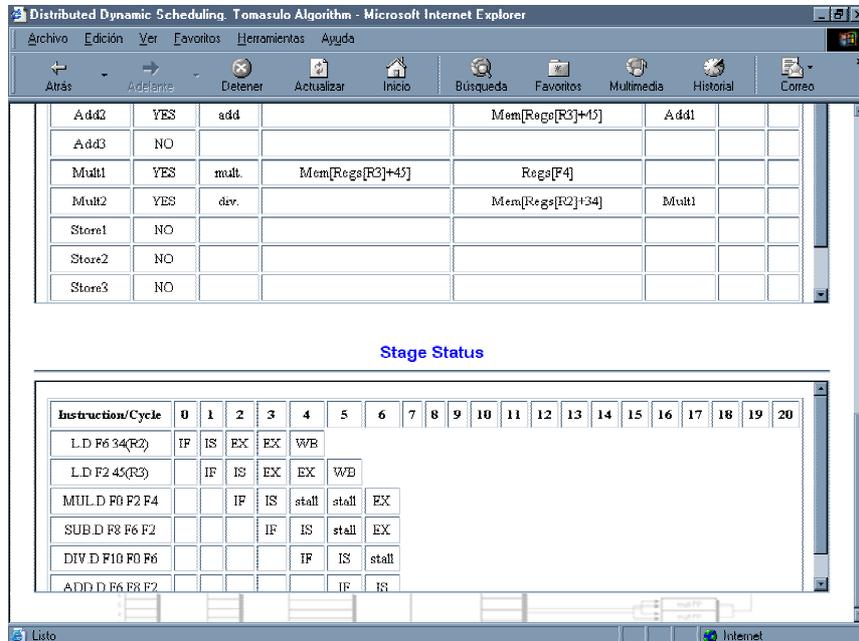


Figura 8. Parte inferior de la página web del simulador para el algoritmo de Tomasulo.

El resto de la página está ocupado por las tablas de estado que utiliza el algoritmo de Tomasulo. La *Tabla de Estado de las Instrucciones (Instruction Status)* muestra el estado de las instrucciones cargadas en el simulador, indicando en qué etapa se encuentra cada instrucción. La *Tabla de Estado de los Registros (Register Result Status)* presenta el estado de los registros destino, tanto coma flotante como enteros. La *Tabla de las Estaciones de Reserva (Reservation Stations Status)* muestra el estado de las estaciones de reserva que usa el algoritmo. Finalmente, la *Tabla de Estado de las Etapas (Stage Status)* refleja la evolución de las instrucciones por las distintas etapas de su ejecución, indicando las detenciones producidas, etc.

3.4. Editor de código

Si el usuario selecciona el botón *Code Editor*, de la página principal (figura 1), accederá al editor de código, donde podrá indicar el programa a simular. En la figura 9 puede apreciarse el aspecto del editor de código.

En la parte superior de la página aparece el menú que ofrece al alumno las distintas funciones del editor. La zona central de la página la ocupa una ventana de edición de texto (donde el usuario introducirá el código, las instrucciones, del programa a simular) e información sobre el subconjunto de instrucciones *MIPS64* que acepta el simulador.

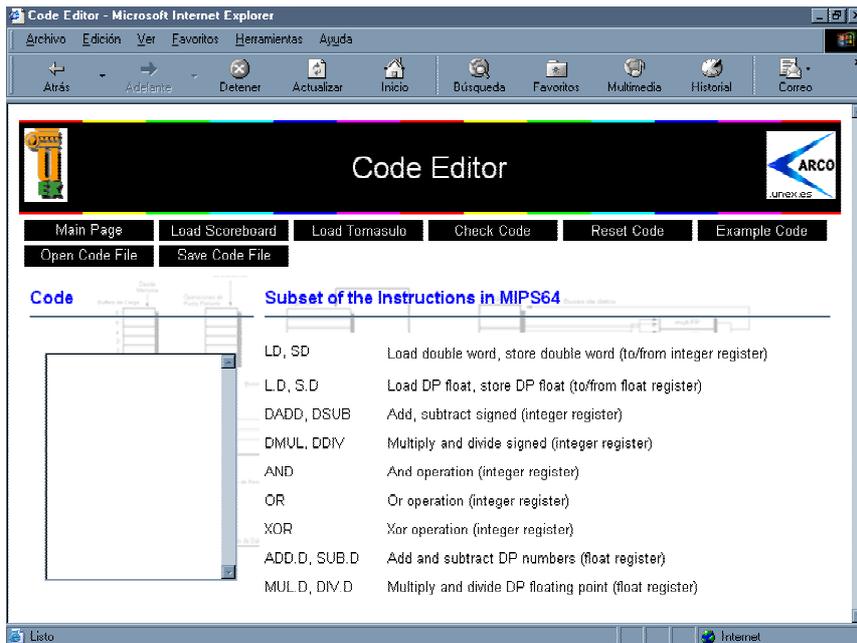


Figura 9. Página del editor de código de PDIWeb.

El menú del editor incluye las siguientes opciones:

- *Main Page*. Es un enlace a la página principal (figura 1) de la plataforma web.
- *Load Scoreboard*. Carga el código que se ha editado en el simulador, realizando la simulación mediante el método del Marcador. El código debe estar validado de forma correcta, sino aparecerá un mensaje de error indicando este hecho.
- *Load Tomasulo*. Carga el código que se ha editado en el simulador, realizando la simulación siguiendo el algoritmo de Tomasulo. Antes de cargar el código, éste debe estar validado.
- *Check Code*. Realiza una comprobación del código escrito en la ventana de edición de texto. En la parte superior de la página se mostrará una tabla con los resultados de esta validación, en la que se indican los errores producidos en cada instrucción, si los hubiera.
- *Reset Code*. Borra el código que se ha escrito en la ventana de edición de texto.
- *Example Code*. Muestra en la parte superior de la página una serie de ejemplos de programas escritos de forma correcta para el simulador.
- *Open Code File*. Permite al usuario abrir un fichero de disco para cargar el código almacenado en él. Tras seleccionar esta opción, aparecerá una ventana en la que el usuario puede examinar el disco duro (y otros dispositivos de almacenamiento masivo) para establecer la ruta del fichero a abrir. El fichero a abrir debe ser de tipo texto (*.txt).
- *Save Code File*. Permite al alumno guardar en un fichero de disco el código editado. Al elegir esta opción, el estudiante podrá examinar el disco duro (y otros dispositivos de almacenamiento) para establecer la ruta del fichero a guardar. El fichero a guardar debe ser de tipo texto (*.txt).

3.5. Teoría

Eligiendo el botón *Theory*, de la página principal (figura 1), el alumno accederá a una breve introducción acerca de los fundamentos teóricos en los que se basan el método del Marcador y el algoritmo de Tomasulo (ver figura 10).

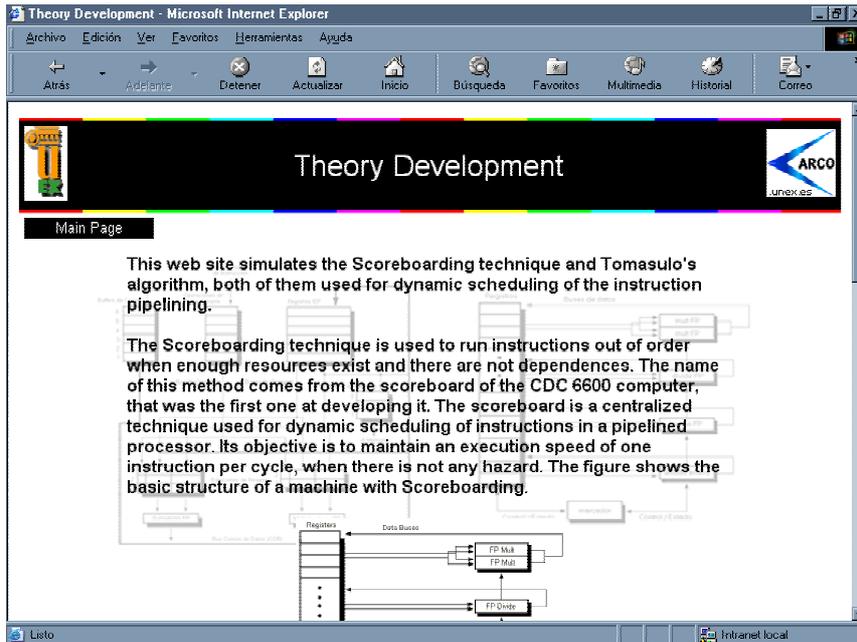


Figura 10. Desarrollo teórico dentro de la plataforma web.

3.6. Descargas

El botón *Download*, de la página principal (figura 1), permite acceder a la página de descargas de PDIWeb. En esta página el alumno puede bajarse del servidor material didáctico relacionado con el simulador. En la actualidad, por ejemplo, es posible la descarga de una aplicación local que simula el método del Marcador (ver figura 11), llamada “*Método del Marcador 2.0*”. Se trata de una aplicación para Windows, con una interfaz gráfica amigable y completa, realizada mediante entorno de ventanas. Esta aplicación ha sido desarrollada en Delphi, utilizando programación orientada a objetos.

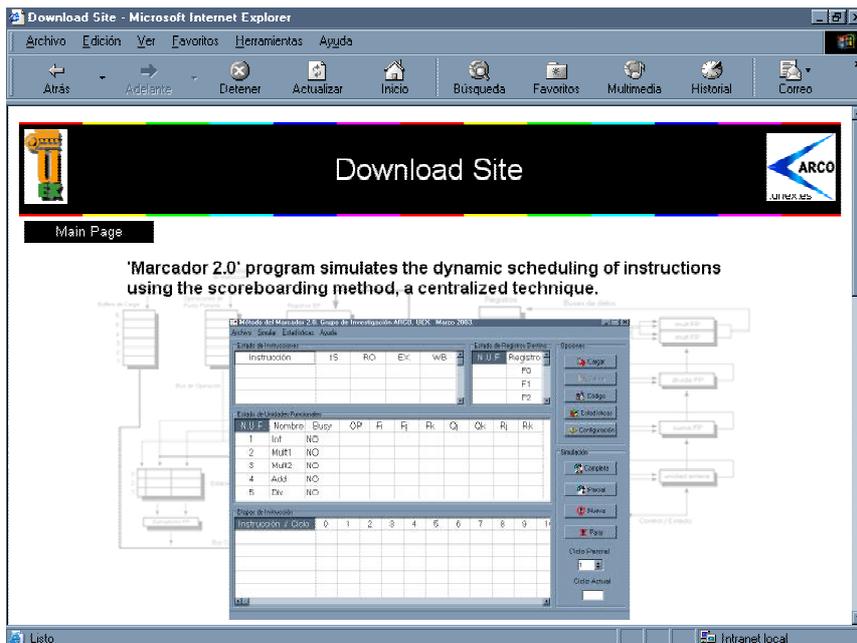


Figura 11. Página de descargas en PDIWeb.

La página presenta una breve descripción de cada aplicación que puede descargarse, mencionando sus características principales y mostrando imágenes de ella, para posteriormente presentar su ficha técnica (nombre del archivo de instalación, tamaño, idioma, etc.) y permitir su descarga.

3.7. Enlaces

Finalmente, con el botón *Links* de la página principal (figura 1), accedemos a una página web donde se muestran al usuario una serie de enlaces a sitios web que, por su contenido, guardan algún tipo de relación con PDIWeb. La figura 12 muestra el aspecto de esta página web.

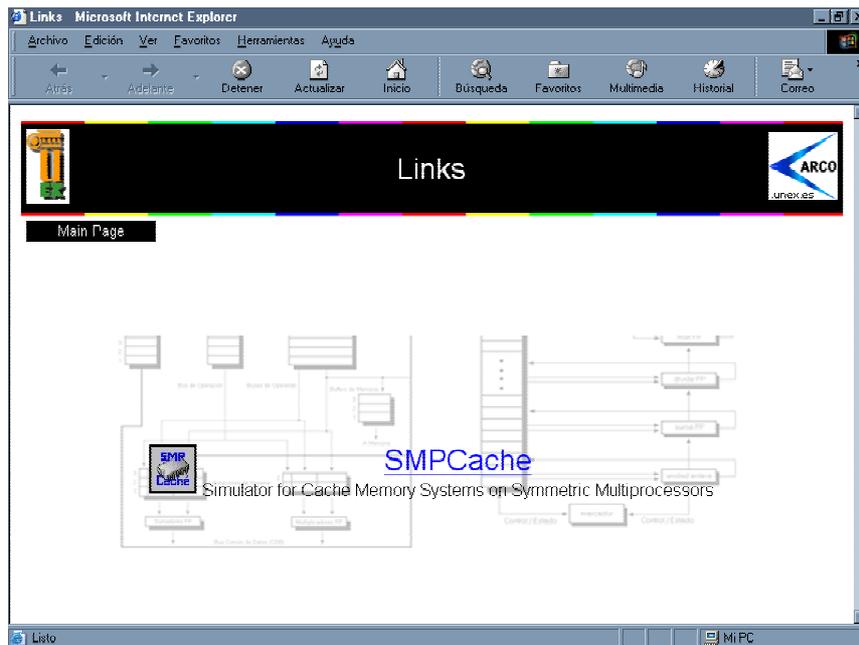


Figura 12. Página de enlaces dentro de la plataforma.

4. HERRAMIENTAS Y MÉTODOS

Para el desarrollo y mantenimiento de PDIWeb se dispone de un servidor que sirve de soporte al sistema docente a través de Internet. El servidor, bajo Windows 2003 Server y con Internet Information Server (IIS), ofrece servicios de administración y publicación de páginas web (WWW), de transferencia de ficheros (FTP), correo electrónico, listas de distribución de correo compatible con *Majordomo*, etc.

Para la implementación de PDIWeb se ha utilizado, principalmente, el lenguaje de programación *PHP 4.1* [2], [8]. *PHP 4.1* se ejecuta en servidores web y permite crear páginas *HTML* de forma dinámica. Además, se trata de un lenguaje similar a *ASP (Active Server Pages)* de Microsoft, pero más potente, rápido, gratuito, multiplataforma y abierto a mejoras y ampliaciones permanentemente. Debido a esto, *PHP* se está imponiendo a otros lenguajes de programación como alternativa para el desarrollo de aplicaciones informáticas en Internet.

Para algunas funciones concretas, ha sido necesario incluir *scripts* codificados en *JavaScript* [6], un lenguaje diseñado para que las aplicaciones desarrolladas en él soporten la ejecución remota. También existen fragmentos de código *HTML* [4], ya que al fin y al cabo el simulador está formado por una serie de páginas web. A pesar de esto, el 80% del código está implementado en *PHP*.

La edición de todo este código en *PHP*, *JavaScript* y *HTML* se ha realizado mediante la herramienta Macromedia Dreamweaver MX [5], una potente aplicación para la creación y mantenimiento de sitios web. Además, también se ha utilizado Macromedia Fireworks MX [1] para el diseño gráfico de diversos elementos web: imágenes, animaciones, textos *flash*, etc.

5. CONCLUSIONES

En este trabajo hemos presentado el sistema PDIWeb, que se ha desarrollado para la enseñanza a través de Internet de la planificación dinámica de instrucciones, materia impartida en la asignatura Arquitectura e Ingeniería de Computadores (AIC), de cuarto curso de la Ingeniería Informática, en la UEX.

Respecto a las ventajas del simulador, es importante resaltar que anteriormente esta parte del temario se explicaba mediante problemas hechos en clase. La realización de los problemas en la pizarra limitaba la versatilidad y cantidad de los mismos. El desarrollo de este simulador permite al alumno seguir practicando los conceptos aprendidos en su propia casa, y a su ritmo de aprendizaje. Además, la cantidad de ejercicios prácticos a realizar sólo está limitada por la cantidad de pruebas que quiera realizar el alumno con el simulador. De hecho, los simuladores son la mejor forma de comprender muchos de los conceptos prácticos relacionados con el diseño de procesadores, en los que interaccionan gran cantidad de elementos que dificultan la realización de problemas. En conclusión, el simulador refuerza los conceptos introducidos en clase, da a los alumnos una mejor apreciación del trabajo interno de un procesador, y motiva a los estudiantes a realizar más ejercicios prácticos, redundando en una mejora de la docencia.

Finalmente, indicar que PDIWeb no sólo es aplicable en la asignatura AIC de la UEX, sino que también puede usarse en otras muchas asignaturas sobre Arquitectura de Computadores que se imparten en la mayoría de Universidades, y en las que se explican los conceptos relativos a la planificación dinámica de instrucciones. En concreto, se puede acceder gratuitamente al simulador a través de la dirección [12].

6. AGRADECIMIENTOS

Este trabajo ha sido financiado en parte por el *Vicerrectorado de Innovación Educativa y Nuevas Tecnologías* y el *Instituto de Ciencias de la Educación (ICE)*, ambos de la Universidad de Extremadura, como Proyecto de Innovación Docente.

7. BIBLIOGRAFÍA

- [1] K. Cavanaugh, *Fireworks MX*, Anaya Multimedia, 2002.
- [2] F.J. Gil, *Creación de Sitios Web con PHP 4*, McGraw-Hill, 2001.
- [3] J.L. Hennessy, D.A. Patterson, *Computer Architecture: A Quantitative Approach*, Morgan Kaufmann, 3ª edición, 2003.
- [4] M. Morrison, *Diseño en HTML y XML*, McGraw-Hill, 2003.
- [5] O. Peña, *Dreamweaver MX*, Anaya Multimedia, 2002.
- [6] T.A. Powell, F. Schneider, *JavaScript. Manual de Referencia*, McGraw-Hill, 2002.
- [7] J.C. Sánchez, J.M. Sánchez, J.A. Gómez, *EDONET: Sistema Piloto para la Docencia a través de Internet*, VIII Congreso de Innovación Educativa en Enseñanzas Técnicas / I International Congress in Quality and in Technical Education Innovation, Donostia-San Sebastián, vol. 2, pp. 25-33, Septiembre 2000.
- [8] The PHP Group, *Sitio Web para PHP*, <http://www.php.net>, 2004.
- [9] M.A. Vega, I. Nieto, J.M. Sánchez, F. Chávez, J.A. Gómez, *System of Tele-Education for People with Hearing Disability*, International Conference on Information and Communication Technologies in Education, ICTE'2002, Badajoz, vol. 3, pp. 1261-1266, Noviembre 2002.

- [10] M.A. Vega, J.M. Sánchez, M. Rubio, J.A. Gómez, *Application of Information and Communication Technologies to the Teaching of Digital Systems*, Informatics Education 2002, VI Latin-American Congress on Computers & Education, IE'2002, Vigo, pp. 1-6, Noviembre 2002.
- [11] M.A. Vega, J.M. Sánchez, J. Ballesteros, *Sitio Web de la Asign. AIC*, <http://arco.unex.es/mavega/AIC.htm>, 2004.
- [12] M.A. Vega, J.C. Burgueño, J.M. Sánchez, J.A. Gómez, *Sitio Web para el Simulador PDIWeb*, <http://atc.unex.es/pdiweb>, 2004.