

# Herramientas Docentes basadas en Internet para la docencia de Arquitectura de Computadores

**José Ignacio Benavides Benítez**

Universidad de Córdoba. Depto. Elect. y Electrónica  
Córdoba. SPAIN.14071  
el1bebej@uco.es

and

**Ezequiel Herruzo Gómez**

el1hegoe@uco.es

and

**José Manuel Palomares Muñoz**

el2pamuj@uco.es

## **Abstract**

Simulators have been proved to be a very useful educational tool. Two simulator programs have been designed: SICAM and SICOME, which perform an interactive simulation on Multilevel Cache Systems and a Simple Computer Architecture, respectively. These programs help the students to make practices reaching higher levels of comprehension on the subjects. However, traditional simulators present the problem of disponibility. To avoid this issue, SEPRAC was developed. SEPRAC is a practice server which allows Internet connections to make and submit practices via remote. This avoids the fact of making the student to physically move to the computers at the University. Nevertheless, this advantage also produces somewhat an isolation of the student out from his/her teaching enviroment. This is due to the fact that students make their practices alone, putting themselves away from colleagues and professors. To avoid it, SAVIAC has been designed. It is a Virtual Chatting Room server which develop remote tutorials and chatting rooms about the subjects.

**Keywords:** Simulators, Computer Architecture, Cache Memory, Internet, e-learning

## **Resumen**

Se ha demostrado ampliamente que los simuladores son una herramienta docente muy útil. Se presentan dos simuladores: SICAM y SICOME, que realizan una simulación interactiva sobre sistemas caché multinivel y sobre una arquitectura de Computadora Sencilla, respectivamente. Estos programas permiten a los alumnos realizar prácticas que mejoran su grado de comprensión de la asignatura, sin embargo, presentan el problema de la disponibilidad de dichos programas. Para evitar esta falta de disponibilidad se ha desarrollado SEPRAC, un servidor de prácticas, que permite la conexión al mismo a través de Internet, pudiendo realizar y entregar de manera remota las prácticas sin necesidad de un desplazamiento del alumno físicamente a los ordenadores de la Universidad. Se podría pensar que esta mejora produce un cierto aislamiento al trabajar el alumno en solitario, alejándole del entorno de compañeros y profesores. Para evitarlo se ha diseñado SAVIAC, un servidor de Salas Virtuales, que permite la conexión para realizar tutorías remotas o salas de conversación sobre las asignaturas.

**Palabras claves:** Simuladores, Arquitectura de Computadores, Memoria Caché, Internet, enseñanza a distancia

## 1 Introducción

El Área docente de Arquitectura y Tecnologías de Computadores ha desarrollado un conjunto de programas de ordenador que apoyan la docencia de las diversas asignaturas que imparte este área. Estos programas implementan diversos tipos de simuladores, desde simuladores de memoria caché hasta simuladores de procesadores microprogramados simples. Durante varios cursos se han impartido la docencia de asignaturas utilizando como herramientas de apoyo para las clases prácticas estos simuladores, que eran proporcionados por el profesorado a los alumnos para que pudieran utilizarlos en sus domicilios. Al abrigo de las nuevas tecnologías, y con el auge de Internet, se han integrado estos simuladores en un entorno de trabajo unificado, que permite no ya la simple ejercitación de las prácticas, sino que facilita el proceso de entrega de los ejercicios prácticos realizados, así como la interacción directa con el profesor para la resolución de problemas y la realización de teletutorías. Además se ha hecho hincapié en el aspecto multiplataforma y en los sistemas basados en Internet para poder trabajar con dichas herramientas independientemente del lugar, momento y sistema operativo con el que el alumno cuente.

Para el entorno de trabajo que se presenta se pueden agrupar en dos conjuntos los programas que se han desarrollado:

**SIMULADORES** Que incluyen:

**SICAM** es un simulador de Memoria Caché Multinivel.

**SICOME** es un simulador de Computadora Sencilla[1].

**SISTEMAS SERVIDORES** Que incluyen:

**SEPRAC** es el sistema Servidor de Prácticas.

**SAVIAC** es un sistema de Salas y Tutorías Virtuales.

## 2 Simulador de Caché Multinivel (SICAM)

Muchos son los libros ([2], [10], [11], ...) que explican con ejemplos el funcionamiento de una jerarquía de memoria, sin embargo para el alumnado el seguimiento de dichos ejemplos se hace complejo en muchos casos. Las computadoras actuales están incorporando varios niveles de caché, consiguiendo mayores rendimientos del sistema de memoria al minimizar el tiempo de espera de datos de la memoria. Desde el área de Arquitecturas y Tecnologías de Computadoras se observó la posibilidad de desarrollar una herramienta docente que permitiera mostrar a los alumnos la problemática que rodea a la jerarquía de memoria. Con esta herramienta se pretende conseguir dar una concepción lo más real posible al comportamiento de los distintos niveles de caché y cómo fluyen los datos a través de esa jerarquía, pudiendo el alumno comprobar el rendimiento que ofrecen los sistemas con distintas configuraciones de caché, e incluso decidir la configuración óptima para la ejecución de determinados programas.

Existen varios simuladores del funcionamiento de la caché:

**Cache Simulator** es un programa implementado en Java, que se encarga de traducir una dirección de memoria y mostrar como quedaría en caché mediante una interfaz gráfica.

**Dinero IV** es un simulador de caché que utiliza trazas de referencia de memoria. No tiene en cuenta los ciclos o el tiempo simulado.

**PLX Cache Simulator** es un simulador que está integrado dentro de un simulador de sistema de mayor nivel. Permite dar un informe preciso sobre el comportamiento de la memoria y de las instrucciones de código del núcleo. Se pueden configurar un gran número de parámetros de caché. Sin embargo, las salidas no son demasiado explicativas ni intuitivas.

**Sismec** es una aplicación que se encarga de evaluar los sistemas multicaché, permitiendo estudiar el sistema de memoria que mejor se adapte a nuestras necesidades antes de su implementación física o simplemente simular sistemas actuales para medir su eficacia y poder comparar resultados de forma fácil.

**SMPCache** permite la evaluación de sistemas jerárquicos de memoria cachés en sistemas multiprocesadores simétricos con memoria compartida por bus. Se muestra en cada instante los accesos a memoria realizados por cada procesador, el estado del bus, de cada caché y de cada bloque dentro de la caché.

**mlCache** es un simulador de caché multilateral (es decir que posee dos o más almacenes de datos que tienen contenidos diferentes entre sí y que pueden operar de forma paralela), dirigido por eventos y sensible al tiempo. Se basa en el modelo de tiempo de caché *Latency Effects*.

**CACTI** es un simulador al que se le pasan los parámetros de la caché que deseemos construir (tamaño de bloque, tamaño de caché, asociatividad, ...) y nos muestra los parámetros físicos necesarios para su implementación (por ejemplo, voltaje, tiempos de acceso, ...).

**XCACHE** [7]. Este simulador estudia el funcionamiento de la memoria caché realizando la simulación de los accesos a memoria de un programa. Es capaz de simular dos tipos de caché: unificadas (no se distinguen datos de instrucciones) y mixtas (existiendo una caché sólo para datos y otra caché sólo para instrucciones). También es posible configurar el tamaño de las memorias caché, el tamaño de los bloques, las políticas de reemplazo y el tipo de escritura. Sin embargo, el principal inconveniente es que los programas se deben introducir en un lenguaje de referencias a memoria, significando para el usuario un duro trabajo de codificación de accesos a memoria para poder transcribir el comportamiento de los programas. Otro gran inconveniente, es que sólo permite trabajar con un único nivel de caché, sin opción de configurar un sistema multinivel. Por último, las estadísticas generadas son pobres y de difícil comprensión, además de no realizar medición alguna sobre los tiempos de acceso a memoria durante la simulación.

De esta manera, el simulador *SICAM* permite simular hasta tres niveles de caché. Todos los niveles son completamente configurables (tamaño, políticas y tipo), permitiendo observar el rendimiento del sistema para distintas configuraciones de memoria. Además, durante la simulación permite visualizar en tiempo real los accesos a memoria y el tráfico de datos entre los distintos niveles. Las estadísticas que se presentan son gráficas y de fácil comprensión permitiendo comprobar la evolución de los accesos a los niveles de caché durante la simulación. Se pueden calcular los tiempos que intervienen en los accesos a memoria. También se puede configurar el ancho de los buses de datos que conectan los niveles de memoria.

Para evitar la ardua tarea de escribir la codificación de los accesos de memoria, se incluye un traductor de instrucciones que permite pasar de código ensamblador a referencias de memoria, de esta forma se facilita la codificación de la simulación de referencias a memoria. El código ensamblador que se introduce es un subconjunto del repertorio de un procesador *MIPS* [8].

Los resultados se almacenan en un fichero externo de tipo texto sencillo, donde el alumno puede deducir y documentar sus conclusiones para una posterior entrega de las prácticas realizadas al profesor.

## 2.1 Traductor de Instrucciones MIPS

El traductor de instrucciones facilita la introducción y edición del código a simular, permitiendo ejecutar dicho código y observar la modificación de los registros durante la ejecución del programa. Se ha implementado una arquitectura de procesador tipo RISC de la familia MIPS con 32 registros. El traductor trabaja con un subconjunto de 32 instrucciones de las que se pueden encontrar en un MIPS. Mediante un sencillo interfaz se pueden introducir el programa a simular y los valores iniciales de los registros. También se permite introducir un programa mediante un simple fichero de texto.

El conjunto de instrucciones que reconoce el simulador es amplia y suficiente, se muestra a continuación:

- **Instrucciones Aritméticas y Lógicas:**

**Add / Addi** Suma entre registros y Suma Inmediata (respectivamente).

**And / Andi** Operador lógico AND entre registros e Inmediata (respectivamente).

**Div** División.

**Mul** Multiplicación.

**Nor** Operador lógico NOR entre registros.

**Or / Ori** Operador lógico OR entre registros e Inmediata (respectivamente).

**Sllv** Desplazamiento lógico de bits hacia la derecha.

**Srlv** Desplazamiento lógico de bits hacia la izquierda.

**Sub / Subi** Resta entre registros e Inmediata (respectivamente).

**Xor / Xori** Operador lógico XOR entre registros e Inmediata (respectivamente).

Address	Assembly	Register 1	Register 2	Register 3	Immediate
00006000	MOVE	\$t0	\$t1		
00006004	MOVE	\$t2	\$t3		
00006008	ADD	\$t4	\$t5	\$t6	
0000600C	BUCLE1: SLT	\$t7	\$t8	\$t9	
00006010	BEQ	\$t0	\$t1		SALIR1
00006014	ADDI	\$t7	\$t8		FFFF
00006018	BUCLE2: SLTI	\$t8	\$t9		0
0000601C	BNEQ	\$t8	\$t9		SALIR2
00006020	MBL	\$t7	\$t8	\$t9	
00006024	MFLC	\$t5			
00006028	ADD	\$t5	\$t6	\$t7	
0000602C	LW	\$t4		0(\$t5)	
00006030	LW	\$t5		4(\$t5)	
00006034	SLT	\$t8	\$t5	\$t4	
00006038	BEQ	\$t8	\$t5		SALIR2
0000603C	MOVE	\$t4	\$t5		
00006040	MOVE	\$t5	\$t7		
00006044	JAL	\$t7			ORCEN
00006048	ADDI	\$t7	\$t7		FFFF

Figura 1: Traductor de Instrucciones MIPS

- **Instrucciones de Transferencia de Datos:**

- Li** Carga inmediata en un registro.
- Lui** Carga inmediata en la parte superior de un registro.
- Lw** Carga una palabra de memoria.
- Sw** Almacena una palabra en memoria.

- **Instrucciones de Salto Condicional:**

- Slr / Slr** Inicializa un registro a 1 si otro es menor que otro tercero, inicializa a 0 en caso contrario y realiza la comparación inmediata (respectivamente).
- Beq** Salta si dos registros son iguales.
- Bneq** Salta si dos registros no son iguales.

- **Instrucciones de Bifurcación Incondicional**

- J** Bifurcar incondicional a una instrucción de destino.
- Jal** Bifurcar incondicionalmente a una instrucción de destino, guardando en el Registro 31 la instrucción siguiente a ésta (dirección de retorno).
- Jr** Bifurca incondicionalmente a la dirección que esté contenida en un registro, guardando en el Registro 31 la instrucción siguiente a ésta (dirección de retorno).

- **Instrucciones de Desplazamiento de Datos:**

- Move** Transfiere el contenido de un registro a otro.
- Mfhi** Transfiere el contenido de la parte alta de un registro a otro.
- Mflo** Transfiere el contenido de la parte baja de un registro a otro.

## 2.2 Simulador Caché Multinivel

El simulador de Caché Multinivel trabaja con un código de referencias a memoria que se obtiene a partir del código del usuario mediante el traductor de instrucciones comentado en el punto anterior. La referencia de memoria se compone de tres campos:

- *Nº de referencia a Memoria.* Indica el número de referencia a memoria dentro del programa a simular.
- *Tipo de Acción.* Indica si el acceso es una lectura o una escritura. Los tipos pueden ser:
  - LI** Indica que es una lectura de Instrucción.

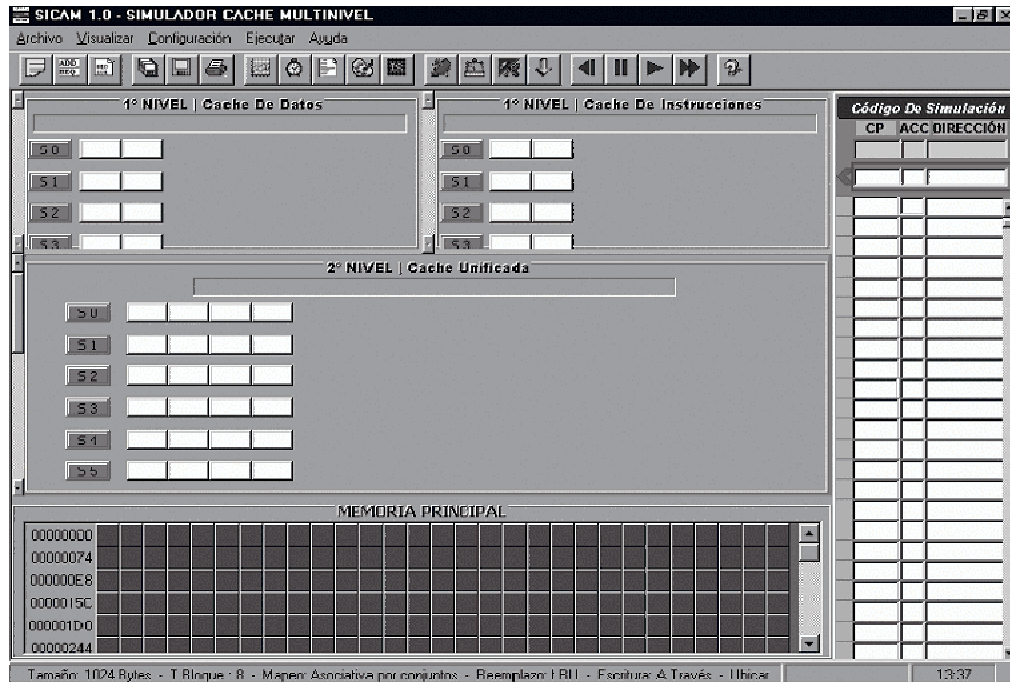


Figura 2: Pantalla principal del simulador SICAM

**LD** Indica que es una lectura de un Dato.

**E** Indica que es una escritura de un Dato.

- *Dirección*. Dirección en memoria principal donde se encuentra el dato o instrucción a tratar.

## 2.3 Memoria Caché: Configuración

Como se ha comentado previamente, el programa permite simular sistemas de hasta tres niveles de caché en los que se pueden configurar en todos ellos:

- Tamaño de la memoria de dicho nivel y tamaño de bloque.
- Políticas de Reemplazo: LRU, FIFO, Aleatoria.
- Formato de Caché: Unificada y Mixta.
- Políticas de Escritura: Escritura Directa y Escritura Diferida.
- Políticas en caso de Fallo de Escritura: Ubicar en Escritura y No Ubicar en Escritura.
- Tipos de Caché: Asociativa, Asociativa por Conjuntos y de Correspondencia Directa.

## 2.4 Estadísticas y medidas de tiempo de simulación

Durante la simulación se muestran estadísticas para cada uno de los niveles de caché del sistema. También se contabilizan las veces que se accede a memoria principal. Se presentan estadísticas de:

- *Tasa de Aciertos*. Número de aciertos que se producen sobre el nivel de caché que se representa. El porcentaje es sobre los accesos totales que se realizan.
- *Aciertos de Lectura*. Es una medida de las veces que se accede a un cierto nivel a leer un dato o una instrucción y se encuentra dicho dato allí. El porcentaje se realiza sobre los accesos de lectura totales.
- *Aciertos de Escritura*. Contabiliza las veces que se accede a un nivel para realizar una escritura y se encuentra el bloque de información. El porcentaje se realiza sobre los accesos totales de escritura.
- *Tasa de Fallos*. Muestra el total de fallos que se han producido en un nivel. El porcentaje se calcula sobre los accesos totales a ese nivel.

- *Fallos Forzosos*. Número de fallos forzosos que se dan en un nivel. El porcentaje viene dado por los fallos totales en ese nivel.
- *Fallos de Capacidad*. Representa los fallos que se producen cuando un nivel no contiene datos (ese nivel de caché está vacío). El porcentaje se calcula sobre los fallos totales en ese nivel.
- *Fallos de Conflicto*. Son los fallos de conflicto que se han producido en dicho nivel. El porcentaje se calcula sobre los fallos totales en ese nivel.
- *Fallos de Lectura*. Son los fallos que se producen en un nivel al intentar leer una instrucción o un dato y no encontrarse en dicho nivel. El porcentaje se consigue dividiendo la cantidad de fallos de lectura entre los accesos de lectura totales en ese nivel.
- *Fallos de Escritura*. Contabiliza los fallos producidos por el acceso a un nivel para la modificación de una posición de memoria inexistente en dicho nivel. El porcentaje se calcula sobre los accesos de escritura totales.
- *Accesos Totales*. Identifica todos los accesos, tanto de lectura como de escritura, que se producen a los diversos niveles de caché.
- *Accesos de Lectura*. Número de lecturas que se realizan sobre un determinado nivel de caché, su porcentaje se genera sobre los accesos totales.
- *Accesos de Escritura*. Número de escrituras que se realizan sobre un determinado nivel de caché. Su porcentaje viene dado sobre los accesos totales.

Las Estadísticas se muestran durante el transcurso de la simulación y su representación en forma de gráfica hace que sean fáciles y rápidas de comprender. Las gráficas se realizan sobre un par de ejes cartesianos, donde el *Eje X* representa el valor del número de referencias a memoria que contiene el código simulado y en el *Eje Y* se representa el elemento estadístico determinado para una referencia en particular de memoria. Además, la representación gráfica se acompaña de los porcentajes para cada elemento estadístico.

Por otra parte, el simulador puede ser configurado con los tiempos de acceso para cada uno de los niveles de memoria que forman la jerarquía. En particular, para cada nivel de caché se pueden establecer valores para los tiempos de: latencia, transferencia y escritura. Estos tiempos vienen especificados en ciclos de reloj. De esta manera, el simulador estimará los ciclos necesarios para completar la operación que se solicite en cada instrucción, en función del nivel de caché donde se encuentre de manera efectiva el dato y los valores temporales proporcionados. Si el dato no se encuentra en un cierto nivel de caché, SICAM realizará además una medición del tiempo de penalización.

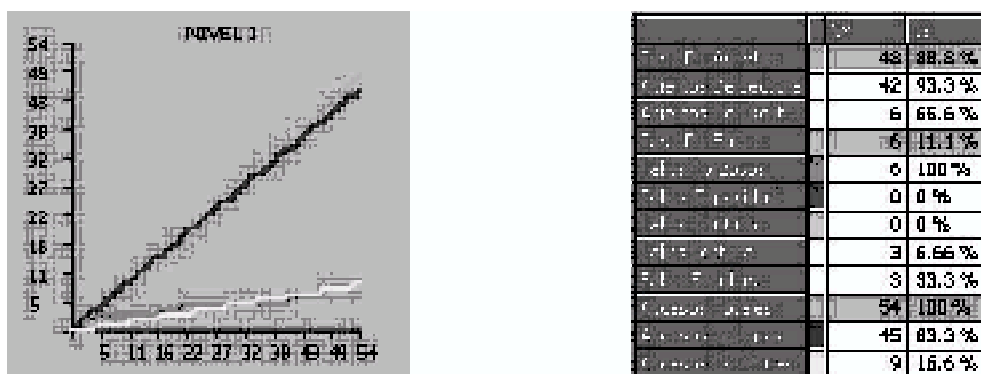


Figura 3: Estadísticas que se obtienen por el simulador SICAM

SICAM permite modificar el rendimiento del sistema de memoria, determinando la configuración de los buses de datos que conectan los diversos niveles de la caché, aumentando o disminuyendo el ancho de los buses. También se permite la modificación de la interconexión de los diversos niveles con el procesador: conexión en serie o en paralelo.

### 3 Simulador de la Computadora Mejorada (SICOME)

La Computadora Mejorada ([1], [2]) es un sistema ampliamente utilizado como herramienta básica para que los alumnos adquieran conocimientos sobre aspectos elementales de la Arquitectura de Computadores. Previamente al desarrollo de este simulador, ya existían simuladores de la Computadora Mejorada en disposición de los docentes del Departamento. Sin embargo, estos simuladores presentaban carencias claras y se decidió subsanar y mejorar esos defectos. Basándonos en la arquitectura original, se han realizado pequeños retoques que mejoran claramente y dan una mayor versatilidad a esta nueva Computadora Mejorada sin restar su sencillez y su eficacia docente.

El simulador consta de una pantalla principal en la cual se muestra el esquema de la Computadora Mejorada, la representación de la memoria y una consola de estado. En el diagrama de la Computadora Mejorada se muestra el contenido de los registros y de la memoria, la unidad aritmético-lógica (que incorpora un registro *QR* para algoritmos de multiplicación y división) y el controlador, además de las señales, los buses activos y un gran número de microoperaciones disponibles. La zona de memoria está formada por 10 campos que indican palabras de memoria, cada una identificada con su dirección. Por último, la consola de status que es un medio más para el control del comportamiento y el estado del simulador en cada momento.

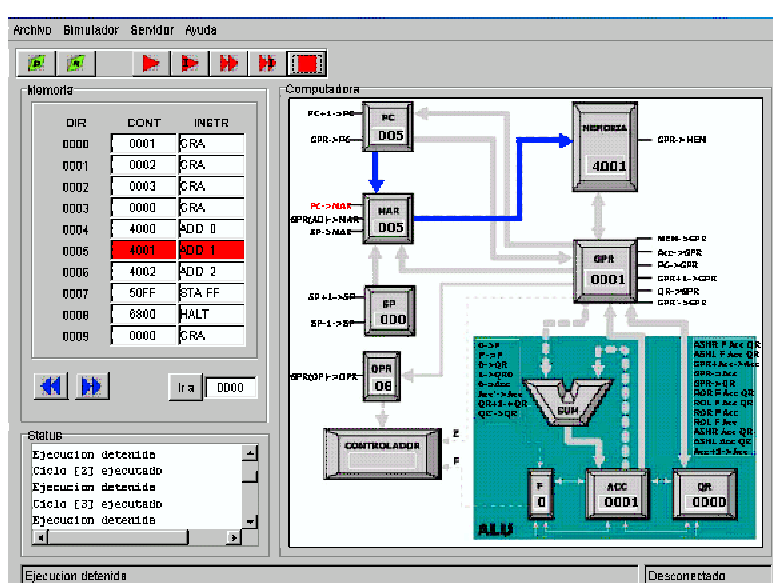


Figura 4: Pantalla principal del simulador SICOME

En el esquema de la estructura del controlador se puede observar los distintos elementos que lo forman, la anchura de los buses y las señales que afectan a dichos elementos. Hay que destacar en el controlador la microprogramación vertical implementada, que permite codificar todas las microoperaciones que se ejecutan con 16 bits de la micropalabra; así como la posibilidad de realizar saltos dentro de la CROM, para lo que se dispone de un registro de 8 bits (*C*) y de señales para su manejo, lo que posibilita la programación de bucles enter las micropalabras en una sola instrucción.

Una gran ventaja que presenta es la posibilidad de implementar un repertorio completo de instrucciones (hasta 32 instrucciones) a partir de las microoperaciones que se ofrecen y definir programas que utilicen esos repertorios.

### 4 Servidor de Prácticas (SEPRAC)

El Servidor de Prácticas actúa como un demonio de red que se ejecuta en segundo plano y lo único que requiere es una máquina virtual JAVA ejecutándose en el servidor. El servidor tendrá un usuario especial que actuará como administrador, estableciendo los parámetros de funcionamiento, cuotas de disco, etc. El servicio proporcionado por el servidor de prácticas es permitir a los alumnos la carga o almacenamiento de prácticas desde/en un host remoto. Existe una página web en la que el usuario del simulador ha de rellenar y remitir un formulario web para que el administrador del servidor SEPRAC pueda darlo de alta. Es tarea del administrador dar de baja a los usuarios (alumnos) cuando éstos dejen de estar asociados a la asignatura

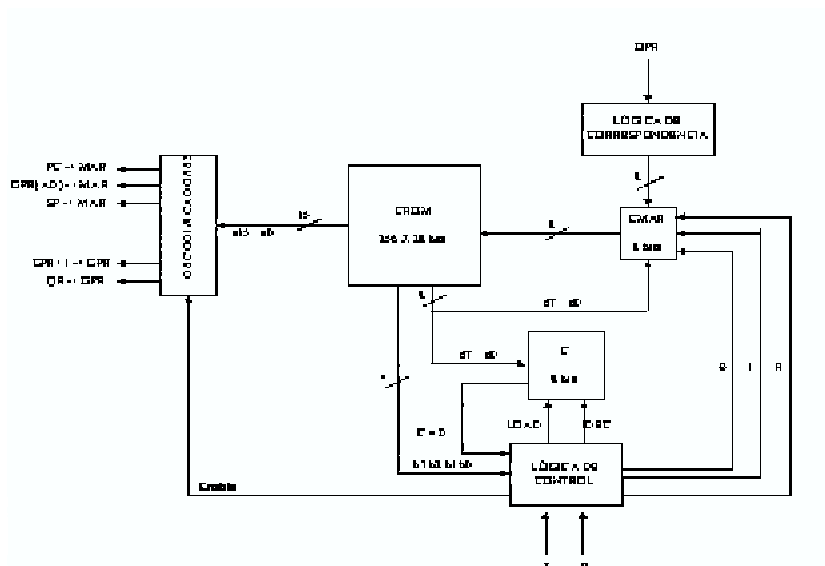


Figura 5: Arquitectura de la Computadora Mejorada del SICOME

(al finalizar el curso).

El protocolo de funcionamiento visto desde el punto de vista de los usuarios es el siguiente: el usuario introduce su login y su clave de acceso, se autentifica en el servidor y en caso correcto, se realiza la conexión. Una vez que la conexión se establece, todas las transmisiones y transferencias de ficheros se realizan de manera encriptada.

Para poder utilizar este mecanismo, es necesario modificar ligeramente los simuladores, de manera que sean capaces de realizar la conexión al servidor y realizar las transferencias de los ficheros de las prácticas. Para ello, se ha añadido una entrada en el menú (*Servidor*) que permite acceder a los servicios del servidor. En caso de no querer acceder al servidor, sino trabajar solo en modo local, no utilizaremos las entradas de dicho submenú.

Como se puede observar este servidor es independiente de los clientes, lo único que realiza es la autenticación de los usuarios y la gestión de las transferencias de ficheros servidor - cliente. Esto hace que el sistema sea muy versátil, pudiéndose ampliar su uso fácilmente, teniendo sólo que modificar los simuladores clientes, añadiéndoles un submenú de acceso a los servicios del servidor.

Otra ventaja que tiene este sistema centralizado, es que unido a un servidor Web, presenta una gran facilidad para actualizar, modificar o añadir simuladores, presentando un sistema de descargas de los ficheros de los mismos a través de páginas web.

Figura 6: Pantalla principal del Servidor de Prácticas SEPRAC



## 5 Salas Virtuales en Arquitecturas de Computadoras (SAVIAC)

Mediante SAVIAC se ha abierto un nuevo camino en el apoyo a la educación dentro de la Universidad de Córdoba. Se puede decir que si con el conjunto de simuladores, los alumnos podían aplicar de manera práctica los conocimientos teóricos adquiridos y mediante la utilización del servidor de prácticas los alumnos podían tener acceso a los simuladores en cualquier momento y desde cualquier lugar, pudiendo entregar las prácticas desde casa. Sin embargo, todas estas mejoras adolecen de un gran fallo, el distanciamiento del alumno de su entorno: compañeros, asignatura y docente. Es decir, al utilizar estos mecanismos el alumno se aleja de la rutina docente habitual, lo cual también tiene sus desventajas. Sin embargo, se pueden paliar estos efectos negativos al incluir mecanismos que relacionen al alumno con el resto del entorno docente: compañeros, asignaturas y docentes.

Mediante este sistema el alumno adquiere una enseñanza mucho más integral que en los sistemas previos, puesto que este sistema no es excluyente de los anteriores sino que se puede tomar como un añadido a los anteriores que los complementa: cada alumno puede realizar las prácticas de las asignaturas del área desde cualquier lugar (como en los casos anteriores), comentar con sus compañeros y/o profesores y asistir a las tutorías del docente a cualquier hora y desde cualquier lugar que posea acceso a Internet, realizando un gran aprovechamiento de los recursos docentes del departamento. La utilidad es más clara en el período de exámenes, al automatizarse la entrega de prácticas, sin la necesidad de que el alumno se tenga que trasladar físicamente a los laboratorios del departamento tanto para la realización como para la entrega de las prácticas. De esta manera, alumnos cuyos hogares están lejos de la ciudad, pueden volver a sus domicilios, manteniendo contacto tanto con la asignatura como el resto de compañeros, consultar al profesor, asistir a las tutorías virtuales que el docente establezca, etc.

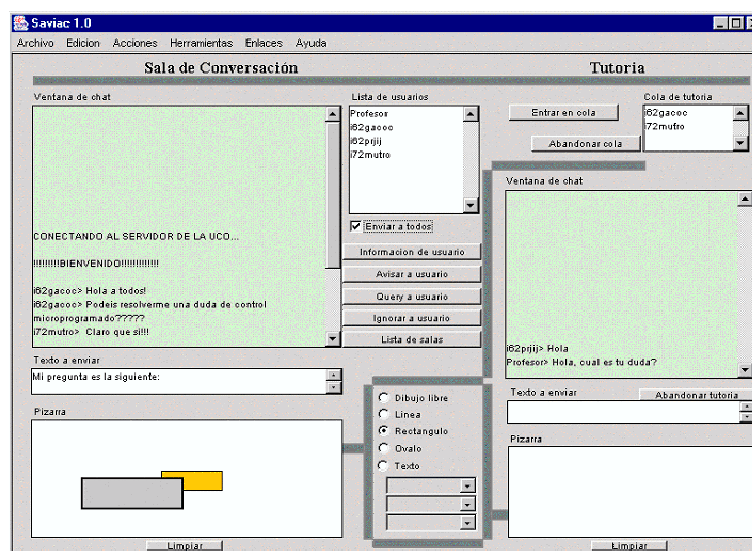


Figura 7: Pantalla principal del servidor de Salas Virtuales SAVIAC

### 5.1 Sala de Conversación

Realmente el sistema presenta una sala virtual centrada en la temática de Arquitectura de Computadores, que pretende eliminar barreras tanto geográficas con la sala de conversación y con las tutorías como temporales con los foros temáticos. Estos últimos permiten a los usuarios conocer las conversaciones que se han mantenido y las dudas que se han resuelto independientemente del momento en el que entren en el sistema. Existen dos perfiles de usuarios: profesor y alumno. Ambos pueden acceder a la sala virtual desde los simuladores o directamente indicando la dirección de Internet donde se encuentra alojado el servidor.

En la barra de menú, hay que destacar el menú *Enlaces*, donde se tiene acceso a páginas relacionadas con la asignatura, apuntes, problemas, etc. así como a los diversos foros temáticos. El resto de la interfaz está dividida en dos partes: ambas con una pizarra virtual para ayudar en algunas explicaciones. La sala de conversación, a la izquierda, se trata de un chat entre todos los alumnos conectados. Las tutorías, a la derecha, es también un chat pero sólo entre profesor y un alumno, que es el que en ese momento asiste a la tutoría. Éste es seleccionado por parte del profesor mediante una cola FIFO en la que el alumno se añade

cuando desea realizar una pregunta al profesor.

El profesor es el único que tiene acceso a la interfaz de administración, que posee dos partes: a la izquierda, donde se manejan las cuentas de los alumnos, se añaden enlaces a la sala, se crean habitaciones privadas, se permite desconectar a alumnos, etc. y la parte de estadísticas, a la derecha, con distintos controles del uso del sistema por alumnos o de manera global. Mediante esta interfaz, el profesor puede limitar el uso de la sala, para evitar situaciones donde un alumno se comporte incorrectamente.

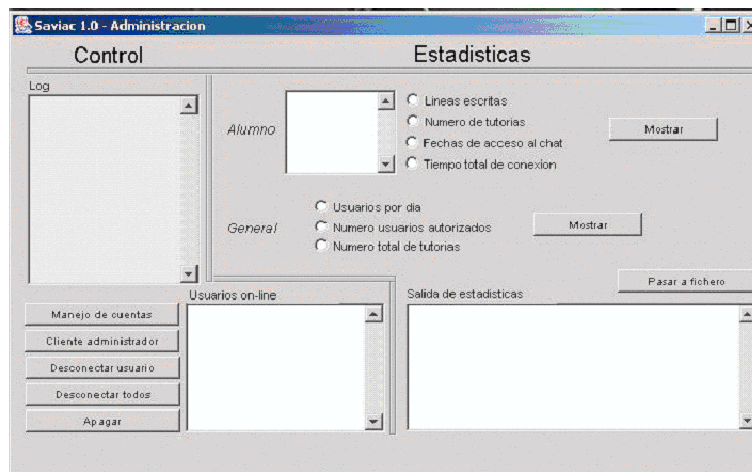


Figura 8: Pantalla de Control y Estadísticas del servidor SAVIAC

Como se ha comentado, los usuarios se pueden conectar directamente desde los simuladores, para lo cual, se ha visto necesario realizar una pequeña modificación en los programas de los simuladores, para darles cabida dentro del servidor de Salas Virtuales. Sin embargo, el funcionamiento interno de cada uno de ellos no se ve afectado, puesto que el acceso a las salas virtuales, al igual que el acceso al servidor de prácticas es un servicio dentro del submenú de servicios del servidor.

## 6 Conclusiones

La utilización de simuladores desde hace algunos cursos ha significado un gran avance en la formación global de los alumnos, ya que han podido comprobar de manera práctica los conceptos de teoría. Más aún, el hecho de que los simuladores hayan sido realizados por miembros del área de Arquitectura y Tecnología de Computadoras de la Universidad de Córdoba (y que, por tanto, se posea el código fuente de los mismos) ha permitido ir realizando mejoras que han llegado a la creación de un sistema-entorno docente de gran aceptación y versatilidad que permite a los alumnos realizar, entregar las prácticas, hacer preguntas y comentar con compañeros y profesores sobre las asignaturas, todo ello sin necesidad de moverse físicamente a los ordenadores de la universidad, pudiéndolo hacer desde su propia casa. El sistema está teniendo una gran aceptación entre el alumnado y se está planteando ampliar su uso a más asignaturas del área docente. Se muestra una pequeña tabla donde se pueden observar los resultados de utilización de los diversos simuladores utilizados en los 3 últimos cursos.

Tabla 1: Uso de los diversos simuladores en asignaturas del Área

Simulador	Alumnos	Aprobados	Aprobados(%)
SICAM	61	59	96.72 %
XCACHE(*)	50	43	86 %
MIPSIM(*)	181	162	89.50 %
SICOME	810	572	70.62 %

(\*) Simuladores no realizados dentro del área ATC de la Universidad de Córdoba.

Se ha podido observar un alto grado de aprobados en las materias que se han impartido, ahora bien, no se puede comparar con otros simuladores, en la mayoría de los casos, puesto que no se han usado otros (salvo en el caso de XCACHE, que se utilizó con anterioridad a la utilización del simulador SICAM). Ahora bien, los resultados son muy interesantes puesto que en las prácticas que se usan estos simuladores el índice

de aprobados es esperanzadoramente elevado. No podemos dar datos sobre la utilización de los entornos de integración de los simuladores (SEPRAC y SEVIAC) ya que han sido probados por un pequeño grupo de alumnos, para comprobar su grado de adaptación, obteniéndose comentarios muy positivos por parte de dicho alumnado, por lo que durante el presente curso 2002-2003 se pasará a utilizar por todo el alumnado de las asignaturas involucradas.

Por parte del profesorado, el sistema tiene grandes ventajas por la facilidad a la hora de estructurar la entrega de prácticas y atender a alumnos en tutorías virtuales, teniendo todo el sistema centralizado. Ahora bien, requiere por parte del profesorado un cierto grado de entrega y de conocimientos sobre Internet para poder ir incrementando el entorno docente en Internet asociado a cada asignatura.

## Referencias

- [1] Taub, H. *Circuitos Digitales y Microprocesadores*. McGraw-Hill. 1983
- [2] Morris Mano, M. *Arquitectura de Computadores*. Prentice-Hall Hispanoamericana S.A. 1983.
- [3] Mesa, J.C. and Cobos, A. and Sáez, E. and Benavides, J.I. *Simulación de la Arquitectura de una Computadora Sencilla con fines docentes*. PFC - Depto. Elect. y Electrónica. Universidad de Córdoba. 1999.
- [4] Conde, S. and Muñoz, A. and Herruzo, E. and Benavides, J.I. *Aplicación interactiva de simulación del comportamiento de una Computadora Mejorada con fines docentes*. PFC - Depto. Elect. y Electrónica. Universidad de Córdoba. 2001.
- [5] Conde, S. and Muñoz, A. and Herruzo, E. and Benavides, J.I. *Servidor de Internet de prácticas docentes (CD-ROM)*. Universidad de Córdoba. ISBN: 8493229911. 2001.
- [6] Conde, S. and Muñoz, A. and Herruzo, E. and Benavides, J.I. *Herramienta de Simulación de la Computadora Mejorada con Fines Docentes*. IX Congreso Universitario de Innovación Educativa en las Enseñanzas Técnicas. Vigo. 2001.
- [7] Grunbacher, H. *MIPSim*. Institut für Technische Informatik. Vienna University of Technology. 1996.
- [8] Grunbacher, H. *MIPSim*. Institut für Technische Informatik. Vienna University of Technology. 1996.
- [9] Pressman, R.S. *Ingeniería del Software. Un enfoque práctico*. 3ª Edición. McGraw-Hill/Interamericana de España S.A. 1994.
- [10] Patterson, D. and Hennessy, J.L. *Organización y Diseño de Computadores. La interfaz hardware/software*. 2ª Edición. McGraw-Hill/Interamericana de España S.A. 1995.
- [11] Stallings, W. *Organización y Arquitectura de Computadores. Diseño para optimizar prestaciones*. 5ª Edición. Prentice-Hall Iberia. 2000.
- [12] Herruzo, E. and Benavides, J.I. *Servidor de prácticas docentes y sala de conversación temática en Internet*. III Conferencia Internacional sobre Educación, Formación y Nuevas Tecnologías. Valencia (España).
- [13] <http://www.ece.gatech.edu/research/ccss/education/Java/1997.Summer/Projects/alm/prj.html>
- [14] <http://www.cs.wisc.edu/~markhill/DineroIV/>
- [15] <http://palms.ee.princeton.edu/plx/>
- [16] <http://arco.unex.es/jangomez/I+D/sismec/sismec.html>
- [17] <http://atc.unex.es/mavega/SMPCache/>
- [18] <http://www.eecs.umich.edu/~estam/mlcache/>
- [19] <http://research.compaq.com/url/people/jouppi/CACTI.html>