DISEÑO DE UNA APLICACIÓN WEB PARA LA ENSEÑANZA-APRENDIZAJE DEL REMPLAZO DE PÁGINAS EN LA MEMORIA VIRTUAL

DESIGN OF A WEB APPLICATION FOR THE TEACHING-LEARNING OF PAGE REPLACEMENT IN THE VIRTUAL MEMORY

Hilda Castillo Zacatelco
Benemérita Universidad Autónoma de Puebla, México
hildacz@gmail.com

Claudia Zepeda Cortés
Benemérita Universidad Autónoma de Puebla, México
czepedac@gmail.com

José Luis Carballido Carranza
Benemérita Universidad Autónoma de Puebla, México
jlcarballido7@gmail.com

Carmen Cerón Garnica
Benemérita Universidad Autónoma de Puebla, México
carmen.ceron@correo.buap.mx

Edgar Castro Martínez
Benemérita Universidad Autónoma de Puebla, México
edgarcm1908 @gmail.com

Recepción: 6/diciembre/2022 **Aceptación:** 14/abril/2023

Resumen

En el curso de sistemas operativos, uno de los temas centrales es la administración de memoria virtual mediante la técnica de paginación. Esta técnica involucra a los denominados algoritmos de remplazo de páginas. En este trabajo se describe el análisis y diseño de un simulador interactivo de paginación para la enseñanza-aprendizaje de los algoritmos de remplazo de páginas, implementado mediante una aplicación web. Además, este simulador cuenta con un módulo de despacho de procesos que utiliza el algoritmo Round Robin, y un módulo que permite elegir entre tres algoritmos de remplazo de páginas: FIFO, conjunto de trabajo y WsClock. Actualmente se cuenta con un prototipo de este simulador, con un diseño de vistas fáciles de usar y de entender.

Palabras Clave: Aplicación web, despacho de procesos, paginación, remplazo de páginas, sistemas operativos.

Abstract

In the course of operating systems, one of the central themes is the management of virtual memory through the paging technique. This technique involves the so-called page replacement algorithms. In this work, the analysis and design of a paging interactive simulator for the teaching-learning of page replacement algorithms, implemented through a web application is described. Besides, this simulator includes a processes scheduler module that uses the Round Robin algorithm, and a module that allows to choose among three algorithms to replace pages: fifo, work set and wsclock. Currently, there is a prototype of this simulator, with a design of easy to use and understand view-features.

Keywords: *Operating systems, page replacement, paging, process scheduler, web application.*

1. Introducción

El tema de administración de memoria virtual en sistemas operativos es de vital importancia. Éste involucra técnicas de control de carga, algoritmos de remplazo de páginas, diferentes formas de implementar una tabla de páginas; además, involucra conceptos como espacio de direcciones virtuales, fallo de página, hiperpaginación, transformación de direcciones virtuales en físicas, etc. En la administración de memoria virtual se estudian dos técnicas principales, paginación y segmentación [Tanenbaum, 2009].

La memoria virtual permite hacer creer al usuario de la computadora y a los programas que la cantidad de memoria es mayor de la que realmente tienen. Para lograr esto, además de la memoria física (RAM), se utiliza memoria secundaria (disco duro). Así, ciertas partes de los procesos se encontrarán en memoria física y otros en memoria secundaria. El espacio de direcciones virtuales se divide en bloques denominados páginas. Para la administración de estas páginas, se utiliza una estructura denominada tabla de páginas. Existen diferentes tipos de tablas de

páginas, en función de la forma en la cual se implementen, en donde, la más simple es la tabla de páginas directa.

En el caso de la tabla de páginas directa, cada localidad contiene el número de marco para página, en donde se encuentra la página asociada a esa localidad, en el caso de que ésta se encuentre en memoria física, en caso contrario, la página estará en memoria secundaria. Esta información está representada en el bit presente/ausente, 0 si está ausente, 1 si está presenta en la memoria física.

Cuando una página que no está en memoria física es invocada, se produce lo que se conoce como un fallo de página. En este caso, se lee la página de memoria secundaria y se carga en memoria física. Si la memoria física está llena, entonces se debe elegir una página de memoria física y bajarla a memoria secundaria para poder cargar la página invocada. Esto se realiza tomando en cuenta que las instrucciones de una página deben estar en memoria principal para poder ejecutarse. Para elegir qué página se va a desplazar de la memoria física, existen varios criterios que dan origen a varios algoritmos, entre los que podemos mencionar: fifo, nru, lru, wsclock, conjunto de trabajo, segunda oportunidad, entre otros [Tanenbaum, 2009].

La gran cantidad de conceptos y técnicas de este tema, pueden hacer que el estudiante se pierda entre ellos y se confunda, por lo que es importante construir una herramienta gráfica e interactiva, que le permita al estudiante seguir paso a paso la forma en la cual se elige la página que se remplazará en la memoria física y se enviará a la memoria secundaria. Además de comprender la forma en que trabaja el algoritmo de remplazo de páginas, el estudiante comprenderá la diferencia entre la memoria física y la memoria virtual. Este simulador deberá ser muy simple, fácil de instalar, con mínimos requisitos de software.

Algunos trabajos relacionados se mencionan a continuación. En [Cahya, 2009], se presentan los resultados de la implementación del primer módulo de un simulador de un sistema operativo, en donde se simula la memoria y el CPU. En [Sibai, 2017], se modificó el código de MOSS [MOSS, 2022] para desarrollar un simulador de memoria virtual que permita intercambiar entre diferentes algoritmos de remplazo de páginas. En [England, 2005], se presenta el diseño e implementación de Moses

2, que es un simulador de memoria virtual que utiliza segmentación paginada. En [Ramos, 2008], se presenta un análisis para la implementación de un simulador para problemas de diseño de disco y administración de memoria. En [Trefftz, 2015], se presentan los resultados de un simulador de hardware referente a procesador, registros y memoria. En [Buendía, 2005], se describen dos herramientas, la primera es un simulador que muestra el funcionamiento básico de la gestión de memoria y la segunda herramienta es una aplicación web que permite el acceso remoto a funciones reales de un sistema operativo. En [García, 2008], se presenta el análisis y diseño de un simulador de algoritmos de manejo de memoria orientado al diseño de un simulador de algoritmos de remplazo de páginas. En [Losa, 2021], se presentan nuevas funcionalidades del simulador LeonViP, el cual busca simular el comportamiento del procesador LEON3. Este simulador incluye un módulo MMU que permite simular memoria virtual sin utilizar hardware real. En [Castillo, 2018], se presenta el diseño de un simulador de asignación de memoria que utiliza heurísticas de empaquetamiento como el primer ajuste y el siguiente ajuste, este simulador incluye un módulo de despacho de procesos. En internet se puede encontrar código para implementar simuladores de memoria virtual [Simulador de memoria virtual, 2022], [Virtual Memory Simulation, 2022], [MOSS, 2022] e incluso videos que explican esta técnica.

En los trabajos anteriores se presenta código que se puede utilizar para implementar simuladores de memoria virtual o para realizar comparaciones entre los diferentes algoritmos de remplazo de páginas, pero no existe alguno que sea totalmente visual y, además, interactivo. Por lo que en este trabajo se describe el análisis y diseño de un simulador interactivo de paginación para la enseñanza-aprendizaje de los algoritmos de remplazo de páginas (de ahora en adelante denominado solamente como simulador), implementado mediante una aplicación web, que permitirá observar, de forma gráfica, la manera en la que se lleva a cabo el remplazo de páginas en la técnica de paginación. El simulador propuesto permitirá al usuario elegir el algoritmo de remplazo de páginas, y el tamaño de la memoria virtual y física a utilizar, aunque para poder visualizar de forma correcta los gráficos que representan estas memorias, fue necesario limitar su tamaño.

El documento está organizado de la siguiente manera, en la sección 2 se presenta la metodología que se utilizó para la construcción del simulador, en la sección 3 se presentan los resultados obtenidos, en la sección 4 se presenta una breve discusión de resultados y finalmente, en la sección 5 se presentan las conclusiones del trabajo.

2. Métodos

Tomando en consideración la necesidad de contar con un simulador interactivo de paginación para la enseñanza-aprendizaje de los algoritmos de remplazo de páginas, se propuso desarrollar una herramienta con las características anteriormente descritas.

Como primera etapa (análisis), se investigó y documentó el funcionamiento de los algoritmos de remplazo de páginas a utilizar en el simulador. El diseño está basado en componentes y clases.

Durante la fase del análisis se encontró que el simulador debe mostrar en pantalla una representación de la tabla de páginas que corresponda a la memoria virtual, y la representación de la memoria física asociada. De esta forma el estudiante puede observar cuándo es necesario invocar a un algoritmo de remplazo de páginas y cuándo se produce un fallo de página. Se propone trabajar con al menos dos algoritmos de remplazo de páginas como FIFO y WsClock.

Se usaron tecnologías web, ya que dan la posibilidad de tener portabilidad al desplegar la aplicación en una página web o hacerla de escritorio mediante un navegador embebido.

Debido a las restricciones de espacio de la aplicación web, la memoria virtual podrá presentar a lo más 16 entradas de la tabla de páginas y 8 marcos para página de la memoria física.

Una vez establecido el objetivo y con la investigación realizada con anterioridad, lo primero en la etapa de planificación y diseño, fue proponer un diagrama de las partes o componentes de un sistema operativo. De acuerdo con el análisis realizado, el simulador debe contar al menos con un módulo que represente al CPU, un módulo de administración de memoria denominado MMU, un módulo de

despacho de procesos (Despachador) y un módulo que permita cargar en memoria los procesos que van llegando (Cargador).

Los módulos de despacho de procesos y de administración de memoria componen al módulo del CPU. Se implementó el algoritmo de Round Robin como despachador que utiliza una cola de procesos. Este planificador debe tener un cargador de procesos, el cual ordena la lista de procesos de entrada y los va agregando a la cola de procesos dependiendo de su tiempo de arribo y el tiempo de ejecución del CPU. El módulo de administración de memoria virtual (MMU), es el encargado de todas las operaciones de sus componentes internos que son la memoria física (Memoria física) y la memoria virtual (Memoria virtual) utilizando la técnica de paginación. Entre las funciones del módulo MMU están las de cargar procesos a memoria, liberar de memoria procesos que hayan terminado su ejecución, realizar las referencias a memoria y gestionar los fallos de página.

En cuanto a los procesos, éstos podrán tener a lo más el número de páginas que tiene la memoria virtual. Si un proceso no cabe en la memoria virtual entonces se deberá presentar al usuario un mensaje de que el proceso no puede ser ejecutado. La aplicación web deberá poder ser visualizada desde cualquier navegador, lugar, y sistema operativo. En la fase de diseño, se propone la construcción del simulador mediante el patrón de diseño Modelo-Vista-Modelo de Vista (MVVM), utilizando el framework de VueJS, quien utiliza este patrón como estándar. La parte lógica se debe implementar con el paradigma orientado a objetos, sin algún patrón de diseño en concreto. Como se aprecia en la figura 1, la idea es que el CPU trabaje en conjunto con el Despachador y la MMU al mismo tiempo o de manera paralela.

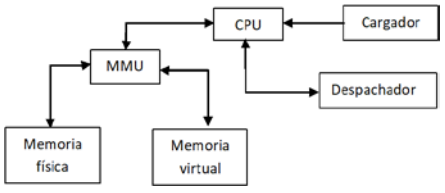

Figura 1 Módulos de la aplicación web.

Es decir, que en el mismo tiempo de CPU se pueda realizar tanto el manejo de procesos, como la administración de memoria. Esto es, agregar nuevos procesos o

eliminarlos, despachar el siguiente proceso y, en cuanto a la memoria, realizar las referencias a páginas, cargar procesos a memoria, o eliminarlos. En la figura 2 se muestra el diagrama de clases del simulador.

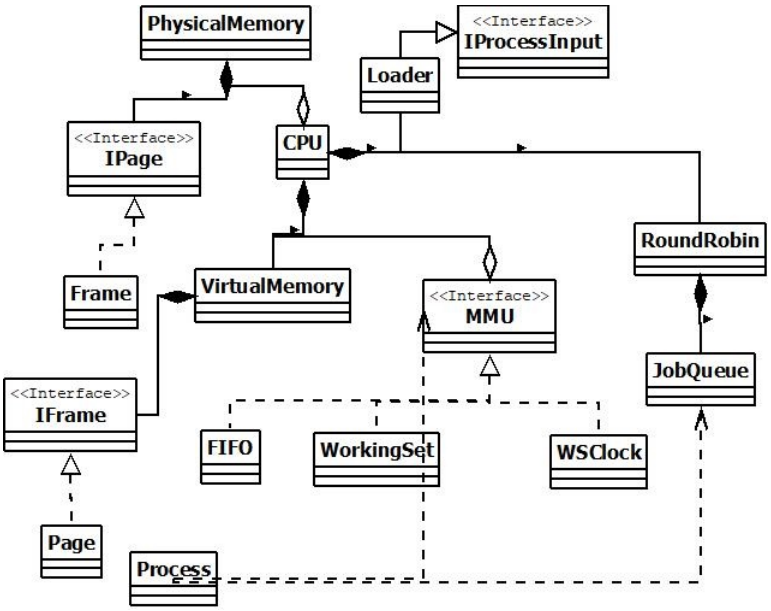

Figura 2 Diagrama de clases.

Para el diseño de la interfaz gráfica se trabajó de la siguiente forma. En primer lugar, se hicieron las propuestas como mockups, diseños de baja fidelidad o bosquejos simples, solo con el objetivo de plasmar la idea. Una vez ya aprobado el mockup, se procedió a generar un diseño con mayor detalle, para esto se utilizó un patrón de diseño llamado “Neumorphism”, que consiste en tener componentes gráficos de un solo color y destacarlos mediante el uso de sombras y luces, creando un efecto visual de profundidad. Además, se utilizó el patrón de diseño gráfico llamado “Atomic design” o “Diseño atómico”. En la figura 3 se muestran dos ejemplos de mockup y sus correspondientes diseños de alta fidelidad o preliminares. Una vez ya definidos los diseños, tanto de funcionamiento como visuales, se inició el desarrollo de la interfaz gráfica.

Al hacer uso de Atomic Design se puede explicar la manera en la cual se implementó este patrón. En primera instancia, se parte de la idea de que un componente (botón, input, imagen) es un átomo, si se reúnen varios átomos se tienen las moléculas,

que son como una tarjeta de información, o parte de un formulario. Si se unen las moléculas se obtienen organismos, que en la estructura del proyecto son las vistas, y si se unen a los organismos se llega a un template o plantilla, que sería el archivo en el directorio src, App.vue. En este archivo se reúnen todas las vistas y se realiza la navegación entre ellas.

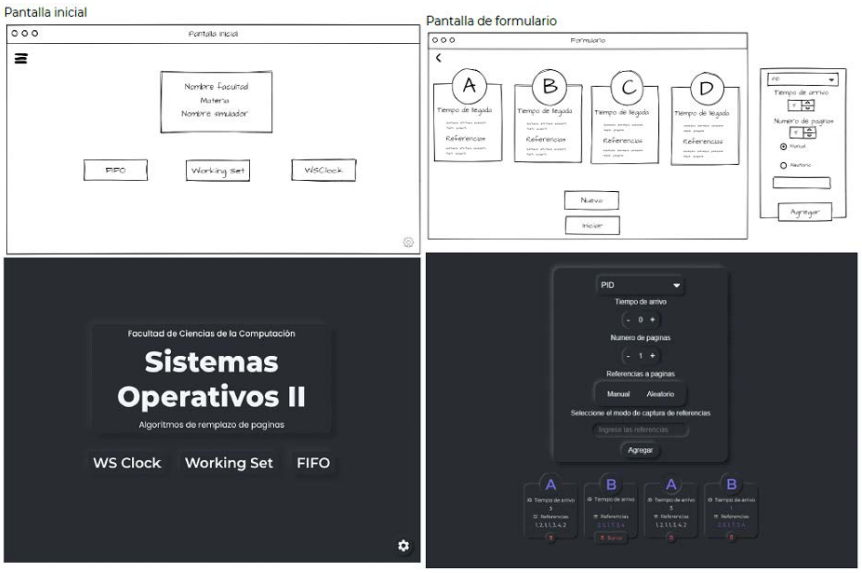

Figura 3 Mockup y su diseño de la página principal y de la pantalla de formulario.

En las vistas se realiza la unión de componentes individuales y dependiendo qué vista sea, se realiza la conexión con la parte lógica almacenada en una carpeta llamada model.

Después de tener la estructura del proyecto, se empezó a desarrollar componente por componente, según se planearon o se iban requiriendo. Hasta el momento, la lista de componentes está conformada por: botón(button), selector de opciones (dropdown), icono (icon), entradas de texto y numéricas (input text, input number), selector de radio (radio) y un switch (toggle).

Con los componentes ya desarrollados se fueron creando componentes más complejos que involucraban la unión de varios componentes simples, como es el caso de las tarjetas de información de un proceso (process card).

Para que el simulador fuese más interactivo, se agregaron un conjunto de configuraciones que el usuario puede cambiar, como lo son: el tamaño de la fuente,

cambiar entre tema claro y oscuro, y elegir un color de acentuación para la parte gráfica. Para la parte de la funcionalidad, el usuario tiene la posibilidad de configurar el número máximo de páginas por proceso de entrada, y la unidad de quantum del simulador. La interfaz cuenta también con una sección de información donde se encuentra un enlace a un repositorio de GitHub en el que se aloja el código y los créditos. Para el desarrollo de la vista se utilizó el framework de JavaScript VueJS y el framework de CSS SCSS. El mapa del sitio actual se muestra en la figura 4.

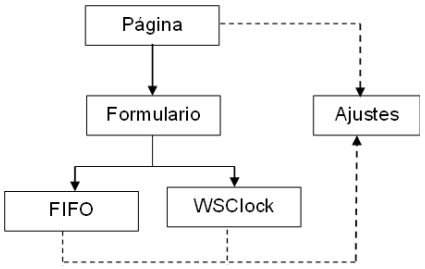

Figura 4 Mapa de sitio.

Para la implementación del simulador se hizo uso de las siguientes herramientas: JavaScript VueJS, que es un framework de JavaScript para desarrollar la interfaz gráfica de la aplicación; TypeScript, que es un lenguaje tipado basado en JavaScript, se puede usar sobre el lenguaje nativo del framework de JavaScript VueJS para tener un mejor control de tipos; Electron, el cual es un framework de código abierto que permite crear aplicaciones multiplataforma con tecnologías web, Electron permite distribuir la aplicación en forma de ejecutable, para poder hacer uso de ella sin necesidad de una conexión a internet; NodeJS, que es una tecnología que ofrece un entorno en tiempo de ejecución multiplataforma para la capa de servidor, en la presente aplicación es la tecnología que ejecuta, compila y construye el simulador; y finalmente, Git y GitHub, donde Git es la tecnología usada para el control de versiones de modo local, y GitHub como control de versiones, pero con un alojamiento remoto. El control de versiones ayuda a tener un control sobre qué instancia del proyecto es la que se está realizando, y en caso de que el código sufra muchas modificaciones, poder volver a una instancia anterior. Por otro lado, GitHub da la posibilidad de hospedar una página web de forma gratuita, de este modo se

pude desplegar la ampliación vía web. Actualmente se ha construido un prototipo de este simulador el cual debe visualizarse en alguno de los siguientes navegadores: Chrome, Brave o Firefox.

3. Resultados

El principal resultado de este proyecto, hasta el momento, es el análisis, diseño y la implementación de un prototipo de una aplicación web que permitirá seguir paso a paso algunos algoritmos de remplazo de páginas como FIFO y WSClock asociados a la técnica de administración de memoria virtual, y en el que también se hace uso del algoritmo de despacho de procesos denominado Round Robin. Esta aplicación va dirigida a alumnos del curso de Sistemas Operativos II o en general a quien desee analizar el funcionamiento de los ya mencionados algoritmos de remplazo de páginas. Todo el código se subió a un repositorio público de GitHub (https://github.com/ObjetosDeAprendizajeFCCBUAP/Simulador-de-paginacion/), y el mismo repositorio y página provee de un servicio llamado GitHub Pages, que permite desplegar alguna aplicación web de manera gratuita (https://objetosdeaprendizajefccbuap.github.io/Simulador-de-paginacion/#/).

Pensando en que no siempre se cuenta con una conexión a internet estable, se generó un ejecutable de la aplicación, el cual, de igual forma se puede descargar desde el repositorio. La figura 5 muestra la vista de la pantalla de inicio del simulador.

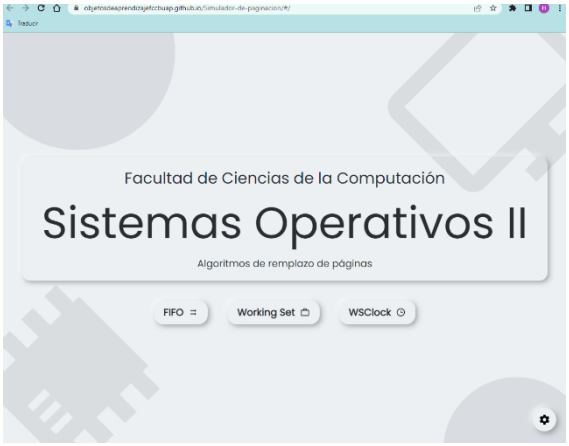

Figura 5 Vista de la pantalla principal del simulador.

En la figura 6, se muestra la vista para el algoritmo de remplazo de páginas WSClock, el cual incluye los siguientes elementos: (1) un panel donde se mostrará el proceso con la página a la que se está haciendo referencia en ese momento, el número de proceso que se han completado en el simulador y el número de fallos de página que ha tenido el simulador; (2) una lista de los procesos que están actualmente en el CPU; (3) la representación de la memoria física mediante una lista; (4) la representación de la memoria virtual mediante una lista; y por último, tres botones de acción en la parte inferior del simulador,(E) el botón para salir del simulador y regresar a la página principal, (N) el botón para ejecutar la siguiente acción del simulador, y (S) el botón para acceder a las configuraciones de apariencia del simulador.

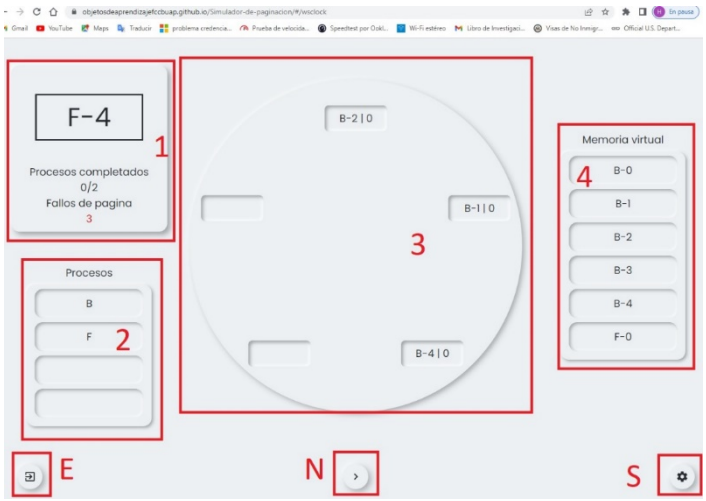

Figura 6 Vista del simulador para el algoritmo de WSClock.

En la figura 7 se muestra la vista del algoritmo de remplazo de páginas FIFO, el cual incluye los siguientes elementos: (1) un panel donde se mostrará el proceso con la página que se está procesando en el CPU en ese momento, el número de proceso que se han completado en el simulador y el número de fallos de página que ha tenido el simulador; (2) una lista de los procesos que están actualmente en el CPU; (3) la representación de la memoria física mediante una lista; (4) la representación de la memoria virtual mediante una lista. Por último, se encuentran tres botones de acción en la parte inferior del simulador, (E) el botón para salir del simulador y

regresar a la página principal, (N) el botón para ejecutar la siguiente acción del simulador, y (S) el botón para acceder a las configuraciones de apariencia de pantalla.

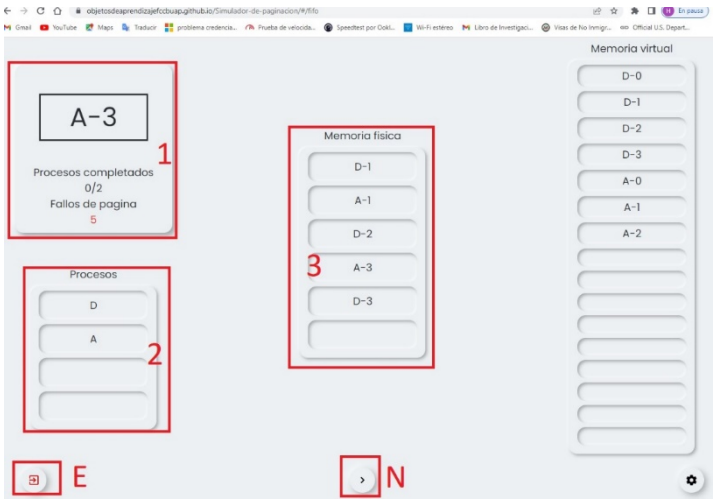

Figura 7 Vista del simulador para el algoritmo FIFO.

Como se podrá observar al ver las figuras, las vistas del simulador son agradables al usuario, sencillas y fáciles de interpretar. Por lo que será útil para aprender los conceptos acerca de la técnica de paginación.

4. Discusión

Después de llevar a cabo el análisis y el diseño del simulador, se construyó un primer prototipo, en donde se pueden observar las diferentes vistas que tendrá el simulador, poniendo atención en la distribución de cada uno de los elementos gráficos y la información que es conveniente que el estudiante observe.

Este primer prototipo permite cargar algunos procesos y el tamaño de la tabla de páginas es de a lo más 16 entradas. El simulador tiene un botón que permite ir avanzando la ejecución de un proceso paso a paso, invocando a ciertas páginas, provocando fallos de página y en el caso de que la memoria física esté llena, invocando algún algoritmo de remplazo de páginas. Esto último aún está en proceso de implementación y se debe realizar una revisión exhaustiva del funcionamiento del simulador. A decir de los estudiantes, el simulador es fácil de usar y de entender,

sin embargo, para asegurar esto es necesario realizar pruebas de usabilidad. Además, con la tecnología propuesta para su implementación, el simulador puede utilizarse en cualquier dispositivo móvil.

5. Conclusiones

En últimas fechas, los alumnos hacen uso de material audiovisual para su aprendizaje, y mejor aún si éste es interactivo. Por lo que proponer un simulador para aprender algoritmos de remplazo de páginas es importante. Con esta aplicación se pretende que los estudiantes en verdad comprendan cómo se maneja la memoria virtual y cómo actúa un algoritmo de remplazo de páginas.

La tecnología actual para el desarrollo de aplicaciones web permite utilizar frameworks que hacen que las aplicaciones sean portables y puedan utilizarse en diferentes dispositivos. De esta forma, el estudiante tiene una herramienta, que en este caso es el simulador, a la mano para repasar ciertos conceptos.

Derivado de este trabajo, se obtuvo el análisis y diseño de un simulador interactivo de paginación para la enseñanza-aprendizaje de los algoritmos de remplazo de páginas, en donde se puede observar el paso a paso de la ejecución de los procesos, y en donde los estudiantes pueden cargar procesos y decidir el número de páginas de un proceso y cuál será el orden en el cual éstas serán referenciadas. La aplicación es un prototipo, y como trabajo a futuro se propone la implementación completa del simulador y la realización de pruebas de usabilidad. Tanto la aplicación web como el código, estarán a disposición de todo público en la web.

6. Bibliografía y Referencias

[1] Buendía, F., Cano, J.C., & Sahuquillo, J. Uso de simuladores y herramientas web para la enseñanza de Sistemas Operativos. Actas del Simposio Nacional de Docencia en la Informática, SINDI2005 (AENUI), 121-128, 2005.

[2] Castillo Zacatelco, H., De la Rosa Flores, R., Zepeda Cortés, C., Cervantes Márquez, A.P., & Cuautle Aguilar, E. Simulador con heurísticas de empaquetamiento para la asignación de memoria y despacho de procesos. Research in Computing Science, 147 (8), 161-174, 2018.

[3] Cahya S. Designing Operating System Simulator: A learning Tool. 11th International Conference on Computer Modelling and Simulation, 156-160, 2009.

[4] England, R. E. Teaching concepts of virtual memory with the Moses2 microcomputer operating system environment simulator. Journal of Computing Sciences in Colleges, 20(6), 84-91, 2005.

[5] García Hernández, Y. Análisis y diseño de un simulador de algoritmos de reemplazo de páginas para la asignatura de Sistemas Operativos en la Universidad de las Ciencias Informáticas. Tesis licenciatura, Universidad de las Ciencias Informáticas, Cuba, 2008.

[6] Losa Cruz, B. Implementación y validación de nuevas funcionalidades para el simulador LeonViP. Tesis licenciatura, Universidad de Alcalá, España, 2021.

[7] MOSS http://www.ontko.com/moss/ Último acceso: 22 de septiembre de 2022.

[8] Ramos León, L., & Pulido Piña, Y. Análisis de los módulos planificación de disco y administración de memoria de un laboratorio virtual de apoyo a la asignatura de Sistemas Operativos. Tesis Licenciatura, Universidad de las Ciencias Informáticas, Cuba, 2008.

[9] Sibai, F. N., Ma, M., & Lill, D. A. Development of a Virtual Memory Simulator to Analyze the Goodness of Page Replacement Algorithms. In 2007 Innovations in Information Technologies (IIT), IEEE, pp. 536-540, 2007.

[10] Simulador de memoria virtual https://github.com/Ricardo96r/Simulador-de-memoria-virtual Último acceso: 22 de septiembre de 2022.

[11] Tanenbaum, A.S. Sistemas Operativos Modernos. Pearson Educación, Tercera edición, México, 2009.

[12] Trefftz Gómez, H., & Cardona Mc'Cormick, J.F. Enseñanza de Sistemas Operativos con un simulador didáctico fácilmente extensible. Encuentro Internacional de Educación en Ingeniería (ACOFI 2015), 1-8, 2015.

[13] Virtual Memory Simulation https://github.com/goodday451999/Virtual-Memory-Simulation Último acceso: 22 de septiembre de 2022.