Actas de las Jenui, vol. 7. 2022.
Páginas: 39-46

# Simuladores para la docencia de algoritmos de asignaturas de sistemas operativos

Adelaida Delgado Domínguez
Departament de Matemàtiques i Informàtica
Universitat de les Illes Balears
Palma de Mallorca
adelaida.delgado@uib.es

Jaume Aloy Vich
Universitat de les Illes Balears
Palma de Mallorca
jaumealoy@protonmail.com

## Resumen

En las asignaturas de Sistemas Operativos existen gran variedad de algoritmos clásicos, como los de planificación de disco, los de planificación del procesador, los de asignación de particiones de memoria o los de reemplazo de páginas, que pueden ser comprendidos mucho más fácil y amigablemente mediante el uso de simuladores. Tras explorar los existentes, se constataron una serie de deficiencias o carencias que no los hacían lo suficientemente útiles en la actualidad. Por ello se decidió implementar diversos simuladores, con una interfaz homogénea, agrupados en un portal web. En cada simulador se permite configurar los parámetros de los diversos algoritmos disponibles y visualizar su ejecución paso a paso. Se obtuvo así una aplicación web multiplataforma que dispone, además, de ejemplos ya almacenados para facilitar su uso, y tutoriales que guían al usuario en su experiencia con cada uno de los simuladores. Tal recurso docente puede ser de utilidad tanto a profesores para amenizar y enriquecer sus explicaciones (especialmente cuando se realizan por videoconferencia) como para los alumnos, que pueden usarlo de forma activa en su proceso de aprendizaje autónomo, dentro o fuera del aula.

## Abstract

You can find in the courses on Operating Systems a large variety of classical algorithms, such as for disk scheduling, processor scheduling, memory partition forecast and page replacement. These can be integrated in a user-friendly way using simulators. After analysing the existing ones we found a series of failures and shortcomings that made of little use nowadays. For this reason we implemented various simulators, with a homogeneous interface, and grouped them in a web portal. It is possible to configure for each simulator the parameters of the various available algorithms and view their execution step by step. We obtained in this way an easy to use multiplatform web application, with pre-stored examples and tutorials that guide the user in their experience with each of the simulators. Such resource can be useful both for teachers, to liven up and enrich their explanations (especially when they are carried out by videoconference), and for students, who can actively use these tools to support their autonomous learning process, inside or outside the classroom.

## Palabras clave

Simulador, planificación de disco, gestión de memoria, planificación del procesador, sistemas operativos, recurso docencia.

## 1. Motivación

Los simuladores didácticos contribuyen a un aprendizaje más ameno, interactivo y perdurable. De hecho se encuentran catalogados en la base de la pirámide de Edgar Dale [4]. Por una parte, el profesor puede utilizarlos para explicar conceptos mediante ejemplos que muestren los procedimientos correspondientes con gráficos de calidad y animaciones. También puede configurar parámetros para ver cómo afectan éstos a los resultados, y obtener gráficas y estadísticas de forma dinámica. Por otra parte, el alumno puede luego en diferido reforzar lo aprendido en clase, interactuando por sí mismo, a su ritmo y las veces que desee, con los simuladores. O incluso también puede usarlos para adquirir, de forma autónoma, los conocimientos asociados, bien *a posteriori* porque no haya podido asistir a clase, o como actividad previa a una sesión en modalidad *flipped classroom*.

En el caso concreto de asignaturas de Sistemas Operativos, hay muchos algoritmos clásicos que se prestan a ser comprendidos de una forma más visual, sencilla y eficiente utilizando simuladores implementados para tal propósito. Concretamente, en el tema de planificación del procesador, se estudian diversas políticas para determinar qué proceso, de entre los procesos listos, se selecciona para su ejecución. En el

tema de Gestión de E/S nos encontramos con algoritmos de planificación de disco, para determinar el orden de atención de las solicitudes de lectoescritura, con el objetivo de minimizar el tiempo de búsqueda (para posicionar el cabezal de lectoescritura sobre el cilindro que contiene la petición). Y en el tema de gestión de memoria también podemos abordar diferentes algoritmos, unos para decidir qué bloque libre de memoria principal asignar a un proceso en el caso de particionamiento dinámico, y algoritmos de reemplazo, o políticas de selección de víctima, para decidir qué marco de memoria principal desalojar cuando la memoria está llena en el caso de usar paginación.

Primeramente se hizo una búsqueda y análisis de simuladores existentes de tales algoritmos, y dado que ninguno satisfacía los requisitos que se establecieron, entonces se decidió proponer su implementación a un alumno de Ingeniería Informática, a modo de portal de simuladores como Trabajo de Final de Grado, tutorizado por una de los docentes de dichas asignaturas en la Universidad de las Islas Baleares (UIB).

La motivación por disponer de tales simuladores se vio incrementada por la situación de no presencialidad de las clases teóricas, ligada a la pandemia de COVID-19, ya que permitirían enriquecer sustancialmente las explicaciones por videoconferencia.

El resto del artículo se estructura de la siguiente forma: la sección 2 incluye los simuladores explorados y las conclusiones extraídas de su estudio. La sección 3 explica cómo se ha llevado a cabo el diseño e implementación del portal de simuladores creado. La sección 4 detalla las funcionalidades que cubren los simuladores implementados. La sección 5 comenta la experienca de uso y divulgación del recurso y finalmente, en la sección 6, se aportan las conclusiones del trabajo.

# 2. Exploración de simuladores existentes

La mayoría de sitios web donde se puede interactuar con simuladores ligados a Sistemas Operativos, o descargarlos, solamente abarcan los algoritmos para la gestión de una sola tarea/componente. Únicamente el OS Sim de Alex Macia [9] permite realizar simulaciones de planificación de disco y de procesos, y aspectos de gestión de memoria (aunque sin cubrir los algoritmos de reemplazo). De los restantes 12 simuladores explorados, cinco son para planificación de disco [2, 3, 8, 12, 13], cuatro para planificación de procesador [7, 10, 17, 18] y tres para gestión de memoria [6, 11, 16], si bien varios sólo implementan uno o alguno de los algoritmos clásicos. (por ejemplo el [17] sólo implementa el *Round Robin*).

En [1] se pueden consultar los aspectos positivos y negativos más destacables que se han detectado en cada uno de ellos.

Del estudio de las soluciones existentes, de manera global, se extrajeron las siguientes conclusiones:

- La mayoría de simuladores están desarrollados con unas tecnologías o librerías que han quedado obsoletas [3], y tampoco proporcionan una versión compilada para su ejecución o las dependencias necesarias para compilarlo [2, 3, 18]. O no es posible hacerlos funcionar correctamente debido a instrucciones de instalación incompletas o inexistentes [18].
- Algunos de ellos están desarrollados para ser ejecutados solo en un sistema operativo determinado como Linux [8, 12] o Windows [13] y la mayoría no son adaptativos.
- Algunos utilizan una interfaz de usuario basada en consola [3, 7, 8, 10, 12, 16], mostrando habitualmente los resultados finales en forma de una lista o tabla simple, sin utilizar gráficos o animaciones de calidad, o sin poder ver el paso a paso de los algoritmos hasta llegar a la solución [17]. Tampoco se suele permitir poder avanzar (salvo en [9]), o retroceder a voluntad por los diferentes estados.
- En general, la entrada de datos resulta poco amigable, poco intuitiva y engorrosa, con pocas o nulas opciones de configuración o interacción [6, 7, 10, 13, 17, 16], sin disponer de la posibilidad de utilizar ejemplos predefinidos (salvo en [13, 16]) o sin poder cargar los datos desde un fichero externo (salvo en [10, 12], o en [7] pero sin indicar su formato), ni salvar la configuración del simulador (salvo en [9]). Incluso se puede requerir reentrar los datos para repetir la ejecución de una misma simulación [8].

# 3. Diseño e implementación del portal de simuladores

## 3.1. Requisitos

El nuevo portal de simuladores tiene que resolver las carencias de las soluciones existentes e introducir características que faciliten su uso.

En primer lugar la aplicación debe integrar los principales simuladores de algoritmos de varios de los temas que se tratan en las asignaturas de sistemas operativos como son: la planificación de procesador, la planificación de disco y la gestión de memoria.

Esta aplicación tiene que poder ejecutarse correctamente sobre distintas plataformas y dispositivos, tanto ordenadores como móviles. Además, el usuario debe poder usar la aplicación localmente, sin Internet, una vez ha descargado la aplicación.

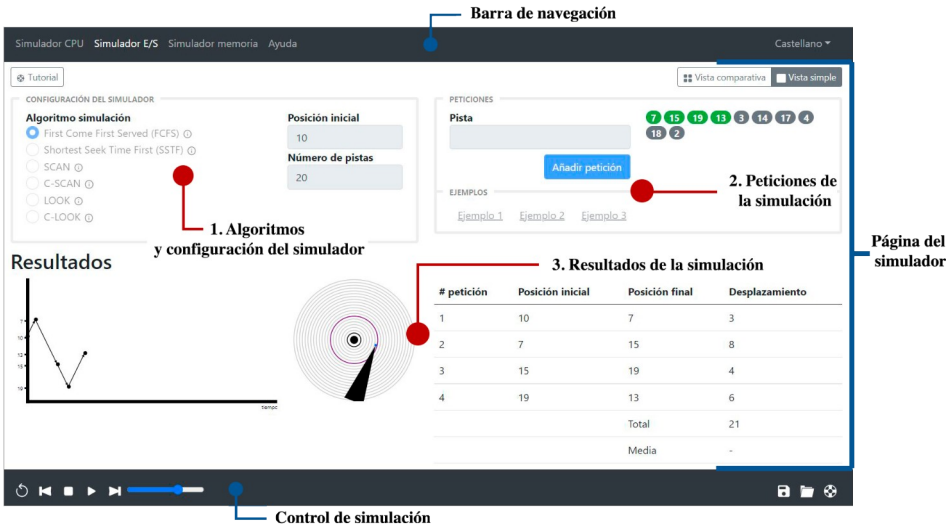



Figura 1: Interfaz del portal de simuladores.

De la misma forma la interfaz de usuario, además de adaptarse a la resolución del dispositivo, debe ser compartida entre los distintos simuladores de la aplicación y disponer de un tutorial que explique paso a paso los distintos elementos de la interfaz.

Con el objetivo de agilizar el uso del simulador y la visualización de los resultados, todos los simuladores deben disponer de ejemplos predeterminados. El usuario también debe poder guardar y cargar configuraciones del simulador desde archivos de texto.

Otra característica que debe estar presente en la aplicación es la posibilidad de visualizar distintos algoritmos simultáneamente para observar las diferencias entre ellos.

Finalmente, el usuario debe poder controlar como se ejecuta la simulación: una simulación paso a paso controlada por el usuario o una simulación automática con una velocidad ajustable.

## 3.2. Organización de la interfaz gráfica de usuario

Uno de los principales problemas de las soluciones existentes es la necesidad de combinar distintos simuladores para poder tratar los principales algoritmos y que cada uno de ellos tiene una interfaz de usuario y manera de interactuar distinta, provocando confusión al usuario.

El portal de simuladores que hemos implementado presenta una interfaz de usuario compartida entre los distintos simuladores y una manera de interactuar común.

La interfaz de la aplicación se puede dividir en 3 secciones, visibles en la figura 1:

1. Una barra de navegación en la parte superior de la ventana. Esta barra de navegación permite navegar por las distintas secciones de la aplicación y cambiar el idioma de la aplicación.
2. Una sección propia de cada una de los simuladores.
3. Una barra de control de la simulación, que permite controlar como se ejecuta la simulación, como se puede observar en la figura 2.

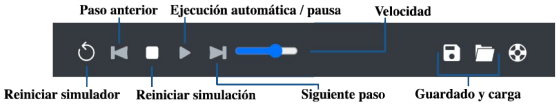



Figura 2: Barra de control del panel de simuladores.

La sección propia de los simuladores depende de cada simulador, ya que cada uno de ellos muestra distintos resultados. A pesar de ésto, es posible diferenciar distintas secciones, como también se puede ver en la figura 1.

1. Una sección para configurar las características del simulador y seleccionar el algoritmo de simulación.
   Cada algoritmo tiene un botón de ayuda que muestra una ventana emergente con una explicación breve del funcionamiento del algoritmo.
2. Una sección para introducir las peticiones que debe atender el simulador.
3. Una sección de resultados que muestra como se han atendido las peticiones en el simulador.

También existe una opción para visualizar el tutorial específico del simulador y un botón para alternar entre la vista simple y la vista comparativa.

La vista comparativa permite visualizar distintos algoritmos procesando las mismas peticiones, cosa que permite ver fácilmente las diferencias en el com-

portamiento de los algoritmos. La principal diferencia en la interfaz se encuentra en la selección de los algoritmos, que se pueden seleccionar múltiples al mismo tiempo. Los resultados de las comparaciones no son tan completos como los de la vista simple por limitaciones de espacio.

Los distintos simuladores tienen tutoriales que explican cómo funciona la interfaz de usuario y guían al usuario en sus primeros pasos. Estos tutoriales marcan zonas de la pantalla y realizan explicaciones sobre estas zonas, como se puede ver en la figura 3.

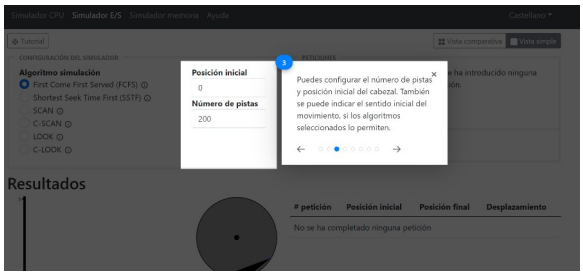



Figura 3: Ejemplo de un tutorial del portal de simuladores.

Otra característica común en los simuladores es la posibilidad de guardar y cargar simulaciones desde archivos de texto. Para guardar la configuración actual el usuario debe utilizar las opciones de la figura 2. Cuando se presione el botón de guardar se mostrará una ventana contextual para introducir el nombre del archivo.

### 3.3. Implementación y uso del simulador

El portal de simuladores se ha desarrollado utilizando tecnologías web como React, aunque este hecho es transparente para el usuario final. El uso de tecnologías web ha permitido que la aplicación sea compatible con todos aquellos dispositivos que puedan ejecutar un navegador web moderno.

La aplicación puede ser accedida utilizando una versión en línea o una versión ejecutable con un navegador web sin conexión:

- Para la versión en línea, el usuario debe acceder al siguiente sitio web[1].
- Para la versión sin conexión, el usuario puede descargar una versión compilada de la aplicación desde el repositorio en Github[2]. Una vez se ha descargado la aplicación, se debe abrir con un navegador web (Chrome, Firefox, Safari, Edge) el archivo principal index.html.

[1] `https://so.jaumealoy.dev/`

[2] En el repositorio de la aplicación `https://github.com/jaumealoy/operating-system-simulators/releases` hay una sección de versiones compiladas en la cual se encuentra un archivo comprimido (.zip) que contiene la aplicación.

Ambas versiones de la aplicación tienen las mismas funcionalidades, solo cambia el modo de acceso y la necesidad de tener una conexión activa a Internet o no.

## 4. Funcionalidades de los simuladores implementados

### 4.1. Planificación de procesador

El simulador de planificación de procesador permite simular los algoritmos tradicionales de la planificación de procesos [14, 15]: *First In First Out (FIFO)*, *Shortest Process Next (SPN)*, *Shortest Remaining Time (SRT)*, *Highest Response Ratio Next (HRRN)*, *Round Robin* y *Feedback*.

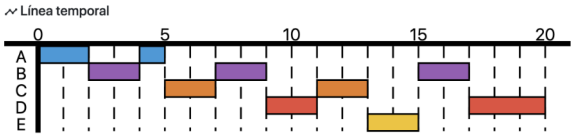



Figura 4: Diagrama temporal resultado de una simulación.

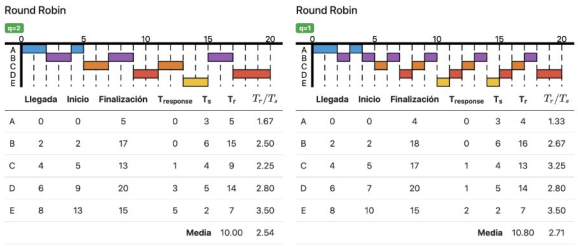
El simulador proporciona un gráfico y una tabla como resultado de la simulación:

- El gráfico de la figura 4 muestra el estado del procesador en cada ciclo de procesador, indicando qué proceso se ha estado ejecutando y cuáles han estado bloqueados.
- La tabla muestra un resumen de la planificación resultante, con métricas como el tiempo de servicio, el tiempo de retorno y su relación.

Algunos algoritmos como el *Round Robin* o el *Feedback* tienen un comportamiento en función de un parámetro de entrada. Por este motivo, en la vista comparativa de algoritmos, es interesante poder comparar el mismo algoritmo pero con valores distintos para tal parámetro. Para ello, el usuario puede crear distintas configuraciones del mismo algoritmo y compararlas simultáneamente como se puede observar en la figura 5.


Figura 5: Configuración de los distintos algoritmos.

Los resultados en la vista comparativa no son tan extensos, pero muestran prácticamente la misma información y permiten ver el comportamiento de cada uno de los algoritmos rápidamente, como se puede ver en la figura 6 con los algoritmos *Round Robin* con valores de *quantum* igual a 2 y 5.




| | Llegada | Inicio | Finalización | $T_{response}$ | $T_S$ | $T_r$ | $T_r/T_s$ |
|---|---|---|---|---|---|---|---|
| A | 0 | 0 | 5 | 0 | 3 | 5 | 1.67 |
| B | 2 | 2 | 17 | 0 | 6 | 15 | 2.50 |
| C | 4 | 5 | 13 | 1 | 4 | 9 | 2.25 |
| D | 6 | 9 | 20 | 3 | 5 | 14 | 2.80 |
| E | 8 | 13 | 15 | 5 | 2 | 7 | 3.50 |
| | | | Media | 10.00 | | | 2.54 |

| | Llegada | Inicio | Finalización | $T_{response}$ | $T_S$ | $T_r$ | $T_r/T_s$ |
|---|---|---|---|---|---|---|---|
| A | 0 | 0 | 4 | 0 | 3 | 4 | 1.33 |
| B | 2 | 2 | 18 | 0 | 6 | 16 | 2.67 |
| C | 4 | 5 | 17 | 1 | 4 | 13 | 3.25 |
| D | 6 | 7 | 20 | 1 | 5 | 14 | 2.80 |
| E | 8 | 10 | 15 | 2 | 2 | 7 | 3.50 |
| | | | Media | | | | 10.80 | 2.71 |

Figura 6: Resultados de la planificación de procesos en la vista comparativa.

## 4.2. Planificación de disco

El simulador de planificación permite simular el orden de atención a peticiones de disco utilizando los algoritmos tradicionales [14, 15] como *First Come First Served (FCFS), Shortest Seek Time First (SSTF)*, *SCAN*, *C-SCAN*, *LOOK* y *C-LOOK.*

Los algoritmos de planificación dependen del número de pistas del disco y la posición inicial del cabezal. Una vez se ha especificado esta información con el formulario de la figura 7, ya se puede realizar la simulación de las peticiones.

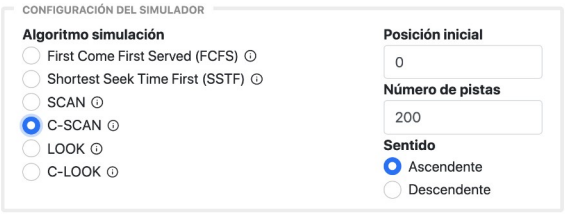



Figura 7: Configuración del simulador de planificación de disco.

En aquellos algoritmos que tienen un comportamiento distinto en función de si el sentido inicial del movimiento es ascendente o descendente, como es el caso de *LOOK* y *SCAN*, aparece una opción para especificar esta información.

Los resultados de la simulación son los siguientes:

- Un gráfico, como el de la figura 8, que muestra la posición del cabezal en función del tiempo. Este gráfico permite visualizar como se han atendido las peticiones.

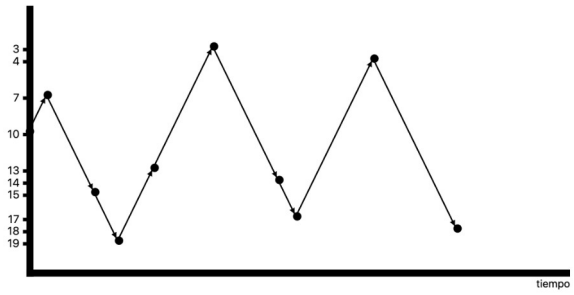



Figura 8: Gráfico de la posición del cabezal del disco en función del tiempo.

- Una tabla, como la de la figura 9, que muestra el orden en el cual se han atendido las peticiones y la distancia que se ha recorrido.
- Un gráfico, como el de la figura 10, que representa un disco duro con un cabezal que realiza una animación entre pista y pista para simbolizar el movimiento real de un cabezal.

| # petición | Posición inicial | Posición final | Desplazamiento |
|---|---|---|---|
| 1 | 10 | 7 | 3 |
| 2 | 7 | 15 | 8 |
| 3 | 15 | 19 | 4 |
| 4 | 19 | 13 | 6 |
| 5 | 13 | 3 | 10 |
| 6 | 3 | 14 | 11 |
| 7 | 14 | 17 | 3 |
| 8 | 17 | 4 | 13 |
| 9 | 4 | 18 | 14 |
| 10 | 18 | 2 | 16 |
| | | Total | 88 |
| | | Media | 8.80 |

Figura 9: Tabla resultado de la simulación de disco.

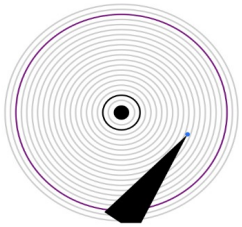


Figura 10: Representación del cabezal sobre el disco duro.

## 4.3. Gestión de memoria

En el tema de gestión de memoria uno de los métodos de asignación que se analizan históricamente es el de particionamiento dinámico usando diversos criterios de asignación de particiones variables [14, 15], los cuales han sido implementado en uno de los simula-

dores. También se estudia el método de asignación de páginas y los diversos algoritmos de remplazo aplicables para la selección de víctima [14, 15].

### Asignación de particiones variables

El simulador de asignación de particiones variables permite observar como se asignan los procesos en la memoria según los algoritmos [14, 15] *First Fit*, *Next Fit*, *Worst Fit*, *Best Fit* y *Buddy System*.

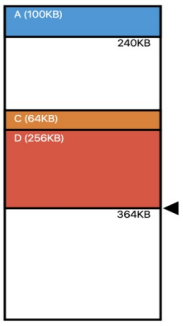



Figura 11: Resultado del simulador de asignación de particiones variables.

El usuario debe introducir los procesos que se tienen que ejecutar, especificando sus características (nombre, duración, espacio necesario y el ciclo de llegada).

El resultado de la simulación es un gráfico que representa la memoria en la cual los distintos procesos se representan como bloques de distintos colores, como se puede ver en la figura 11. También se muestra una tabla, que es un resumen de los procesos que se han asignado.

En función del algoritmo seleccionado el gráfico de memoria muestra una información distinta, como se puede observar en la figura 12:

- Los algoritmos simples, que no requieren información adicional, muestran un gráfico simple.
- El algoritmo *Next Fit* representa el puntero interno utilizado como una flecha sobre la memoria.
- El algoritmo *Buddy System* muestra la divisiones internas que se realizan en la memoria y la fragmentación interna que se produce.

### Reemplazo de páginas

El simulador de reemplazo de páginas permite visualizar cómo se asignan las páginas en la memoria en un sistema de memoria paginada con reemplazo local. Los algoritmos de reemplazo de páginas implementados son: óptimo, *First In First Out (FIFO)*, *Least Recently Used (LRU),* reloj y *Not Recently Used (NRU)*.

Estos algoritmos son propios de los sistemas de paginación local con asignación fija. Para utilizar el simulador se deben especificar los procesos, su cantidad de páginas y las solicitudes de memoria de cada uno de ellos.

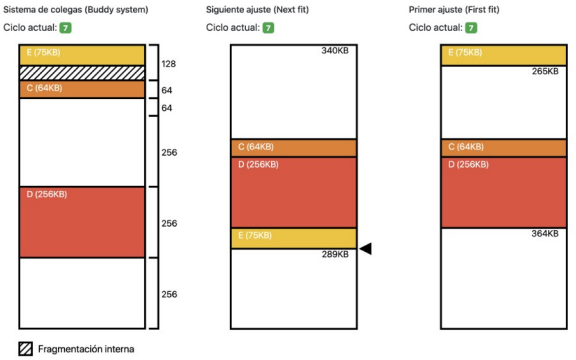



Figura 12: Gráfico de memoria en función del algoritmo seleccionado.

Los resultados del simulador son, como se puede ver en la figura 13:

- Un gráfico que representa una memoria. Este gráfico muestra los distintos marcos y la página que reside en ellos en cada instante.
- La tabla de páginas de cada proceso y su histórico.

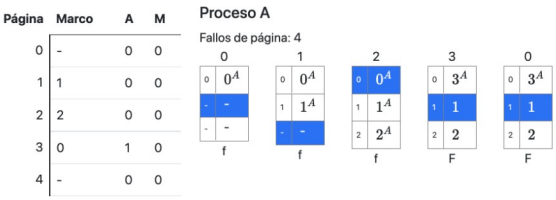



Figura 13: Ejemplo de los resultados del simulador de reemplazo de páginas.

Algunos de los algoritmos como el reloj y el *Least Recently Used* hacen uso de punteros y realizan múltiples pasos antes de asignar una página a un marco. Estas acciones previas son básicamente la actualización de campos de la tabla de páginas.

Con el objetivo de facilitar el seguimiento de estos algoritmos el simulador muestra mediante animaciones todos los cambios que produce el algoritmo en el recorrido de la lista circular antes de llegar al estado final de la memoria.

Por ejemplo, en la figura 14 se está atendiendo una solicitud de la página 3 utilizando el algoritmo *Not Recently Used*. Con esta petición se mostrará de forma animada el recorrido circular sobre los diferentes marcos efectuando cambios en los bits de acceso y modificación antes de determinar cuál será la víctima que será reemplazada y mostrar el estado final.

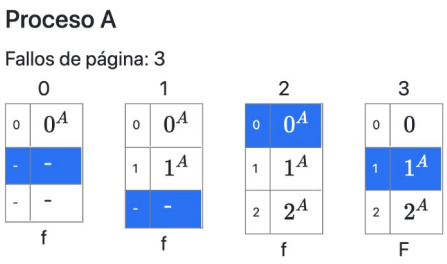



Figura 14: Ejemplo de un paso intermedio del algoritmo.

## 5. Experiencia de uso y divulgación del recurso docente

Durante el segundo semestre del curso 2020-21, se tuvo la oportunidad de utilizar por videoconferencia los simuladores de planificación de disco y los de gestión de memoria, en la asignatura Sistemas Operativos II de la Ingeniería Informática de la Universidad de las Islas Baleares. La experiencia fue muy positiva por parte de la docente al poder mostrar visual y dinámicamente el funcionamiento de los algoritmos involucrados sin tener que dibujar a mano alzada cada uno de los estados de progresión de cada algoritmo, pudiéndose incluso mostrar la ejecución de varios de ellos simultáneamente en pantalla y obtener gráficas comparativas. Por parte de los alumnos, manifestaron satisfacción con el recurso, tanto en los comentarios en directo por el chat durante las sesiones de videoconferencia, como en los agradecimientos que nos hicieron llegar por mensajería del aula digital, resaltando el haber podido probar de forma autónoma los algoritmos con diferentes juegos de datos y el haber podido avanzar y retroceder a voluntad por los diferentes estados de ejecución de cada algoritmo.

También cabe señalar que en la pregunta tipo test del examen relacionada con la planificación de disco el acierto fue del 96,9%, y en la pregunta relacionada con las políticas de reemplazo el acierto fue del 86,7%.

El portal de simuladores también fue evaluado por varios profesores, que habían sido docentes de asignaturas de Sistemas Operativos en la UIB, y que formaron parte del tribunal al que se presentó el recurso como Trabajo de Final de Grado.

Durante el curso 2021-22 se ha tenido la oportunidad de que los alumnos utilizaran los simuladores como refuerzo y ampliación de los ejemplos vistos en clase, y contestaran una encuesta de valoración, basada en la propuesta por [4] y cuyos resultados se sintetizan en la Cuadro 1.

| | Puntuación |
|---|---|
| ASPECTOS TÉCNICOS<br>Calidad de medios, menús de ayuda, variedad de opciones, tamaño de gráficos y letras, respuesta a acciones, carga de la web, relación coste-calidad | 8,65 |
| CALIDAD DE CONTENIDOS<br>Actualización, calidad, secuencia y estructura, originalidad, claridad de explicaciones, idiomas | 8,43 |
| MOTIVACIÓN DE USO<br>Grado de atracción de la herramienta, interés que despierta, duración, alcance de logros intermedios | 8 |
| ORGANIZACIÓN INTERNA DE LA INFORMACIÓN<br>Incluye ejemplos y tutoriales, síntesis de fundamentos, interacción web, información textual auxiliada por recursos multimedia | 8,87 |
| VALOR DIDÁCTICO<br>Adaptación al temario, favorece el proceso de aprendizaje, vistas comparativas, retroalimentación | 8,96 |
| CALIDAD DE DISEÑO<br>Coherencia de estilo gráfico, zonas estables en pantalla, tamaños de fuentes, contraste de colores, distribución de elementos | 7,83 |

Cuadro 1: Resultados de la encuesta de valoración.

## 6. Conclusiones

Las asignaturas de Sistemas Operativos conllevan muchos algoritmos de gestión de recursos tales como procesador, disco, o memoria, cuyo funcionamiento resulta mucho más fácil de explicar y comprender mediante el uso de simuladores que permitan ver el paso a paso de su ejecución, de forma visual e interactiva.

Los simuladores explorados no satisfacen plenamente las necesidades de docentes y alumnos, por razones como ser incompletos, no estar orientados al usuario final, quedar obsoletos por falta de mantenimiento (e incluso no poderse ejecutar con las tecnologías existentes), no ser adaptativos, o no ser suficientemente intuitivos y además carecer de instrucciones de uso.

Para paliar esas deficiencias y poder aplicar una metodología activa y atractiva en la docencia de este tipo de algoritmos, se ha desarrollado un portal web multiplataforma que aglutina de forma homogénea diversos simuladores y que puede ser utilizado en dife-

rentes dispositivos, tanto en línea como de forma local previa descarga.

Los simuladores llevan incorporados un tutorial de uso, así como un icono para cada algoritmo que permite abrir una ventana contextual explicando en qué consiste tal algoritmo. También se ha dotado a cada simulador de un conjunto de ejemplos predefinidos, extraídos de las principales bibliografías de Sistemas Operativos. Además permite la entrada controlada de datos para la creación de nuevos ejemplos de forma dinámica.

En cada simulador, la ejecución puede ser de un sólo algoritmo o simultáneamente de varios seleccionados para su comparación, pudiéndose detener, avanzar y retroceder la visualización de los resultados en cada paso.

El portal puede ser de gran utilidad tanto para docentes, facilitándole la explicación de los algoritmos de manera visual y animada, como para los alumnos que experimenten con ellos de forma autónoma, consiguiéndose así un aprendizaje más activo y significativo.

# Referencias


[1] Jaume Aloy. Desenvolupament de simuladors per a la docència de sistemes operatius. 2021. Disponible en https://dspace.uib.es/xmlui/handle/11201/157224. Accedido en febrero 2022.

[2] Raghav Arora. Disk scheduling and memory management. 2018. Disponible en https://github.com/lennon2298/disk-scheduling-mem-mgmt. Accedido en julio 2021.

[3] Scott Bouloutian. Disk Scheduling Simulation. 2015. Disponible en https://github.com/ScottBouloutian/Disk-Scheduling-Simulation. Accedido en julio 2021.

[4] Julio Cabero-Almenara y Jesús Costas. La utilización de simuladores para la formación de los alumnos. En *Prisma Social*, núm. 17, diciembre 2016, pp. 343-372.

[5] Edgar Dale. Audiovisual methods in teaching. 1969.

[6] Jamie Goodson. Memory Management Simulator. 2017. Disponible en https://github.com/jamiegdsn/memory-management-simulator. Accedido en julio 2021.

[7] Minsu Kim. CPU Scheduling Algorithm Simulator. 2019. Disponible en https://github.com/alstn2468/CPU_Scheduling_Simulator. Accedido en julio 2021.

[8] Brian Kirotich. Disk Scheduling Simulation. 2013. Disponible en https://github.com/kirotich/disk-scheduling. Accedido en julio 2021.

[9] Alex Macia. OS Sim. 2015. Disponible en https://sourceforge.net/projects/oscsimulator/. Accedido en julio 2021.

[10] Jason Marcel. Scheduler. 2009. Disponible en https://github.com/jasmarc/scheduler. Accedido en julio 2021.

[11] Padua University. SiGeM. 2014. Disponible en https://sourceforge.net/projects/sigem/. Accedido en julio 2021.

[12] Safiyat Reza. Disk Scheduling. 2013. Disponible en https://github.com/safiyat/DiskScheduling. Accedido en julio 2021.

[13] Menno Sijben. Disk Scheduling Simulation. 2017. Disponible en https://github.com/mpsijben/DiskSchedulingSimulation. Accedido en julio 2021.

[14] Abraham Silberschatz, Peter Baer Galvin y Greg Gagne. Operating System Concepts Essentials (2nd edition). 2014.

[15] William Stallings. Operating Systems: Internals and Design Principles (9th edition). 2018. Pearson.

[16] Dakota Szabo. Memory Management Simulation. 2012. Disponible en https://github.com/szabodabo/Memory-Management-Simulation. Accedido en julio 2021.

[17] Ruchiranga Wickramasinghe. Scheduling Simulator. 2014. Disponible en https://github.com/Ruchiranga/SchedulingSimulator. Accedido en julio 2021.

[18] Ahmad Yayha y Hamed Hijazi. Operating System Scheduling. 2019. Disponible en https://github.com/AhmadYahya97/OperatingSystemScheduling. Accedido en julio 2021.