

Decisiones del sistema

Joan David Saenz Botero

Nicolas Felipe Rueda Aguilar

Sistemas operativos

Grupo 503M

Julián David Alvarado Caicedo

27 de febrero de 2026

Introducción

Los sistemas operativos han sido una de las herramientas más importantes y utilizadas a nivel global tanto de forma empresarial como de forma casual por clientes que cuentan con dispositivos tecnológicos de uso cotidiano, beneficiando aspectos como el rendimiento, la eficiencia y la experiencia de usuario. Históricamente su evolución ha desencadenado la aparición de nuevas tecnologías permitiendo una mayor accesibilidad y expandiendo su uso y sus funcionalidades. El objetivo de esta actividad será profundizar en como cada paso en la evolución de los sistemas operativos permitió llegar a las tecnologías que conocemos hoy en día. Adicional, se realizarán pruebas de rendimiento y comparaciones entre resultados para llegar a conclusiones claras acerca de cómo influyen estos en el rendimiento del dispositivo.

Evolución de los sistemas operativos (SO)

La evolución de los sistemas operativos se encuentra relacionada a las demandas de procesamiento de sectores como los industriales, militares y académicos. Antes de que existieran sistemas que permitieran comunicarse entre equipos, los programadores interactuaban directamente con el hardware mediante código de maquina (ceros y unos) (*Espinoza Rojas, s.f.*).

La primera generación de sistemas operativos surge en 1950 con *GM-NAA I/O*, creado por General Motors siendo considerado el primer sistema operativo con aplicaciones claves en investigaciones corporativas y desarrollo militar (*Espinoza Rojas, s.f.*).

Tiempo después, con la llegada de la segunda generación en 1960, los ámbitos académicos e investigativos tuvieron una fuerte presencia con sistemas como UMES, de la universidad de Michigan y el BESYS, utilizado en los laboratorios Bell, los cuales trajeron consigo la multiprogramación y el tiempo compartido, permitiendo que varios usuarios pudieran interactuar entre sí en un mismo mainframe, reduciendo y optimizando los tiempos de respuesta (*Espinoza Rojas, s.f.*).

La llegada de los microprocesadores en la cuarta generación en 1980 permitió que los computadores fueran más accesibles en tamaño y costo para fines académicos y facilitando la creación de bases de datos, resolviendo así problemas y afectaciones empresariales (*Espinoza Rojas, s.f.*).

Por último, la necesidad de mayor poder de cómputo abrió paso a los sistemas paralelos y distribuidos en los cuales el procesamiento se reparte entre varios equipos conectados a traves de redes de comunicaciones. Este concepto dio paso a las bases teóricas y técnicas para la actual computación en la nube (*Espinoza Rojas, s.f.*).

Algoritmos de planificación

Dentro de los algoritmos de planificación utilizados por los sistemas operativos, según el libro “Planificación de procesos” de Gunnar Wolf encontramos:

- FIFO (FSCS): Catalogado como el algoritmo más simple de todos. Funciona bajo una lógica de cola básica, en el cual, de forma sencilla de entender, cada proceso es recibido por el procesador y no termina de trabajar en el hasta que lo finaliza (*Wolf et al., s.f.*). Este tiene el beneficio de que reduce la sobrecarga administrativa del sistema, pero los procesos cortos que van detrás de los largos son los que más se ven perjudicados. Este modelo es poco eficiente en sistemas modernos, ya que al aumentar la carga de trabajo los cuellos de botella se multiplican y no escala correctamente ante peticiones de múltiples usuarios.
- Round Robin: Este algoritmo divide a cada proceso por turnos y asigna un tiempo de trabajo máximo (quantum) para cada uno y devolviéndolo al final de la fila en caso de que no sea resuelto en este lapso (*Wolf et al., s.f.*). Su mayor ventaja es su equidad y respuesta rápida para cada proceso, pero esta depende principalmente del quantum que le haya sido asignado a cada proceso, ya que un quantum pequeño puede limitar el trabajo en los procesos. Funciona de manera eficiente en entornos interactivos ya que se puede escalar el sistema de forma optimizada para multiusuario, siempre y cuando se configure correctamente el quantum.
- SJF (SPN): Este algoritmo siempre selecciona el proceso más corto para realizar entre todos los que se encuentran en la fila de espera (*Wolf et al., s.f.*). Su principal y notable ventaja es la optimización general del rendimiento del sistema finalizando de forma rápida estos procesos cortos, pero por consecuencia los procesos largos podrían

sufrir de inanición si hay tareas cortas constantemente. Este esquema ofrece un rendimiento general mucho mayor con respecto a su eficiencia, pero es inviable al momento de querer ser escalado en sistemas de propósito general por la dificultad de predecir la duración exacta de cada proceso.

- SRTF (PSPN): Este algoritmo tiene unas características similares al SJF, con la única diferencia de que en caso de que llegue un proceso con un tiempo de solución menor al que está siendo resuelto, este se reemplaza automáticamente (*Wolf et al., s.f.*). A pesar de que es el algoritmo con menor tiempo de espera de todos, castiga con su inanición severa y un desgaste adicional al interrumpir tareas activas constantemente. Su escalabilidad y eficiencia son prácticamente nulas ya que la sobrecarga constante por cambio de procesos en trabajo y la baja predicción del tiempo de solución lo vuelve inviable en entornos de propósito general.

Resultados del simulador

El simulador utilizado para las pruebas de los algoritmos expuestos anteriormente se llama “Scheduling Algorithm Simulator” de la universidad Tennessee y se encuentra en la web.

A cada algoritmo se le asignaron 15 tareas para ver el rendimiento en distintos procesos.

Algoritmos	Tiempo medio de espera	Tiempo medio de Respuesta	Throughput (Rendimiento)	Utilización de la GPU
FCFS (Primero en llegar)	87,8	104,33	0,06	100%
Round Robin (q=4)	103,27	119,8	0,06	100%
SJF (Trabajo más corto)	55,13	71,67	0,06	100%
SRTF (Tiempo restante más corto)	54,27	70,8	0,06	100%

Tabla de autoría propia basada en los resultados del simulador.

A continuación, se presenta la gráfica comparativa entre tiempos de espera y de respuesta por cada algoritmo.

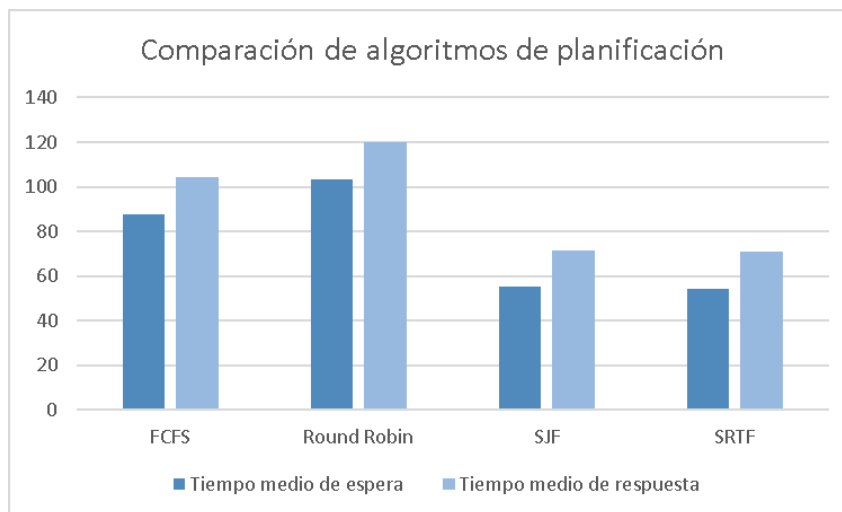


Gráfico generado con la herramienta de gráficos para Excel

Conclusiones

Elegir correctamente un algoritmo de planificación no es solo un ajuste técnico, sino una decisión que define la calidad de nuestra experiencia digital diaria al equilibrar la eficiencia del equipo con la atención justa a cada usuario. Los datos del simulador demuestran que, aunque enfocarse en las tareas más cortas agiliza el sistema al reducir los tiempos de espera, en la práctica es muy difícil predecir cuánto tardará exactamente cada proceso, lo que hace que estos modelos sean complicados de aplicar en computadoras de uso general. A medida que la tecnología ha evolucionado desde los primeros mainframes industriales hasta la flexibilidad de la nube, ha quedado en evidencia que la verdadera eficiencia no solo busca la velocidad, sino la capacidad de manejar múltiples peticiones a la vez sin crear cuellos de botella que afecten el rendimiento.

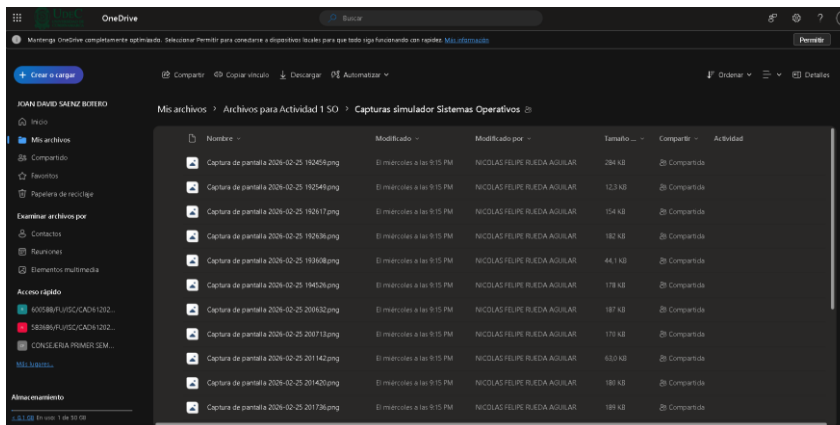
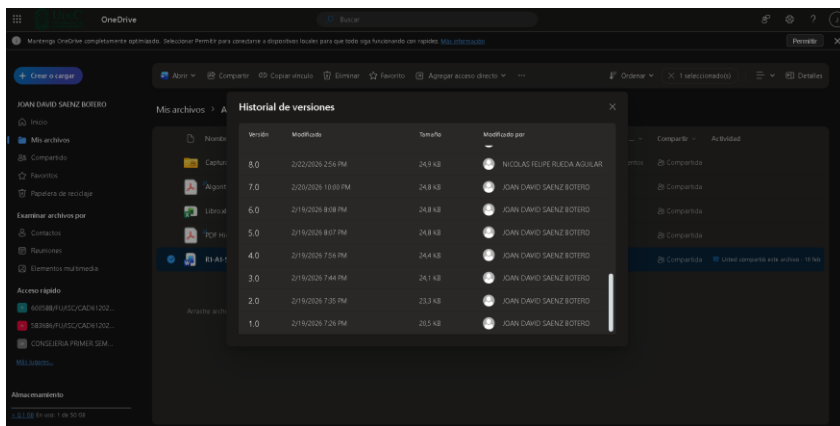
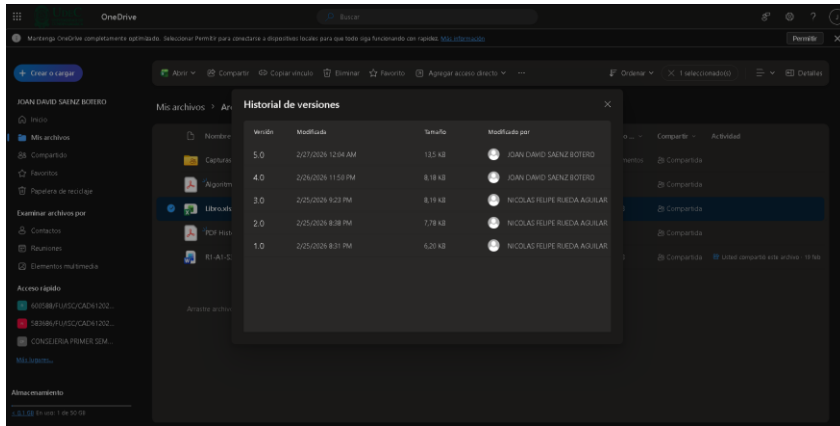
Referencias y bibliografía

Espinoza Rojas, L. (s.f.). *Evolución de los sistemas operativos desde 1940 a la actualidad*. Escuela de las Ciencias de la Computación e Informática. Recuperado de https://asteriscus.com/sistemas-operativos/espinoza_a.pdf

Wolf, G. (s.f.). *Planificación de procesos*. Universidad Nacional Autónoma de México. Recuperado de http://sistop.gwolf.org/pdf/04_planificacion_de_procesos.pdf

Anexos

Trabajo del grupo



Capturas del simulador

