

**AYUDAS DE LA UNIVERSIDAD DE SALAMANCA PARA LA  
INNOVACIÓN DOCENTE PARA EL CURSO 2008/2009**

**MEMORIA FINAL**

**PROYECTO ID/0047**

**Autoaprendizaje Interactivo para la Materia “Sistemas  
Operativos” en el Marco del EEES**

Investigador principal: [Dr. Roberto Therón Sánchez](#)  
Departamento de Informática y Automática  
Universidad de Salamanca - Facultad de Ciencias  
Plaza de la Merced, s/n  
37008 Salamanca

## Tabla de Contenidos

1. DATOS DEL PROYECTO .....	3
2. OBJETIVOS PROPUESTOS .....	4
3. DESCRIPCIÓN DEL DESARROLLO DEL PROYECTO Y RESULTADOS LOGRADOS .....	6
3.1 <i>Introducción</i> .....	6
3.2 <i>Ciclo de vida</i> .....	6
3.3 <i>Especificación de requisitos</i> .....	7
3.4 <i>Análisis y Diseño</i> .....	7
3.5 <i>Descripción del sistema</i> .....	7
3.6 <i>Servidor de sockets</i> .....	8
3.7 <i>Disco</i> .....	9
3.8 <i>Paginación</i> .....	11
3.9 <i>Planificación</i> .....	15
3.10 <i>Portabilidad y Accesibilidad</i> .....	21
3.11 <i>Fase de Explotación</i> .....	21
4. MATERIAL Y GRADO DE CUMPLIMIENTO DEL PROYECTO INICIAL .....	23
4.1 <i>Material adquirido</i> .....	23
4.2 <i>Grado de cumplimiento</i> .....	23
4.3 <i>Publicaciones generadas</i> .....	24
5. CONCLUSIONES.....	25

## 1. DATOS DEL PROYECTO

**TÍTULO:** Autoaprendizaje Interactivo para la Materia “Sistemas Operativos” en el Marco del EEES

**REFERENCIA:** ID/0047

**INVESTIGADOR PRINCIPAL:** THERÓN SÁNCHEZ, ROBERTO

**ORGANISMO:** UNIVERSIDAD DE SALAMANCA

**CENTRO:** FACULTAD DE CIENCIAS

### **INVESTIGADORES QUE FORMAN EL EQUIPO:**

Roberto Therón Sánchez

Luis Antonio Miguel Quintales

Guillermo González Talaván

Ana Belén Gil González

**DURACIÓN:** octubre 2008 a mayo 2009

## 2. OBJETIVOS PROPUESTOS

El objetivo principal de este proyecto es desarrollar un sistema *on-line* audiovisual e interactivo que muestre el funcionamiento de los principales algoritmos de los sistemas operativos en las áreas de acceso a disco, reemplazo de páginas de memoria y distribución de la CPU entre los procesos del sistema.

### Como objetivos generales se presentan:

- 1) Favorecer y apoyar el autoaprendizaje de alumno de las materias de “Sistemas Operativos”, dada la importancia de una correcta asimilación de estos contenidos para poder abordar con éxito el resto de asignaturas de su titulación. El alumno podrá ver e interactuar con las partes que normalmente se le ocultan a un usuario de un sistema operativo.
- 2) Incidir positivamente en el proceso educativo, proporcionando material didáctico más atractivo para el discente. Los alumnos podrán adquirir la capacidad de abstracción imprescindible para comprender los detalles de bajo nivel en el diseño y programación de sistemas operativos.
- 3) Proporcionar un complemento a las clases magistrales en la materia de “Sistemas Operativos”, poniendo en las manos del docente un mini sistema operativo, mediante representaciones visuales abstractas, que permite dar la necesaria visión holista. De esta forma, el alumno puede fácilmente comprender que lo visto teóricamente y de manera aislada para cada uno de los subsistemas, tiene en la práctica su repercusión en el sistema operativo completo.

### Como objetivos específicos cabe citar:

- 1) **Disco:** Debe simularse el funcionamiento de un disco magnético y proporcionar un método para solicitar bloques del mismo. La simulación deber permitir también al usuario escoger el algoritmo que se empleará para atender las peticiones y mostrar claramente el orden en que se van atendiendo las mismas.
- 2) **Paginación:** Se simulará un sistema formado por una memoria física y unos procesos que accedan a dicha memoria. La herramienta debe mostrar claramente cómo se realiza el reemplazo de unas páginas por otras de la memoria, permitiendo al usuario solicitar las páginas que desee para personalizar así la simulación. Al igual que con el disco se podrá seleccionar el algoritmo que se desee estudiar.
- 3) **Planificación:** En este caso deberemos ver cómo el sistema distribuye la CPU entre los procesos del sistema dependiendo del algoritmo seleccionado. Dado que se pretende utilizar un modelo de proceso de 7 estados, habrá que incluir también en el sistema una memoria física para que existan procesos activos (cargados en memoria) y suspendidos (no cargados en memoria). Esto lleva a incluir también en esta misma herramienta un planificador de medio plazo que siga alguna mínima estrategia para decidir qué procesos se deben cargar en memoria y cuáles tienen que ser descargados de la misma.

- 4) Las simulaciones producidas por la herramienta deben ser claras e intuitivas, permitiendo al usuario configurar la mayor cantidad de parámetros posible de manera que éste pueda experimentar y personalizar la simulación acorde a sus necesidades. Para que la experiencia resulte satisfactoria el usuario deberá recibir gráficamente la mayor cantidad de información posible y tener la posibilidad de ver con mayor claridad la información que le resulte más útil.
- 5) Otro de los objetivos que se persigue con el proyecto es el de complementar dos tecnologías en alza, Flash y C#, explotando las principales virtudes de cada una. Para ello dividiremos el sistema en tres partes:
  - a. **Núcleo:** Comprende toda la parte operativa de la herramienta, está formada por todos los componentes empleados para implementar los algoritmos y realizar la simulación. Estará íntegramente desarrollado en C#.
  - b. **Interfaz de usuario:** Serán las páginas web desarrolladas en Flash. Éstas deberán mostrar al usuario la representación gráfica de la simulación que estará teniendo lugar en el núcleo, a la vez que le sirven como canal para configurar los parámetros de la misma.
  - c. **Intermediario:** Será la parte del sistema encargada de comunicar el núcleo con la interfaz de usuario. Dado que no hay una forma específica de conseguir dicha comunicación se buscará un método viable y eficiente para tal efecto.
- 6) Una cualidad importante de la aplicación tiene que ser la de la reusabilidad, ya que si tiene éxito, el sistema podría ser ampliado con nuevas herramientas tanto de esta como de otras materias para formar un sistema de mayor tamaño; por tanto tiene que ser fácilmente ampliable.
- 7) También sería una cualidad muy deseable la portabilidad, ya que sería interesante que pese a ser desarrollada en un entorno Windows la aplicación pudiese portarse también a un entorno Linux, por ser estos los sistemas más utilizados como servidores en la actualidad.
- 8) Al ser una aplicación web debería estar disponible para cualquier usuario con una conexión a Internet y una extensión de Adobe Flash para el navegador Web, independientemente del sistema operativo que el cliente utilice.

**Finalmente, como objetivos docentes se consideran:**

1. Desde el punto de vista docente, el sistema de autoaprendizaje ofrecerá:
  - a) Un sistema centralizado que permite llevar un seguimiento del uso del sistema por parte de todo el alumnado.
  - b) Un sistema que separa la complejidad de implementación de los diversos algoritmos y estructuras de datos que mantiene un sistema operativo, de su representación abstracta orientada a la formación de alumnos.

- c) Un sistema previsto para crecer, en el que se puedan incorporar nuevas formas de representación abstracta que faciliten el aprendizaje de conceptos complejos.
  - d) Un sistema operativo en miniatura que permite proporcionar la versión integradora de los distintos subsistemas, que exhibe las características de bajo nivel sin la necesidad de involucrar al alumno en excesivamente exigentes y largos desarrollos.
2. Desde el punto de vista discente, el sistema de autoaprendizaje:
- a) Acceso a abstracciones visuales adecuadas que faciliten la comprensión tanto de conceptos como de funcionamiento de algoritmos
  - b) La posibilidad de realizar cualquier número de simulaciones con juegos de datos no predefinidos, seleccionados por el alumno.
  - c) La posibilidad de ver de forma integrada la interrelación de los distintos subsistemas, comprobando las repercusiones de determinadas acciones que el propio alumno puede desencadenar.
  - d) La posibilidad de estudiar los algoritmos vistos en las clases magistrales, analizar la repercusión de las variaciones de los parámetros que rigen cada uno de ellos, reproducir las animaciones resultantes de los juegos de datos seleccionados, tanto en modo automático como en modo paso a paso.

### 3. DESCRIPCIÓN DEL DESARROLLO DEL PROYECTO Y RESULTADOS LOGRADOS

#### 3.1 Introducción

En este apartado se recogen los aspectos más notables del proyecto. Al tratarse de un desarrollo informático, en esta memoria se incluyen tanto aspectos más técnicos propios del software desarrollado (documentación y descripción del sistema que se ha llevado a cabo), como los tocantes a los objetivos como proyecto de innovación docente. Así, se describen los servicios que ofrece el sistema tutorial basado en la Web. No obstante, si se requiere una discusión más detallada acerca de él puede consultarse el “Manual de usuario” de la aplicación desarrollada.

#### 3.2 Ciclo de vida

Para el desarrollo de este proyecto se optó por seguir una metodología orientada a objetos debido especialmente a las características de escalabilidad, extensibilidad y reutilización; **estas características son idóneas para la**

**continuidad del tutorial en el futuro, posiblemente a través de nuevos proyectos de innovación.** Entre las distintas posibilidades existentes, finalmente se decidió realizar este proyecto basándose en el Proceso Unificado, ya que es uno de los procesos de desarrollo software que más se utiliza actualmente y uno de los que mejor se adapta a cualquier tipo de situación y contexto.

A continuación se especificarán las técnicas utilizadas y los aspectos más importantes de la especificación de **requisitos, el análisis, el diseño y la implementación tenidos en cuenta a la hora de desarrollar un sistema de autoaprendizaje robusto y preparado para ampliarse y evolucionar al recoger la realimentación de los alumnos.**

### 3.3 Especificación de requisitos

**La obtención de requisitos se ha basado en la experiencia,** acumulada a lo largo de numerosos años impartiendo esta asignatura de forma presencial por los miembros proyecto.

### 3.4 Análisis y Diseño

En el análisis y el diseño de este proyecto se ha utilizado el Lenguaje de Modelado Unificado (UML). Se han creado diversos diagramas que incluyen diagramas de secuencia para diseñar la vista dinámica y los diagramas de clases para diseñar la vista estática. **Todo esto ha generado gran cantidad de documentación técnica que se legará a futuros desarrolladores del tutorial Web.**

### 3.5 Descripción del sistema

Podríamos dividir la aplicación en tres partes, la Figura 1 muestra un esquema del funcionamiento del sistema:

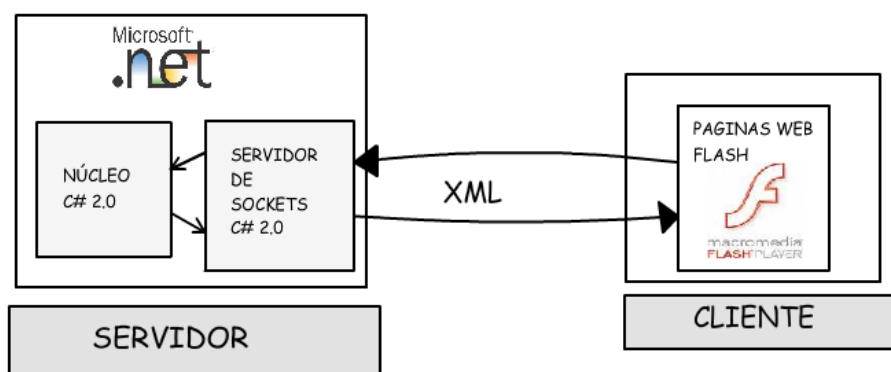


Figura 1.- Esquema de funcionamiento del sistema

- **NÚCLEO**

Realizado en C# 2.0, es el conjunto de clases que implementan los algoritmos de la aplicación. Todos los cálculos de la simulación se realizan

en esta parte, que envía los resultados mediante mensajes XML a las páginas cliente a través del servidor de *sockets*. Es por tanto aquí dónde realmente se está produciendo la simulación, sólo que de manera interna, es decir, sin interfaz gráfica.

- **SERVIDOR DE SOCKETS**

Es una aplicación de consola realizada en C# 2.0 bajo la cual se ejecuta la parte del núcleo. Su función es la de servir los *sockets* XML que conecten la parte cliente (páginas flash) con la parte núcleo correspondiente (herramientas de disco, paginación y planificación).

Básicamente el proceso es el que sigue: El usuario accede a la página Web, ésta pide una conexión al servidor de *sockets* indicándole el tipo de simulación que precisa (disco, paginación o planificación), si el servidor acepta la solicitud crea un hilo que ejecuta la herramienta correspondiente y le asigna un socket de comunicación con la página. Después el servidor se queda a la espera de nuevas peticiones.

- **PÁGINAS CLIENTE (FLASH)**

Páginas Web realizadas en Flash. Forman la Interfaz de Usuario de la aplicación. Se conectan con el servidor mediante un recurso de flash que son los XMLSocket. La gran mayoría de operaciones se realizan en el núcleo mientras que estas páginas reciben la salida del mismo y le comunican los cambios de preferencias introducidos por el usuario mediante mensajes XML. La enorme capacidad visual de Flash, que posibilita la producción de **una interfaz de usuario visualmente atractiva y clara, que permita la abstracción de los alumnos, a la par que esconder la complejidad de los sistemas operativos a bajo nivel, se convirtió en una parte crucial de la aplicación Web propuesta.**

Por tanto, de acuerdo con **el objetivo, en el marco del Espacio Europeo de Educación Superior, de potenciar el aprendizaje a través de actividades no presenciales, así como el autoaprendizaje**, todo lo necesario para que la aplicación funcione es un servidor Web y la aplicación "servidor de sockets" ejecutándose en la máquina servidor. **Por parte del cliente (la parte que debe usar el alumno) se requiere únicamente un navegador con los plugin de Flash actualizados.**

### 3.6 Servidor de sockets

Como se ha dicho, **en este proyecto de innovación, se ha hecho especial énfasis en que los alumnos puedan aprender por su cuenta, independientemente de dónde se encuentren. Para ello se ha requerido conjugar una potencia visual y de interacción eficiente (Flash), con una capacidad de cálculo y simulación automático (C#).** Como consecuencia, uno



de los principales retos a afrontar para la realización de este proyecto fue el de conectar los componentes Flash de la aplicación con los componentes C#. Las páginas Web con los objetos Flash embebidos debían formar la interfaz gráfica de usuario, pero todas las operaciones relativas a la ejecución de las simulaciones debía estar contenida en los componentes C# de la aplicación. Había que encontrar pues una forma equilibrada de enviar los datos del cliente al servidor de un modo lo suficientemente eficiente como para que la simulación no resultase pesada o poco realista. Por otra parte se tenía que enviar una cantidad de información importante, puesto que en una sola unidad de tiempo virtual se actualizan múltiples valores como pueden ser el estado de los procesos, los bits de las páginas, etc. Por lo tanto el canal de comunicación debía ser fluido y eficiente pese a ser vía Web.

El servidor de *sockets* es una aplicación de consola que admite peticiones de conexión de los clientes. En el momento de conectarse, el cliente envía al servidor de *sockets* un mensaje XML indicando la simulación que quiere ejecutar.

En este caso “Simulación” puede tomar tres valores: **disco**, **paginación**, **planificación**.

Una vez creado, cada objeto cliente se ejecuta como un hilo de ejecución independiente, mientras que el servidor se queda a la espera de nuevas peticiones. Finalmente, se puede añadir que, pensando en el futuro del tutorial, se puso un especial énfasis en el desarrollo de un sistema que permitiera la incorporación de nuevas funcionalidades.

### 3.7 Disco

La planificación de acceso a disco consiste en asignar un orden de atención a las peticiones de lectura/escritura de los procesos sobre los bloques del disco. Por lo tanto, la simulación tenía que mostrar al menos los siguientes elementos:

- **Peticiones:** No es necesaria información sobre el proceso que la realiza, ya que las peticiones son meros números de bloque y lo que nos interesa es ver el orden en que se van atendiendo dichas peticiones.
- **Planificador de Disco:** Es el que ejecuta los distintos algoritmos para seleccionar el orden de atención de las peticiones.
- **Disco:** Está compuesto por un conjunto de bloques sobre los que se hacen las peticiones.
- **Sistema Automático de peticiones:** Era necesario un sistema que realizase peticiones aleatorias de forma automática para mejorar la usabilidad de la herramienta.

Con estos elementos, **el sistema está dotado de funcionalidad, pero la vocación didáctica de la herramienta, en la que el aprendizaje se deja en las manos del propio estudiante, hace necesario la propuesta de una solución visual que facilite al alumno la comprensión de los conceptos fundamentales involucrados y un fácil seguimiento de la ejecución de los algoritmos.** Para ello era necesario un elemento gráfico que mostrase el orden de atención de las peticiones con claridad (Figura 2):

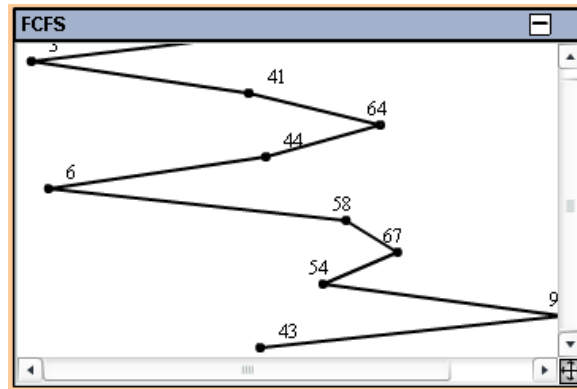


Figura 2.- Representación vectorial de disco

La propuesta había que completarla de una forma que **permita al alumno interactuar con la visualización con el objeto de que ésta se adapte a su propio ritmo de aprendizaje**. Por tanto, había que incluir los controles necesarios para realizar las peticiones, además de mostrar un listado de las peticiones que estaban pendientes de ser atendidas (Figura 3):

Peticiones no atendidas	
9 24 70 88 31 11 72 15 50 0 77 90 84	
Stop	Random OK
	Ultima peticion 84
Peticiones atendidas	
3 41 64 44 6 58 67 54 96 43	
Stop	Recorrido 235

Figura 3.- Representación de peticiones

La experiencia enseñando esta parte del funcionamiento de un sistema operativo recomendó incluir una representación de un disco más realista, para que los usuarios pudiesen asociar con mayor facilidad los movimientos de la cabeza lectora de un disco con las peticiones de bloques (Figura 4):

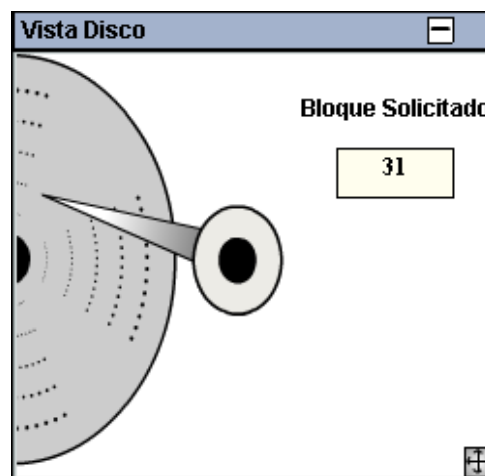


Figura 4.- Representación de disco real

Por último, se consideró necesario incluir un control que permitiese seleccionar el algoritmo deseado para la simulación (Figura 5).

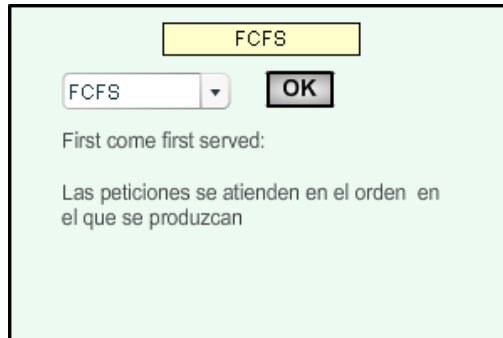


Figura 5.- Control de selección de algoritmo de disco

La Figura 6 muestra el aspecto final de la herramienta:

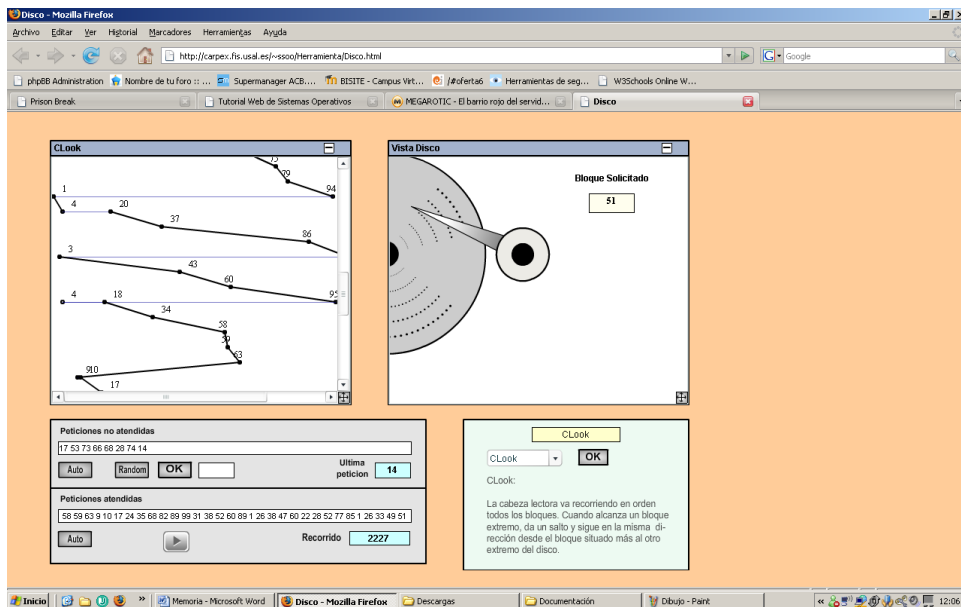


Figura 6.- Herramienta de Disco

### 3.8 Paginación

La herramienta de paginación trata de mostrar el funcionamiento de algunos algoritmos utilizados por los sistemas operativos para decidir qué páginas de las cargadas en memoria física deben ser retiradas cuando hay que hacer sitio para otras nuevas. **Ésta es una de las partes del temario que tradicionalmente más problemas de comprensión plantea a los estudiantes.** Partiendo de los conceptos involucrados, que el alumno debe asimilar, se identifican algunos elementos que debían figurar en la simulación:

- **Páginas:** Contienen información necesaria para la ejecución de los algoritmos, tal como los bits de referencia, validez y modificación, o el PID del proceso al que pertenece.
- **Procesos:** Tienen un espacio de direcciones formado por un conjunto de páginas de memoria. Necesitan cargar sus páginas en la memoria principal

para poder ejecutarse, así que son los que realizan las peticiones para cargar páginas en memoria.

- **Memoria Física:** Está formada por un conjunto de marcos del mismo tamaño que las páginas de los procesos. Es en estos marcos donde se alojan dichas páginas.
- **Planificador de reemplazo:** Es el elemento que debe decidir qué página de la memoria hay que reemplazar cuando queremos cargar una página en la memoria y no hay ningún marco libre.

De forma análoga al caso anterior, una vez implementados estos elementos y puestos en funcionamiento, **es necesario prestar atención al estudiante, facilitarle el autoaprendizaje, por lo que hay que desarrollar una interfaz para mostrar toda la información posible sobre la ejecución de la simulación y para que pueda interactuar con ella, adaptándola a su ritmo de asimilación de conceptos complejos.** El aspecto final de la herramienta es el que muestra la Figura 7.

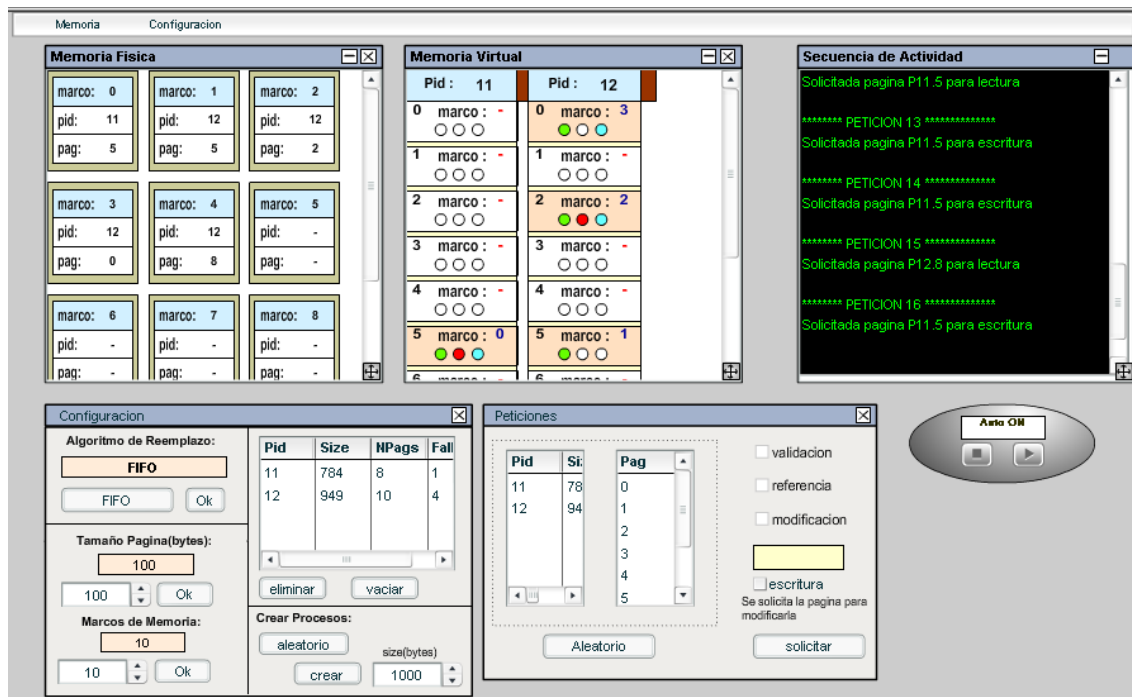


Figura 7.- Herramienta de Paginación

Se debe aclarar que, en primer lugar, los alumnos han demostrado dificultades en la asociación de conceptos vistos en unidades temáticas distintas. Para facilitar esta asociación de conceptos y la asimilación del funcionamiento del sistema operativo como un todo, y no como módulos aislados entre sí, se identificó como imprescindible mostrar el estado de la memoria en cada momento (Figura 8):

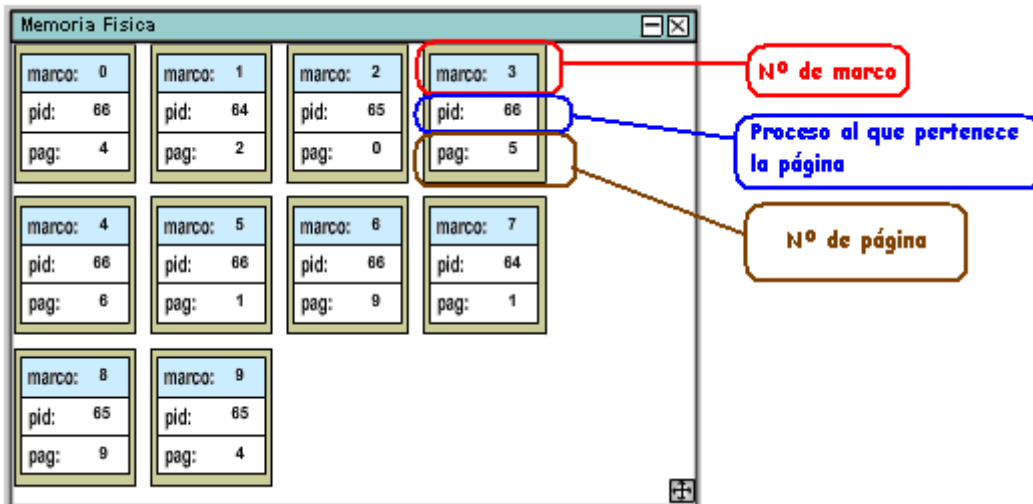


Figura 8.- Ventana de Memoria Física

Lógicamente, si se busca potenciar la capacidad de aprendizaje autónomo, el sistema tutorial necesita mostrar información, de una forma clara y coordinada con el resto de las vistas, sobre la memoria virtual del sistema, es decir, sobre el estado de las páginas de los procesos (Figura 9):

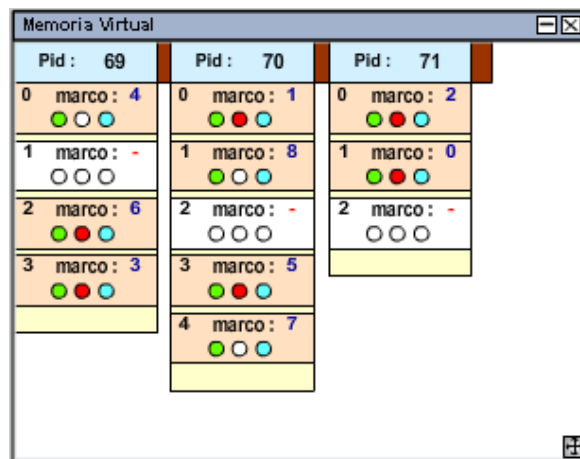


Figura 9.- Ventana de Memoria Virtual

En la Figura 10 se detalla información sobre la memoria virtual:

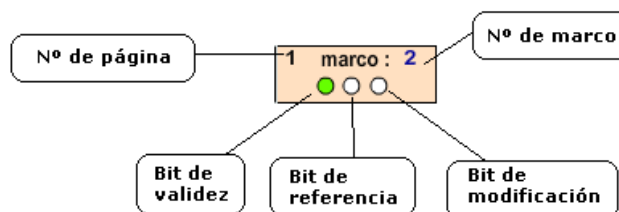


Figura 10. - Información de página de memoria

Para facilitar el seguimiento de la simulación se incluyó una ventana de actividad, basado en un intuitivo sistema de colores, que mostrase un listado de las operaciones que se ejecutan en cada instante de la simulación

(Figura 11):

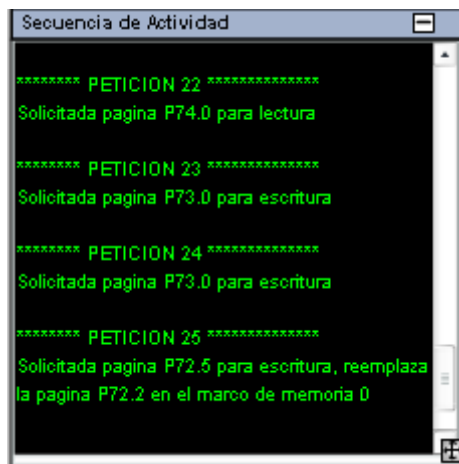


Figura 11.- Ventana de actividad

Un elemento clave que facilita el autoaprendizaje a través de enfoques visuales como el del proyecto, es la interacción, lo que permite al usuario adatar tanto la simulación como la propia vista del estado de la misma. Así, esto se proporcionó a través de la incorporación de unos cuadros de diálogo para que el usuario pudiese configurar la aplicación a su gusto. El cuadro de la Figura 12 nos permite observar una lista de los procesos del sistema y sus características (PID, tamaño, número de páginas y fallos de página producidos hasta el momento), así como crear nuevos procesos y eliminar los ya existentes. En la parte izquierda del control se puede configurar aspectos como el algoritmo, el tamaño de página o el tamaño de la memoria:

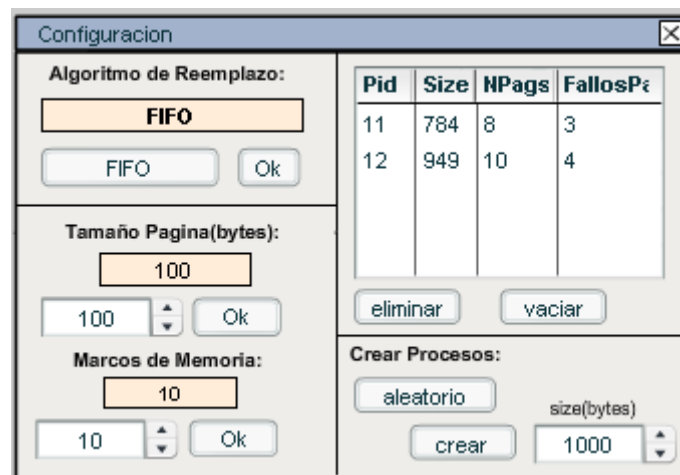


Figura 12.- Configuración de Paginación

El cuadro de la Figura 13 permite realizar peticiones de página. Se pueden realizar peticiones específicas o aleatorias. Al seleccionar una página, se muestra el estado de sus bits en la parte derecha de la ventana.

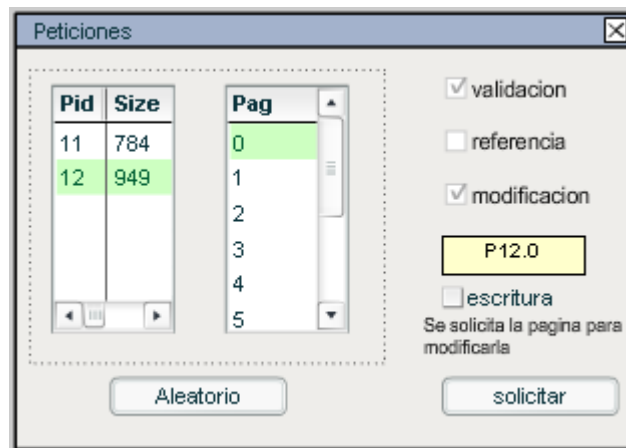


Figura 13.- Ventana de peticiónes

En una primera etapa, el alumno no tiene destreza suficiente para controlar (o parametrizar una situación que desee analizar en detalle), por ello se ha dotado al tutorial de un control que permite poner la simulación en modo automático, de forma que se realizan peticiones de páginas de forma aleatoria y automática (Figura 14).



Figura 14.- Control automático

### 3.9 Planificación

En correspondencia con las dificultades que presentan los alumnos enfrentados a esta parte de los sistemas operativos, el desarrollo de la herramienta de planificación de procesos fue el más elaborado.

Uno de los aspectos cruciales en el aprendizaje de los sistemas operativos es la capacidad de abstracción y de integración de las partes que se estudian por separado. Las visualizaciones propuestas permiten un acercamiento por parte del alumno a un sistema informático abstracto y completo. En consecuencia, además de todo lo comentado, hubo que introducir en la herramienta un nuevo elemento, la memoria física. El hecho de introducir la memoria implicaba también la necesidad de un mecanismo para cargar y descargar las páginas de los procesos en la misma, así que se empleó uno de los algoritmos empleados para la herramienta de paginación, en este caso el algoritmo FIFO.

A modo de ejemplo de la **evolución (y la complejidad de desarrollo que conlleva) que ha sufrido la abstracción de un sistema operativo que se ofrece a los alumnos en el sistema tutorial propuesto**, se muestra la Figura 15.

En este punto lo que teníamos era un sistema compuesto por los procesos, un planificador de corto plazo, una memoria física y un planificador de reemplazo de páginas (que ejecutaba un único algoritmo, el FIFO). Teníamos pues un sistema realista que podía hacer una planificación de procesos de corto plazo y con capacidad para cargar y quitar páginas de los mismos en memoria. Con todo

esto seguíamos con el problema de la suspensión, ya que aún necesitábamos algún componente que decidiera qué procesos debían estar en memoria y cuáles debían ser extraídos de la misma. Se introdujo entonces un nuevo elemento en la simulación, el planificador de medio plazo. Existen numerosas estrategias que este planificador puede utilizar para decidir qué procesos deben ser suspendidos, pero como el objetivo inicial era realizar simplemente una simulación de los algoritmos de corto plazo, en lugar de implementar todas las estrategias posibles se eligió sólo una de ellas, dejando el sistema preparado para la posible inclusión de más estrategias de forma sencilla en el futuro.

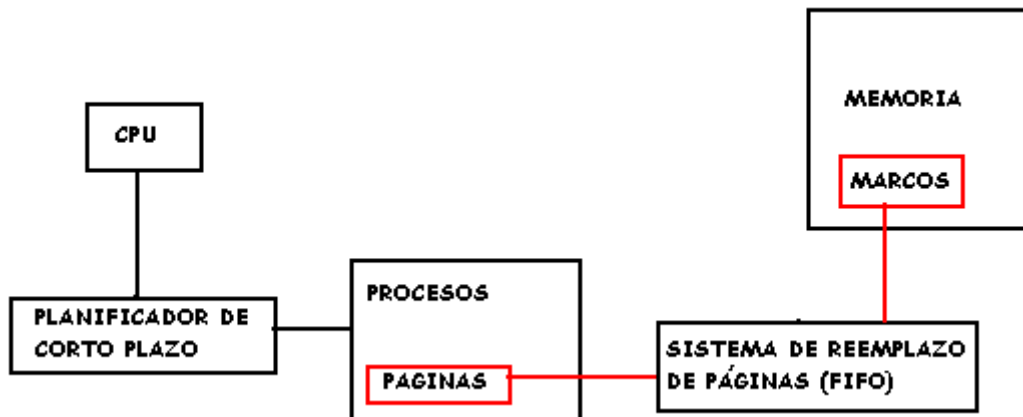


Figura 15.- Esquema de estado parcial del sistema de planificación

Implementar un algoritmo para la planificación de medio plazo, entramos en la teoría de los conjuntos residentes. El desarrollo detallado de este punto del proyecto excede el alcance de esta memoria. **Sólo se debe destacar que siempre se ha tenido en cuenta en las soluciones propuestas el dejar abierta una puerta a futuras mejoras (y que fueran sencillas de introducir).**

Como elemento final, y de cara a una futura ampliación se incluyó un planificador de largo plazo, cuya finalidad es la de decidir qué nuevos procesos son admitidos en el sistema. En resumen, la simulación final quedó compuesta por los siguientes elementos:

- Los procesos
- La memoria física
- El planificador de reemplazo de páginas (con el algoritmo FIFO)
- El planificador de corto plazo (con los algoritmos correspondientes)
- El planificador de medio plazo
- El planificador de largo plazo
- El gestor de conjuntos residentes. (estrategia de Denning)

En la Figura 16 se ilustra un esquema de los componentes mencionados:



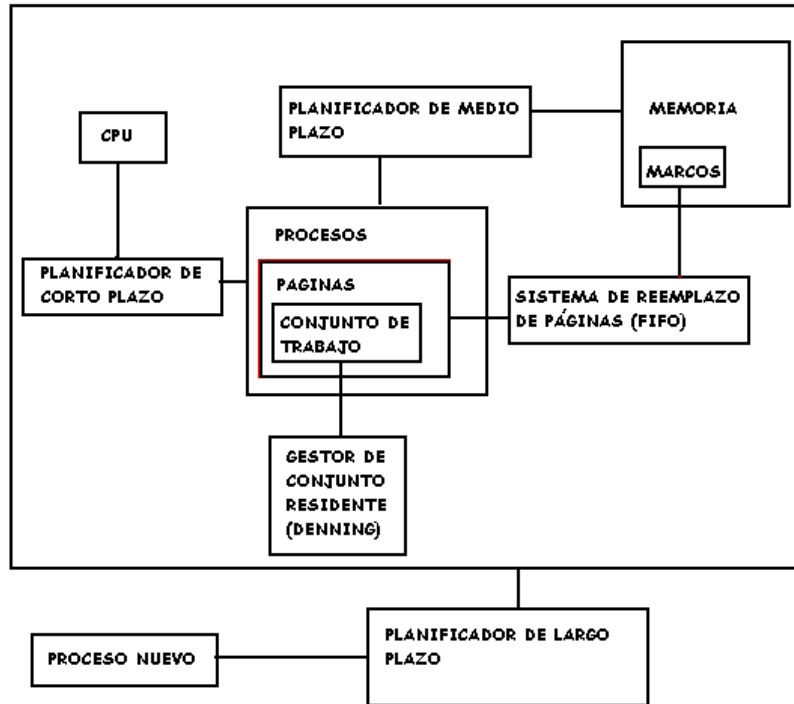


Figura 16.- Esquema final de componentes de planificación

La interfaz de usuario de esta parte de la aplicación reutiliza los elementos de memoria física y memoria virtual de la herramienta de paginación. Aún con eso era necesario añadir nuevos elementos para recoger la información que permitiese seguir la ejecución de los algoritmos. **La ventana de actividad, aunque también es un elemento de la interfaz de paginación, en este caso muestra al alumno otra información diferente (Figura 17):**

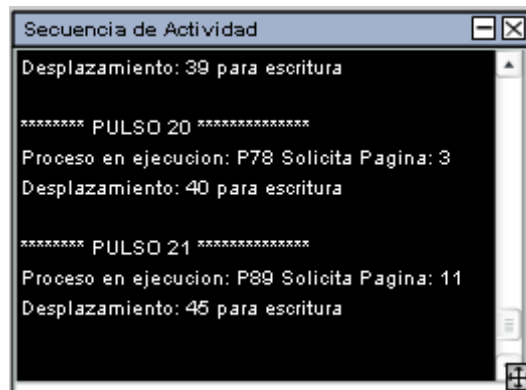


Figura 17.- Ventana de actividad de planificación

Sin embargo, **de forma coherente con el resto de la herramienta, y consecuente con el objetivo general propuesto de proporcionar al alumno una herramienta intuitiva, visualmente atractiva y altamente interactiva,** había que introducir un elemento gráfico que permitiese seguir clara y fácilmente la secuencia de ejecución y aparición de los procesos en la simulación (Figura 18):

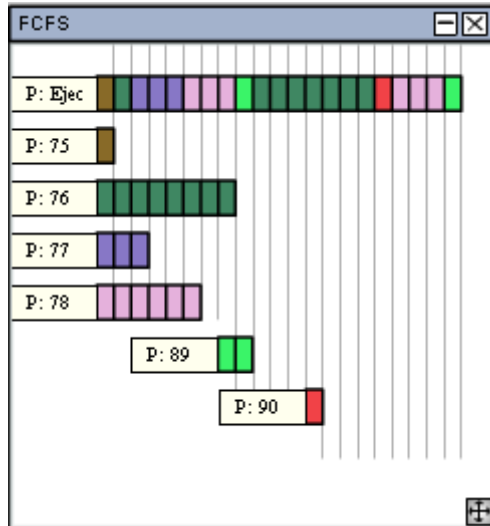


Figura 18.- Ventana de representación de procesos

En esta vista se puede seguir mediante colores la secuencia de ejecución representada mediante la primera barra. También se observa el instante de llegada de los procesos y la duración de los mismos en la barra asignada a cada uno.

Era necesario, además, un elemento que permitiese seguir toda la información relativa a los procesos (PID, estado, tamaño, número de fallos de página, probabilidad de bloqueo) y además permitiese crear otros nuevos (Figura 19). **De esta forma el alumno es el que proporciona los casos de estudio que está interesado analizar..**

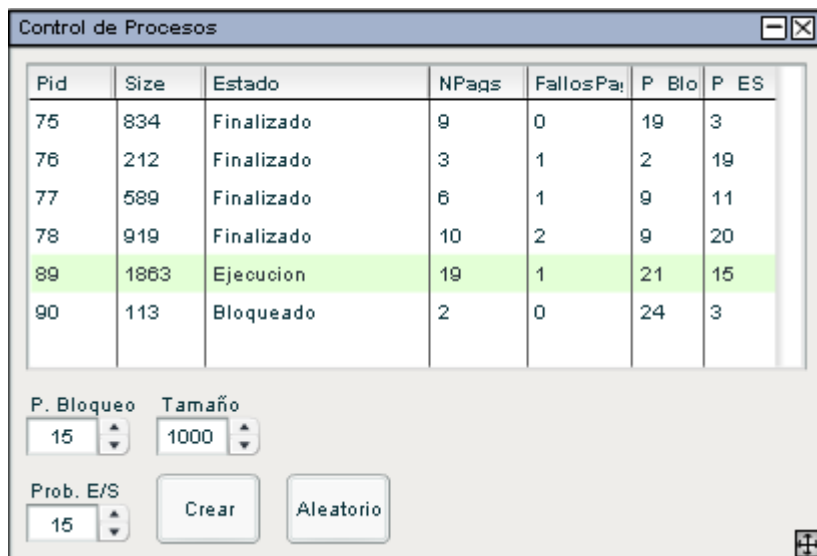


Figura 19.- Control de procesos de planificación

**El éxito de la propuesta, como se ha mencionado en numerosas ocasiones radica en la interacción del alumno con la herramienta proporcionada.**

A continuación se presentan algunas de estas posibilidades: en primer lugar, faltaba incluir los controles para cambiar los parámetros de la simulación. Como en este caso había un gran número de características se utilizó un sistema de

pestañas en un único cuadro de diálogo. La primera permite seleccionar el algoritmo y las características propias del mismo, además de mostrar una descripción (Figura 20)

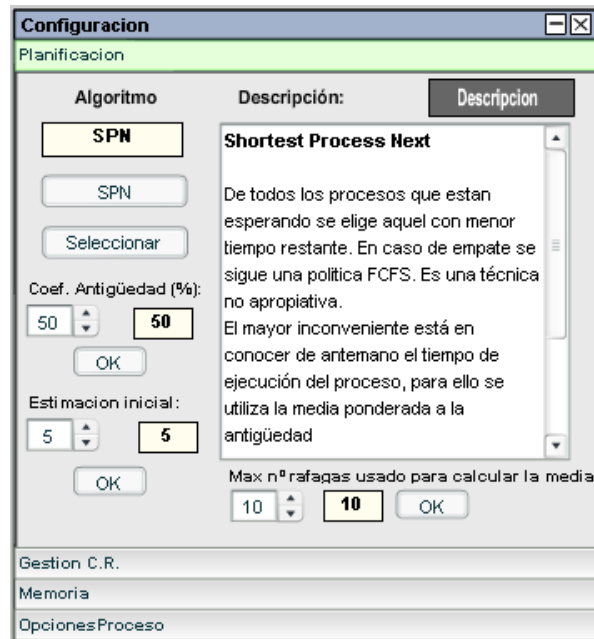


Figura 20.- Pestaña de selección de algoritmo

La pestaña de Gestión del Conjunto Residente (Figura 21) nos permite configurar los parámetros de la estrategia de gestión del conjunto residente y además muestra un listado que nos permite hacer un seguimiento del conjunto de trabajo de cada proceso.

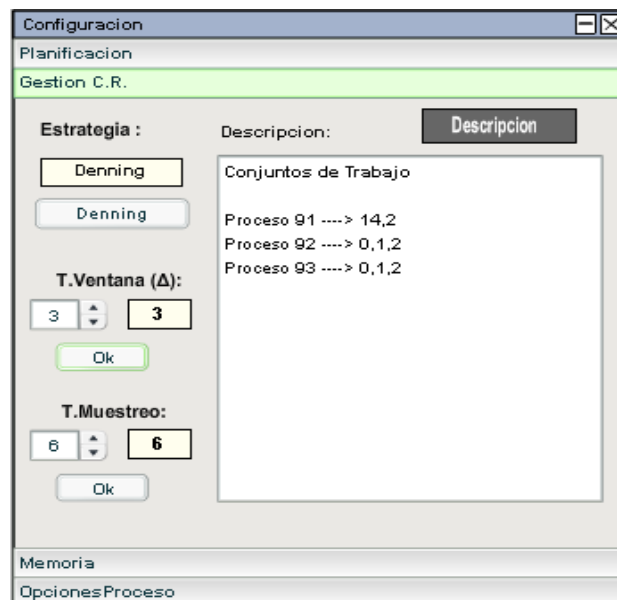
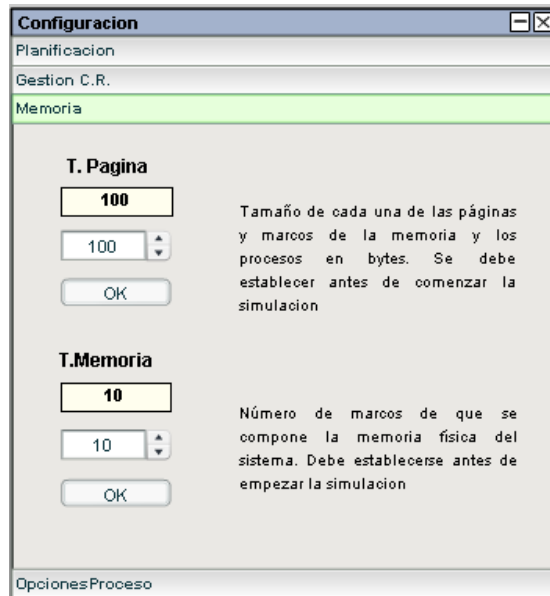


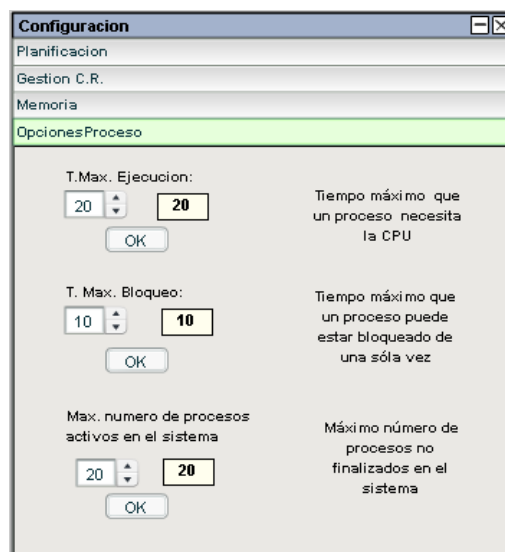
Figura 21.- Pestaña del conjunto residente

La pestaña de configuración de memoria (Figura 22) permite cambiar parámetros relacionados con la parte de paginación de la simulación, como el tamaño de página y el tamaño de la memoria principal.



**Figura 22.- Pestaña de configuración de memoria**

Finalmente la pestaña “Opciones de proceso” (Figura 23) permite configurar ciertos parámetros relacionados con los procesos, como la duración máxima de una ráfaga de CPU, o el máximo número de procesos activos en el sistema.



**Figura 25.- Pestaña de opciones de proceso**

El aspecto final de la herramienta se ilustra en la Figura 23.

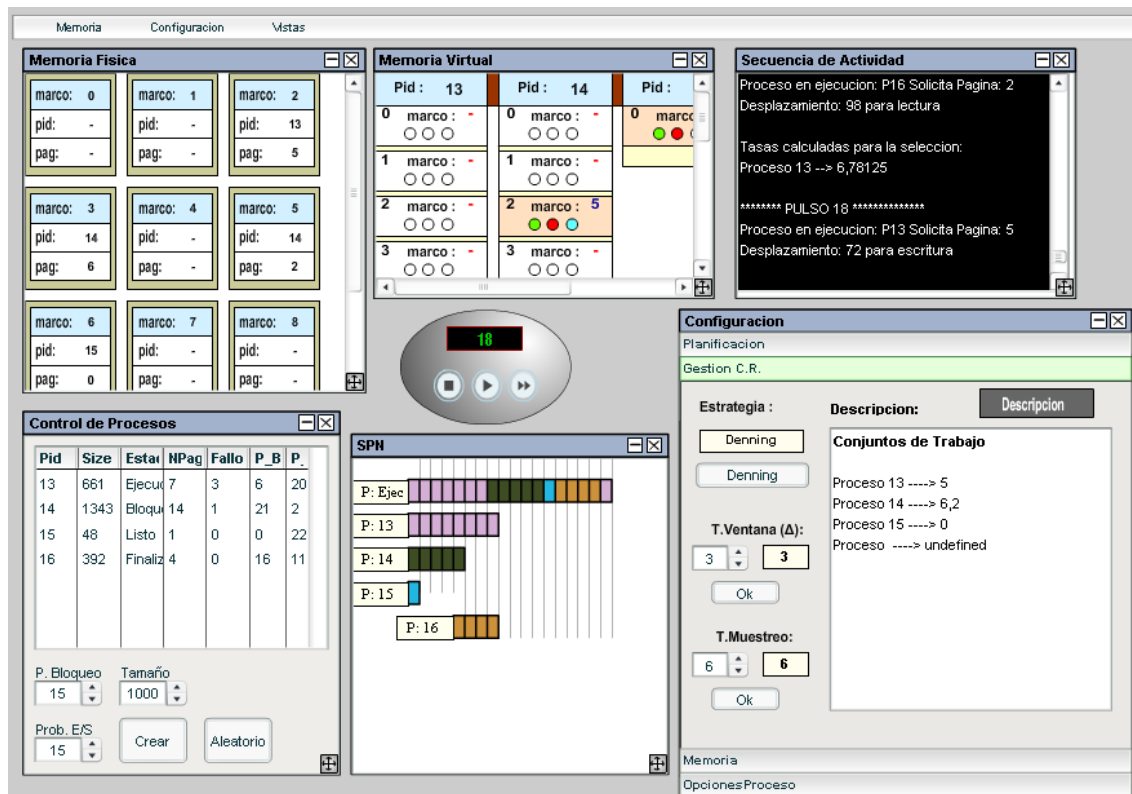


Figura 23.- Herramienta de planificación

### 3.10 Portabilidad y Accesibilidad

En este punto es pertinente recordar que **si queremos facilitar el autoaprendizaje del estudiante, debemos huir de la imposición de determinados soportes tecnológicos. También se debe favorecer la deslocalización de los recursos, haciéndolos accesibles fuera de las facultades o laboratorios.**

En el lado servidor (el controlado por el Departamento de Informática y Automática), la aplicación ha sido desarrollada con el Visual Studio .NET en un sistema Windows XP Profesional. Bajo este entorno no hay ningún inconveniente para ejecutar la herramienta, puesto que todo lo que se necesita es tener funcionando el servidor de *sockets* (aplicación de consola C#) y un servidor de páginas Web.

En el lado cliente, **el alumno no necesita más que un navegador convencional, actualizado con el *plugin* Flash Player (gratuito).**

### 3.11 Fase de Explotación

Todo lo anterior no tendría sentido, si el tutorial no se pusiera a disposición de los alumnos. Por tanto se habilitó un sitio web para que los usuarios pudiesen acceder a la aplicación y utilizarla como herramienta de apoyo de la materia de Sistemas Operativos. El sitio en cuestión es:

**<http://carpex.fis.usal.es/~ssoo/>**.

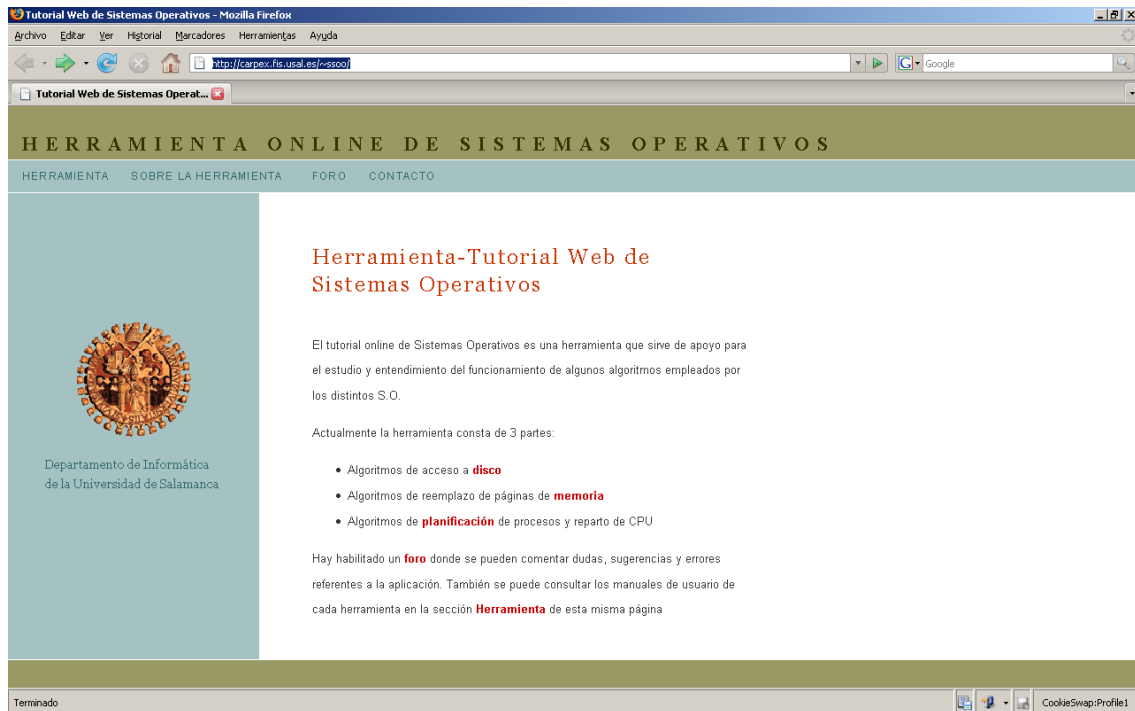


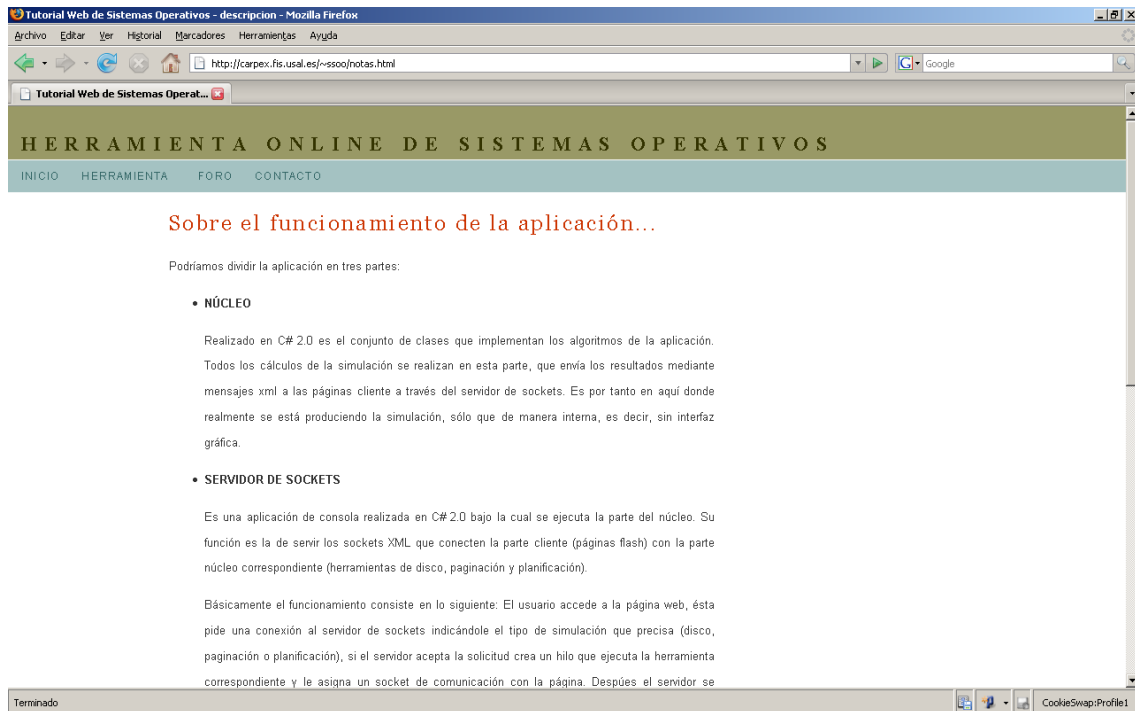
Figura 24.- Sitio web de la aplicación

Desde el sitio Web se permite acceder tanto a las herramientas de simulación como a los manuales de usuario de cada una (Figura 25).



Figura 25.- Acceso a herramienta y manuales

Se ha incluido también una sección que describe brevemente la herramienta para que los usuarios puedan hacerse una idea del funcionamiento de la misma (Figura 26).



**Figura 26.- Pantalla descripción del sistema**

## 4. MATERIAL Y GRADO DE CUMPLIMIENTO DEL PROYECTO INICIAL

Se debe comenzar exponiendo el contexto de ejecución del proyecto. El plazo de desarrollo del proyecto solicitado no cubre una anualidad (octubre-mayo), con lo que el tiempo disponible para realizar los objetivos propuestos es notablemente reducido. Además, el presupuesto es limitado para el desarrollo de un sistema informático completo. En consecuencia, el presupuesto ha sido empleado íntegramente en la compra de un equipo de desarrollo y validación del sistema propuesto.

### 4.1 Material adquirido

Se ha comprado un ordenador portátil y un disco duro externo, que han servido para el desarrollo tanto de la parte Servidor (C#) como de la parte cliente (Flash).

### 4.2 Grado de cumplimiento

Durante toda la duración del proyecto se ha experimentado en el desarrollo de nuevas herramientas de visualización interactivo que faciliten el aprendizaje. Algunas de estas exitosas experiencias han sido ya incluidas en el sistema tutorial Web y se pretende incluir el resto en desarrollos futuros. En este sentido, los resultados son excelentes.

En suma se puede estimar que, a pesar de las limitaciones temporales, se ha cubierto prácticamente todos los objetivos del proyecto, y se estima que, una vez finalizado el proyecto se empiecen a abordar objetivos que no estaban inicialmente previstos para la totalidad del proyecto.

Hay que destacar que el servidor está funcionando y está siendo utilizado por

el equipo de investigación, como apoyo para sus clases, aunque es necesario todavía un periodo de uso continuado y con carga real, para dar por completamente validado el sistema desarrollado.

Por otro lado ha sido diseñado e implementado con vistas de futuro, para que la incorporación de nuevas técnicas de simulación y visualización sea un proceso rápido y exento de complejidades.

Finalmente, **el equipo de investigación está especialmente satisfecho con la labor de innovación en la propuesta de nuevas soluciones visuales para el autoaprendizaje de la materia Sistemas Operativos.** En este sentido se han obtenido resultados que van más allá de los objetivos previstos inicialmente y que permiten asegurar un fructífera línea de investigación que merece la pena ser explorada en sucesivos proyectos de ámbito tanto regional como nacional, e incluso internacional.

#### 4.3 Publicaciones generadas

Diego A Gómez-Aguilar, R Therón, Francisco J García-Peñalvo (2009) *Semantic spiral timeline as a support for e-learning* ***Journal of universal Computer Science*** (IN PRESS)

E M Morales Morgado, D A Gómez Aguilar, F J García-a Peñalvo, R Therón (2009) *Supporting the Quality of Learning Objects Through Their Ranking Visualization* ***International Journal of Emerging Technologies in Learning (iJET)*** 4: 0.



## 5. CONCLUSIONES

El trabajo correspondiente al proyecto consistía en sentar las bases para poder desarrollar completamente el entorno interactivo de autoaprendizaje dentro del marco del Espacio Europeo de Educación Superior, para la materia Sistemas Operativos.. Por otro lado, se coordinó entre los distintos docentes que tienen asignada docencia en esta materia, toda la información que ha permitido el despliegue de un sistema de simulación de un sistema operativo y de un tutorial Web interactivo que potencia el autoaprendizaje.

La revisión de los sistemas internacionales especializados en el área, así como las entrevistas mantenidas con los alumnos, entre los miembros del equipo y con compañeros de otras universidades, ha permitido que se disponga del modelo de datos necesario para sustentar el sistema.

Las experiencias obtenidas en cuanto a la innovación de técnicas de autoaprendizaje interactivo han sido excelentes y la integración de algunas de ellas en el sistema a desarrollar ha sido correcta.

Los resultados obtenidos tras duración de este proyecto han permitido colocar al equipo investigador a la vanguardia de la docencia en Sistemas Operativos. En concreto, el entorno de autoaprendizaje visual desarrollado es uno de los primeros de su clase y en gran medida, sus resultados son transportables a otras materias.

Todo lo anterior permite concluir que **la línea abierta con este proyecto de innovación docente tiene excelentes perspectivas de futuro, y que previsiblemente seguirá produciendo resultados relevantes para la comunidad universitaria.**

Finalmente, se puede añadir que, a la vista de los resultados obtenidos y a todos los efectos, **el proyecto ha sido desarrollado satisfactoriamente.**