

PipeSim: Simulador para la planificación de unidades funcionales segmentadas

José M. Chaves González, Miguel A. Vega Rodríguez, Juan A. Gómez Pulido,
Juan M. Sánchez Pérez

Universidad de Extremadura. Departamento de Informática
Escuela Politécnica. Campus Universitario s/n
10071 Cáceres. Spain
{jm, mavega, jangomez, sanperez}@unex.es

Resumen

La segmentación, los procesadores segmentados y la planificación de los mismos son conceptos muy importantes en las arquitecturas de computadores de hoy día. Todos los procesadores actuales utilizan la segmentación para mejorar el rendimiento; de hecho, la eficiencia con la que se planifique el uso de los recursos de un procesador va a influir de manera muy significativa en el rendimiento del mismo. Por tanto, hablando de manera más general, los conceptos de *unidad segmentada* y de *planificación de unidades segmentadas* son temas muy relevantes en los estudios de Informática, por lo que es importante que dichos conceptos sean adquiridos por los alumnos de forma robusta y clara. Con el trabajo que presentamos en este artículo se ha querido dotar tanto a profesores como a alumnos de Informática con una herramienta que permite estudiar, enseñar y aprender de forma sencilla y completa cómo trabaja y se planifica una unidad segmentada (*pipeline*), ya sea estática o dinámica.

1. Introducción

La motivación para la construcción del simulador *PipeSim* surgió dentro de la asignatura troncal anual de 4º curso de la Ingeniería en Informática: Arquitectura e Ingeniería de Computadores (AIC). Esta asignatura, como algunas otras, tiene un temario extenso, por lo que no se dispone del tiempo deseable para abordar ciertas partes del mismo que tienen una complejidad especial. Entre estas partes se encuentran los conceptos de segmentación y planificación de unidades segmentadas, cuyo aprendizaje (tras las correspondientes explicaciones teóricas) pasa por la resolución de una serie de problemas que, aunque tienen una metodología bien marcada, no

son ni de resolución breve ni de resolución sencilla en la mayoría de las ocasiones.

La forma clásica de resolver este tipo de problemas es mediante la realización de ejercicios de forma manual en pizarra, lo cual, obviamente, limita la versatilidad y la cantidad de los mismos. Por esta razón los alumnos, quizás, no pueden disponer de un conjunto de ejemplos suficientemente amplio para comprender todos los casos que se pueden dar en la planificación de una unidad segmentada. Por este motivo, el uso de un simulador que genere, facilite la realización y corrija problemas de planificación es sumamente útil para el alumno. De hecho, se ha demostrado que la utilización de herramientas que ayuden y agilicen, a la vez que profundicen y mejoren la enseñanza de conocimientos de las partes más complejas e importantes de algunas asignaturas hace que el aprovechamiento de los alumnos sea mayor, ya que además de que el profesor cuenta con una herramienta más para explicar los conceptos y métodos en el aula, los alumnos pueden seguir practicando sobre dichos conceptos aprendidos en su propia casa, y a su ritmo de aprendizaje. En realidad, 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 concreto, *PipeSim* es una aplicación totalmente interactiva con la que se facilita al alumno la posibilidad de dominar los conceptos de planificación en unidades segmentadas mediante una interfaz muy sencilla y un funcionamiento muy descriptivo y didáctico, ya que además de tener un completo sistema de ayuda en el que el alumno puede apoyarse o incluso estudiar a partir de él, el simulador es totalmente flexible en cuanto a la visualización de los detalles que se

muestran. PipeSim es capaz tanto de mostrar el resultado final de la simulación, como de mostrar cualquiera de los pasos intermedios por los que hay que pasar cuando se está resolviendo un problema de forma manual. De esta manera, el alumno puede corregir y verificar un problema hecho a mano en todos los pasos por los que ha pasado para resolverlo, lo cual es muy ventajoso para su aprendizaje.

Hablando sinceramente, no conocemos ningún otro recurso docente similar, por lo que nos es imposible indicar las diferencias con él (ventajas e inconvenientes). De hecho, fue la inexistencia de recursos docentes en esta temática concreta lo que nos animó a desarrollar PipeSim.

2. Breve introducción teórica

Los conceptos más importantes sobre planificación y unidades segmentadas en los que se ha basado la construcción del simulador PipeSim se pueden encontrar en las referencias bibliográficas [2] y [7], en los capítulos correspondientes a esta temática. No obstante, a continuación se da una breve explicación de algunos conceptos relevantes a los que se hace referencia en este artículo.

Unidad segmentada (pipeline): Es una unidad funcional que divide la ejecución de sus operaciones en etapas, solapando la ejecución de éstas, y por tanto, ganando en rendimiento. En el caso de que se trate de una unidad segmentada de instrucciones, también se conoce como procesador segmentado, solapando la ejecución de múltiples instrucciones, por lo que mejora la productividad del procesador. Una unidad segmentada *estática* sólo puede ejecutar un tipo de operación a la vez, mientras que una *dinámica* puede ejecutar varios tipos al mismo tiempo (cada tipo pasa por distintas etapas de ejecución). PipeSim trabaja tanto con unidades segmentadas estáticas como dinámicas.

Tabla de reserva (TR): La TR es una matriz con tantas filas como etapas tenga la unidad segmentada y tantas columnas como ciclos de reloj necesite la operación que ésta describe. De esta manera, muestra cuándo están en uso las etapas de dicha unidad segmentada para una función particular.

Lista de latencias prohibidas (LP): Si no se evitaran las LP se permitiría que dos datos

colisionaran, es decir, llegaran a la misma etapa de la unidad segmentada al mismo tiempo. La lista LP es una lista de enteros que se corresponden a estas latencias prohibidas que hay que *evitar* en la planificación de una unidad segmentada.

Vector de colisión inicial (VCI): Es una cadena de dígitos binarios de longitud $N+1$, donde N es la latencia prohibida más grande dentro de la lista prohibida. El VCI, por tanto, se crea a partir de la lista de latencias prohibidas.

Diagrama de estados (DE): Se crea a partir del VCI. El DE es la representación (mediante un grafo multivaluado) de la ocupación de las etapas de la unidad segmentada para cada ciclo de reloj. Con dicho diagrama es más fácil planificar los datos de entrada a la unidad segmentada para que no tengan colisiones, lo cual se hace mediante el estudio de los ciclos del diagrama y la latencia calculada a partir de los mismos.

Técnica de inserción de retardos: A veces es posible modificar la tabla de reserva de una unidad segmentada sin que su funcionalidad cambie, pero su productividad final se incrementa. Si bien, con esta técnica, no es posible modificar la arquitectura de la unidad segmentada ni la operación que ésta debe realizar, es posible insertar retardos en algunos puntos determinados durante el transcurso de la ejecución de la operación con el fin de que se obtenga un aprovechamiento óptimo en la utilización de los recursos (etapas) de la unidad segmentada. La inserción de un retardo en la TR refleja la inserción de un cerrojo delante o detrás de la lógica de una etapa. La posición de los retardos se elige de tal forma que la fila de la tabla siga ciertos criterios que se desean obtener, pero en general se quieren los mínimos retardos posibles para conseguir un rendimiento óptimo de la unidad segmentada para la operación que está ejecutando y los recursos de que cuenta.

Síntesis de TR: Creación de una nueva TR, siguiendo un determinado método heurístico, a partir de una información específica, que puede ser un ciclo del DE o un VCI arbitrarios.

3. La interfaz de PipeSim

El simulador PipeSim se ha construido con objetivo didáctico, por lo que se ha puesto especial cuidado en que tanto su funcionamiento como la representación de los resultados que

obtiene sean lo más claros e instructivos posibles. Se ha construido siguiendo el Ciclo de Vida clásico de ingeniería del software [3] y se ha utilizado para su implementación el compilador *Borland C++ Builder 6* [1]. En cuanto a los requisitos hardware y software, PipeSim ha sido diseñado para que pueda ser utilizado en un amplio rango de sistemas, ya sean actuales o un poco más antiguos. Esto se ha hecho pensando en que los alumnos pueden tener un PC con características muy variadas en sus casas, por tanto, aunque PipeSim ha sido optimizado para su funcionamiento bajo un sistema actual con Windows XP, se ha probado satisfactoriamente en un amplio rango de equipos y sistemas (el equipo más antiguo fue un Pentium II a 300 MHz, con 96 megabytes de RAM y Windows 98 instalado). Además, la instalación del simulador en el disco duro ocupa menos de 5 megas, por tanto, se han puesto todas las facilidades posibles para que PipeSim pueda ser utilizado por la mayoría de los alumnos, o de las personas en general, que quieran hacerlo.



Figura 1. Ventana principal de PipeSim

En cuanto a las facilidades de funcionamiento, el simulador se diseñó con una finalidad didáctica bien definida desde sus primeras etapas de desarrollo. Debido a dicho objetivo final, el entorno de PipeSim está basado en ventanas, se ha procurado que sea muy fácil de manejar y se ha diseñado de la forma más atractiva y amigable posible, habiéndose cuidado mucho el aspecto final del mismo (botones con iconos descriptivos, gestión de ratón y de teclado cuidadas, explicaciones descriptivas de los pasos que se dan,

amplio sistema de ayuda, etc.). Además, el simulador controla todas las posibles causas de error durante su ejecución, mostrando al usuario un *gran número de mensajes de advertencia o error* claros y detallados.

Como se puede observar en la figura 1, su ventana principal es muy sencilla. En la parte superior de la misma hay un *menú de opciones* y en el resto de su espacio se reparten *9 botones*, agrupados por filas en tres categorías: En la fila superior se encuentran las opciones para *introducir una nueva tabla de reserva (TR) en el sistema*, ya sea estática o dinámica (rellenando la tabla con las marcas correspondientes en cada caso); en la fila central se encuentran las opciones para *cargar una TR en el simulador*, bien mediante *síntesis* (de un vector de colisión o de un ciclo del diagrama de estados) o mediante la carga de un fichero que contenga la información; y en la fila inferior se encuentran los botones para las opciones que son más útiles en el simulador (*Mostrar la TR que se encuentra cargada en el sistema*; *Mostrar el esquema hardware de la unidad segmentada* obtenido a partir de la TR cargada; y *Mostrar el diagrama de estados* obtenido en la planificación de la unidad segmentada. Este diagrama es la representación de la ocupación de la unidad segmentada en cada ciclo de reloj, por lo que se puede considerar la representación gráfica de la planificación de la unidad segmentada que se está estudiando.

Pero con los botones de acceso rápido no se puede controlar toda la funcionalidad de PipeSim. Para el control total del simulador ha de utilizarse su menú principal, que como se puede observar en la figura 2, tiene cuatro opciones principales.

Menú Tabla: Desde este menú se manipulan las tablas de reserva (TR) que maneja el simulador, tanto estáticas como dinámicas. Las operaciones que se pueden hacer con una TR (sin trasformarla, es decir, sin obtener la unidad segmentada que representa o la lista de latencias prohibidas que se consigue a partir de ella -como primer paso en la planificación-) son básicamente *crearla*, *verla* y *guardarla*. Por tanto, desde este menú se permite crear una nueva TR (de diversas formas), mostrar la TR que se encuentra cargada en el simulador, guardar la TR actual o cargar una desde un fichero en disco y, por ser el menú más general de la aplicación, desde éste también hay una opción para cerrar el simulador.

Tabla	Estática	Dinámica	Ayuda
Nueva ▶	Unidad segmentada	Unidad segmentada	Uso del simulador
Mostrar	Lista de latencias prohibidas	Listas de latencias prohibidas	Conceptos teóricos
Cargar...	Vector de colisión inicial	Matrices de colisión iniciales	Idioma
Guardar...	Diagrama de estados	Diagrama de estados	Acerca de...
Salir	Estudio de ciclos simples...	Estudio de ciclos simples...	
	Inserción de retardos...		
	Síntesis de tabla de reserva...		

Figura 2. Menú principal de PipeSim con sus opciones desplegadas

Menú Estática: Desde este menú se puede acceder a todas las operaciones que tienen que ver con tablas de reserva estáticas, es decir, con la planificación de unidades segmentadas estáticas. Estas operaciones son: dibujar la unidad segmentada (el hardware) que representa la tabla de reserva estática cargada en el simulador, obtener a partir de ella la lista de latencias prohibidas, calcular el vector de colisión inicial que se genera a partir de la lista de latencias, dibujar el diagrama de estados calculado a partir del vector de colisión, hacer un estudio de los ciclos de dicho diagrama de estados, insertar retardos en una TR que no tiene ninguna de sus etapas con ocupación máxima y sintetizar una tabla de reserva partiendo de un ciclo simple. A excepción de la síntesis de TR a partir de un ciclo simple, la secuencia del resto de operaciones forman los pasos que hay que dar para hacer un problema completo de planificación de unidades segmentadas estáticas. Sin embargo, no es necesario ir ejecutando paso a paso cada una de las opciones del menú, ya que si se pulsa una de ellas, por ejemplo la que corresponde con el *estudio de ciclos simples*, se ejecutarán los pasos anteriores necesarios para que la opción se pueda llevar a cabo y se muestre en ese momento lo que pide el usuario (automáticamente se calculan la lista de latencias prohibidas, el vector de colisión inicial y el diagrama de estados).

Menú Dinámica: Desde este menú se accede a las opciones relacionadas con la planificación de unidades segmentadas dinámicas. Estas operaciones son análogas a las de planificación estática, descritas anteriormente, pero más complejas, porque se realizan en una unidad segmentada dinámica (que realiza varios tipos de operaciones al mismo tiempo). Como en el caso

de la planificación estática, se puede ejecutar paso a paso cada una de las opciones que representan las etapas que se siguen en la planificación, o ejecutar la operación que se desee de forma directa. De esta manera, se dota al simulador de flexibilidad en su manejo.

Menú Ayuda: Desde este menú se accede a las opciones de ayuda, cambio de idioma y créditos del simulador. La ayuda está dividida en un completo *manual de usuario* de la aplicación y en una amplia explicación de todos los *conceptos teóricos* en los que se basa el simulador para su funcionamiento. Además, utilizando este menú es posible realizar el cambio de idioma (de español a inglés y viceversa) en cualquier momento de la ejecución del simulador. Toda la aplicación funciona en ambos idiomas, incluidos los ficheros de ayuda, el sistema de instalación, etc.

4. Características de PipeSim

Como ya se ha comentado, PipeSim es un simulador para uso didáctico, por tanto, una de sus principales características es la de ser una herramienta que ayude a las personas que lo usen a afianzar los conocimientos teóricos y sobretodo a adquirir los conocimientos prácticos de la parte de arquitectura de computadores referida a la planificación de unidades funcionales segmentadas. Pero a parte de la misión didáctica, PipeSim también es un entorno de simulación que permite el análisis y estudio detallado de la planificación de las citadas unidades segmentadas, un aspecto que es de gran interés dentro del campo de la segmentación y, por tanto, del paralelismo. El simulador tiene las siguientes características:

- Permite el estudio tanto de unidades segmentadas estáticas como de unidades segmentadas dinámicas, lo cual lo hace ser un simulador más completo y versátil.
- La entrada de datos al simulador puede hacerse de múltiples maneras, ya sea mediante la introducción manual de tablas de reserva (estáticas o dinámicas), o cargando la información desde ficheros, o mediante la inserción de un vector de colisión inicial, o incluso mediante la síntesis de una tabla de reserva a partir de un ciclo determinado del diagrama de estados que representa la ocupación de la unidad segmentada. Esto facilita la entrada de datos al simulador, a la vez que se permite que la heterogeneidad de los problemas que PipeSim puede realizar sea muy amplia.
- El simulador permite una configuración multi-idioma completa en español y en inglés, posibilitando el cambio de idioma en cualquier momento de la ejecución del mismo.

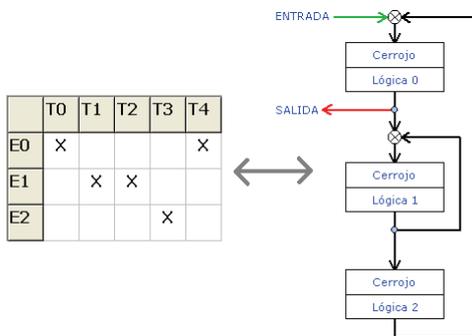


Figura 3. TR estática y unidad segmentada asociada

- Representación del hardware (dibujo en el que aparecen las etapas de la unidad segmentada y el camino que siguen las operaciones a través de ellas en cada ciclo de reloj). PipeSim puede representar unidades segmentadas de hasta 100 etapas, tanto estáticas como dinámicas (En las figuras 3 y 4 se muestran ejemplos de unidades segmentadas estáticas y dinámicas con 3 etapas -no se muestran ejemplos mayores por cuestiones de espacio-). Además, en los dibujos se distinguen las entradas y salidas de la unidad segmentada, así como sus

componentes hardware más importantes (camino de datos, etapas -con los cerrojos correspondientes-, multiplexores y demultiplexores).

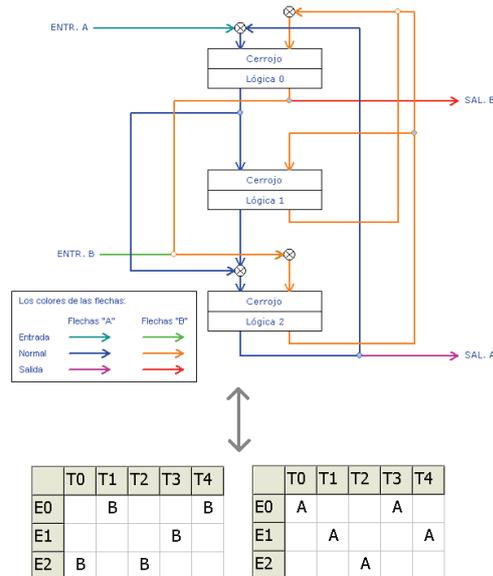


Figura 4. Unidad segmentada dinámica y TR que lo generan

- Cálculo de la lista (o listas en el caso dinámico) de latencias prohibidas para la tabla de reserva introducida.

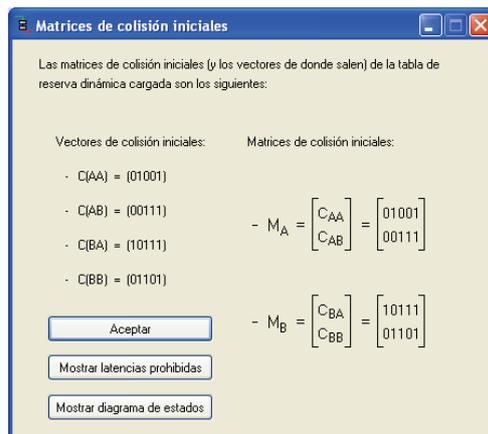


Figura 5. Matrices de colisión iniciales

- Cálculo del vector de colisión inicial (o de las matrices de colisión iniciales en el caso dinámico) que se genera a partir de las lista (o listas) de latencias anteriores. La figura 5 muestra un ejemplo.
- Obtención y representación del diagrama de estados que se obtiene del vector de colisión inicial (o de las matrices de colisión) de la tabla de reserva cargada, y que muestra los diferentes estados de una unidad segmentada para un periodo de tiempo dado.

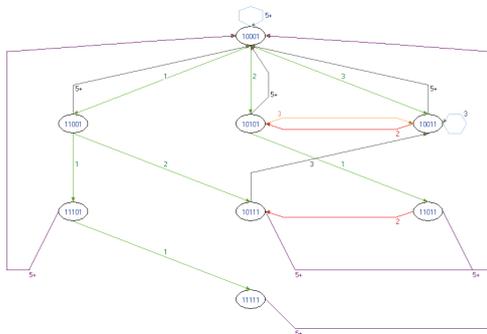


Figura 6. Diagrama de estados estático

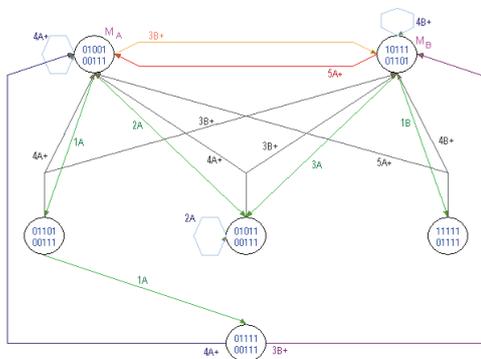


Figura 7. Diagrama de estados dinámico

Como se puede ver en las figuras 6 y 7, se emplean distintos colores y una distribución clara para que el diagrama de estados, ya sea estático o dinámico, sea perfectamente legible. El algoritmo de dibujado de *cualquier diagrama de estados* que se genere a partir de unos datos arbitrarios introducidos por el usuario es una de las partes más complejas de PipeSim debido a que es posible representar

diagramas de estados de hasta 128 estados, por lo que el grafo que representa el diagrama de estados puede llegar a ser realmente complejo. En estos casos es muy útil la opción que ofrece el simulador de ver el diagrama de estados en forma de tabla (matriz de adyacencia). La figura 8 muestra la matriz de adyacencia que representa al diagrama de la figura 6.

	10001	11001	10101	10011	11101	10111	11111
10001	5+	1	2	3
11001	5+	1	2	..
10101	5+	3	1
10011	5+	..	2	3
11101	5+	1
10111	5+	3
11011	5+	2
11111	5+

Figura 8. Diagrama de estados en forma de tabla

- Estudio de los ciclos simples del diagrama de estados (ver figura 9), tanto estático como dinámico. Sirve para determinar cuándo una unidad segmentada es óptima (su ocupación es máxima en alguna etapa) para los datos de entrada introducidos; o si no es óptima, cómo podemos modificar la disposición de dichos datos para que la productividad sea óptima. El simulador, por tanto, realiza un estudio completo del rendimiento que alcanza la unidad segmentada, dando una completa información al usuario de los datos que se obtienen (periodo y latencia de cada ciclo, mínima latencia media, mínima latencia constante, ciclos avariciosos...).

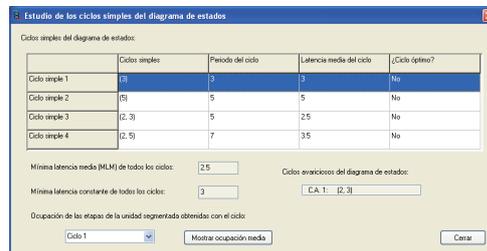


Figura 9. Estudio de los ciclos simples

- Estudio de la mejora del rendimiento mediante inserción de retardos (con la posibilidad de

explicar varios ejemplos en un tiempo mucho menor al que necesitaría para hacerlo a mano (en pizarra) y de una manera mucho más clara y dinámica que si utilizara transparencias o PowerPoint. Esto se traduce en que los conceptos de planificación son explicados en mayor profundidad y con mayor claridad que si se hiciera de forma manual, ya que hay más tiempo para explicar más casos y problemas.

Tras las explicaciones del profesor, el alumno puede descargarse, del sitio web habilitado para el efecto, el instalador del simulador. Cuando ejecute este archivo en su ordenador, se instalará el simulador en sí, su completo sistema de ayuda - que además de contener una explicación completa de cómo funciona PipeSim, tiene una amplia explicación de la teoría en la que se fundamenta el simulador- y algunos ejemplos representativos que el alumno podrá cargar para realizar sus primeros ejercicios. El alumno utiliza el simulador en su casa en la medida que necesite, y aprende a su ritmo, con el número de problemas que considere necesario, cómo funciona y se planifica una unidad segmentada. En este sentido, el profesor propone problemas adicionales que no soluciona en clase, y que el alumno podrá realizar (a mano) y verificar utilizando el simulador. De esta manera, se pueden realizar muchos más problemas y verificar todos los ejercicios que se puedan proponer. Para hacer más atractiva esta tarea, los problemas propuestos por el profesor son problemas incluidos en una relación de "Problemas de Examen", es decir, problemas que se han puesto en exámenes anteriores de la asignatura. Ésta es una motivación extra para el alumno. En todo caso, el profesor siempre informa de que él dispone de todas las soluciones de los problemas de examen y que si alguno no queda claro, incluso usando PipeSim, el alumno en cuestión debería acercarse a tutorías.

Por tanto, el simulador es una herramienta para agilizar y potenciar las clases, y para que el alumno pueda estudiar, aprender y profundizar en esta importante parte de la asignatura.

6. Conclusión

Se ha conseguido construir un simulador muy completo y versátil acerca de la planificación de unidades segmentadas. De hecho, todo lo que puede hacer el simulador no se ha descrito en este

artículo por falta de espacio. Si bien, quizás, el simulador podría utilizarse con otros fines (investigación), fue pensado, y está siendo utilizado como aplicación docente. Además está dirigida a los alumnos, inexpertos al principio en la materia que se enseña, por lo que se ha puesto especial cuidado en ir mostrando resultados intermedios, y explicar todos los pasos que se dan en la simulación. Por tanto, aunque PipeSim es utilizado por el profesor para hacer problemas de planificación en clase, es una herramienta pensada para que sean los alumnos los que principalmente la utilicen en sus casas, haciendo tantos problemas como necesiten para aprender los conceptos. La ventaja de este método de trabajo es que cada persona aprende a su ritmo de aprendizaje, explotando el simulador tanto como necesite, por lo que en general, tanto cada alumno de manera individual, como la clase en su conjunto obtienen más beneficios que si no se contara con este tipo de herramientas.

Finalmente, indicar que el simulador está disponible de forma gratuita, con fines docentes y de investigación, en [6].

Referencias

- [1] Charte, Francisco. Programación con C++ Builder 5. Anaya Multimedia, 2000.
- [2] Hwang, Kay. Advanced Computer Architecture: Parallelism, Scalability, Programmability. McGraw-Hill, 1993.
- [3] Pressman, Roger S. Ingeniería del Software: Un Enfoque Práctico. McGraw-Hill, 2002.
- [4] Vega, Miguel A., Sánchez, Juan M., Gómez, Juan A. An Educational Tool for Testing Caches on Symmetric Multiprocessors. *Microprocessors and Microsystems*, vol. 25, no. 4, pp. 187-194, Junio 2001.
- [5] Vega, Miguel A., Burgueño, J. Carlos, Sánchez, Juan M., Gómez, Juan A. Web Simulator for the Teaching of Instruction Dynamic Scheduling. 6th International Symposium on Computers in Education, Cáceres, España, pp. 1-8, Noviembre 2004.
- [6] Vega, Miguel A., Chaves, José M. PipeSim. <http://arco.unex.es/pipesim>, 2006.
- [7] Zargham, Mehdi R. Computer Architecture: Single and Parallel Systems. Prentice-Hall, 1996.