

# Algoritmos de Planificación

Laura Tatiana Ramírez Rodríguez, *Cód. 20182020098*  
Julián David Pérez Chaparro, *Cód. 20192020017*  
Kevin Duvan Chaparro Torres, *Cód. 20192020038*  
Universidad Distrital Francisco José de Caldas  
Bogotá, COL

**Resumen**—El presente trabajo tiene como objetivo ofrecer una comprensión detallada de los algoritmos de planificación de sistemas operativos, incluyendo FIFO, SJF, SRTF y Round Robin. Se proporcionará una explicación de la función de cada algoritmo y cómo operan en la gestión de procesos en un sistema operativo. Además, se presentará un software de simulación que ilustra de manera práctica el funcionamiento de estos algoritmos. El estudio busca brindar una visión integral de la planificación de procesos en sistemas operativos y su importancia en el rendimiento del sistema.

**Palabras Clave**—Algoritmos de planificación, sistemas operativos, FIFO, SJF, SRTF, Round Robin, simulación, gestión de procesos.

## I. INTRODUCCIÓN

La planificación de procesos es un componente esencial en la gestión eficiente de sistemas operativos. Los algoritmos de planificación determinan cómo los procesos compiten por los recursos del sistema y afectan directamente el rendimiento general. En este contexto, este documento se centra en los algoritmos FIFO, SJF, SRTF y Round Robin, examinando sus características y comportamientos. Para ilustrar estos conceptos de manera práctica, se ha desarrollado un software de simulación que permitirá visualizar y comparar el impacto de estos algoritmos en un entorno controlado. Este enfoque práctico busca brindar a los lectores una comprensión más profunda y aplicable de los principios teóricos, facilitando así la toma de decisiones en el diseño y la implementación de sistemas operativos eficientes.

## II. OBJETIVO

El objetivo principal de este trabajo es proporcionar una comprensión clara de los algoritmos de planificación de sistemas operativos, destacando su función, operación y efectos en el rendimiento del sistema. Además, se busca ilustrar estos conceptos a través de la simulación de los algoritmos FIFO, SJF, SRTF y Round Robin en un software específico, brindando una perspectiva práctica de su funcionamiento.

## III. MARCO TEÓRICO

### III-A. Algoritmos de Planificación

Cuando un proceso es ejecutable, el sistema operativo debe decidir cuál de ellos, debe ejecutarse en primer término. Esa parte del sistema operativo que debe llevar a cabo esa decisión se llama el planificador y el algoritmo que utiliza se llama algoritmo de planificación.

En las épocas pasadas de los sistemas de procedimientos por lotes, con una entrada de forma de imágenes de tarjeta en una cinta magnética, el algoritmo de planificación era sencillo: sólo había que ejecutar el siguiente trabajo en la cinta. En los sistemas de multiusuario de tiempo compartido, que se combina en un fondo de trabajos procesados en lote, el algoritmo era más complejo. De forma invariable, existían varios usuarios en espera de servicios y podían existir otros trabajos para ser procesados en lotes, en épocas pasadas de los sistemas de procesamiento por lotes.

Existen varios criterios para los algoritmos de planificación específicos, para que sea un buen algoritmo de planificación.

- Equidad: garantizar que cada proceso obtiene su proporción justa de la CPU.
- Eficacia: mantener ocupada a la CPU el 100
- Tiempo de respuesta: minimizar el tiempo de respuesta para los usuarios interactivos.
- Tiempo de regreso: minimizar el tiempo que deben de esperar los usuarios por lotes para obtener sus resultados.
- Rendimiento: maximizar el número de tareas procesadas por hora.
- Tiempo de Llegada: Hora a la que el proceso llega a la cola de listos.
- Tiempo de finalización: Hora en la que el proceso completa su ejecución.
- Burst Time: Tiempo requerido por un proceso para la ejecución de la CPU.
- Turn Around Time: Diferencia horaria entre la hora de finalización y la hora de llegada.  $\text{Tiempo de respuesta} = \text{Hora de finalización} - \text{Hora de llegada}$
- Tiempo de espera (WT): diferencia de tiempo entre el tiempo de respuesta y el tiempo de ráfaga.  
 $\text{Tiempo de espera} = \text{tiempo de respuesta} - \text{tiempo de ráfaga}$

Los mecanismos de planificación pueden ser usados, al menos en teoría, por cualquiera de los tres tipos de planificadores. En algunas ocasiones, algunos algoritmos se adaptan mejor a las necesidades de un tipo determinado de planificador. Según que se use una disciplina determinada de planificación por un planificador a largo o corto plazo.

Se puede establecer una división importante entre los enfoques de la planificación, según que dependan de la sustitución (por derecho preferente y no derecho preferente). Al aplicarlo a la planificación a corto plazo, la no sustitución (por derecho preferente), implica que el proceso en ejecución conserva el

disfrute de los recursos asignados, incluida la CPU, hasta que dicho proceso en ejecución cede voluntariamente el control al SO. Al proceso en ejecución no se le puede forzar a renunciar a la posesión de la CPU cuando un proceso de prioridad más alta queda preparado para su ejecución. No obstante, cuando el proceso en ejecución queda en suspenso como resultado de su propia acción, por ejemplo, por esperar a la terminación de entrada/salida de datos, se puede planificar otro proceso ya preparado.

La planificación con derecho preferente, en cualquier momento, se puede sustituir un proceso en ejecución por otro de prioridad más elevada, lo cual se consigue activando el planificador, siempre que se detecta un evento que cambia el estado del sistema. Dado que dichos eventos incluyen varias acciones, además de la cesión voluntaria del control por parte del proceso en ejecución, la sustitución (por derecho preferente) hace necesaria una ejecución más fuerte que el planificador.

Una complicación que deben de enfrentar los planificadores es que cada proceso es único e impredecible. Algunos de los procesos utilizan una gran cantidad de tiempo en espera del archivo de entrada/salida mientras que otros utilizan la CPU por varias horas consecutivas si tienen la oportunidad. Cuando el planificador comienza a ejecutar algún proceso, nunca sabe a ciencia cierta cuánto tiempo transcurrirá hasta que el proceso se bloquee, ya sea para entrada/salida o debido a un semáforo, o por alguna otra razón.

Para garantizar que ningún proceso se ejecute por un tiempo excesivo, casi todas las computadoras tienen un cronómetro electrónico o un reloj incluido, que provoca una interrupción en forma periódica.

La estrategia de permitir que procesos ejecutables sean suspendidos en forma temporal se llama planificación apropiativa, en contraste con el método de ejecución hasta terminar de los primeros sistemas por lotes. La ejecución hasta terminar también recibe el nombre de planificación no apropiativa. Los algoritmos de planificación más comunes que hoy en día se conocen son:

- FCFS (First come, first served)
- SJF (Shortest Job First)
- SRTF (Shortest Remaining Time First)
- RR (Round Robin)
- Planificación por prioridad.
- Planificación con colas de niveles múltiples (MLQ, multiple level queue).
- Planificación con expropiación basada en prioridades (ED, Event Driven).

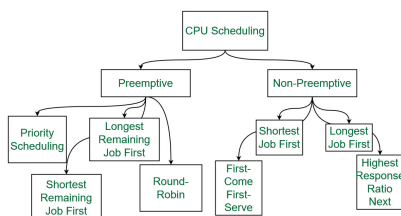


Figura 1. Tipos de Algoritmos de Planificación

**III-A1. First Come First Serve:** Se implementa una cola FIFO, el primer proceso que llega es el primero que se sirve, se atienden por orden de llegada. Es el algoritmo más sencillo, pero no garantiza buenos tiempos de respuesta. Algunas de sus características son: no es un algoritmo de planificación apropiativo, es un algoritmo predecible, los procesos largos tienen ventaja frente a los cortos y el tiempo depende del número de procesos y la duración de cada uno de ellos. Ejemplo de algoritmo de FCFS:

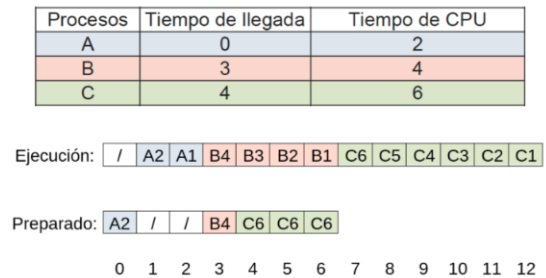


Figura 2. Ejemplo FCFS

**III-A2. Shortest Job First:** El proceso que se selecciona para ejecutar es el que tiene un menor tiempo de ejecución estimado, por lo tanto, es necesario conocer el tiempo de ejecución que va a necesitar cada proceso. La ventaja de este algoritmo de planificación es que se ejecutan muchos procesos en poco tiempo y una desventaja es que, si hay muchos procesos cortos, los procesos largos tendrán un tiempo medio de espera muy elevado. Algunas de las características de SJF; es un algoritmo de planificación no apropiativa; resulta difícil de poner en práctica porque hay que conocer de antemano los tiempos de ejecución y no siempre es fácil; es un algoritmo poco predecible; beneficia a los procesos cortos y perjudica a los largos.

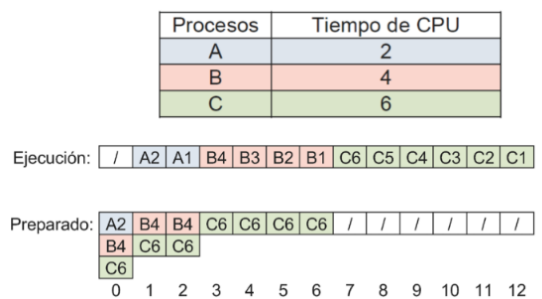


Figura 3. Ejemplo SJF

**III-A3. Shortest Remaining Time First:** El algoritmo SRTF funciona igual que el SJF, pero se le añade la expulsión de procesos. Si se está ejecutando un proceso largo y llega uno más corto, se expulsará al largo para pasar a ejecutar al corto. El mayor inconveniente de este algoritmo es que si un proceso es muy largo puede estar expulsándose muchas veces y no poder terminar nunca debido a la llegada de otros procesos más cortos. Ejemplo de SRTF:

**III-A4. Round Robin:** Se conoce también como planificación en rueda. Intenta mejorar el algoritmo FCFS y el SJF

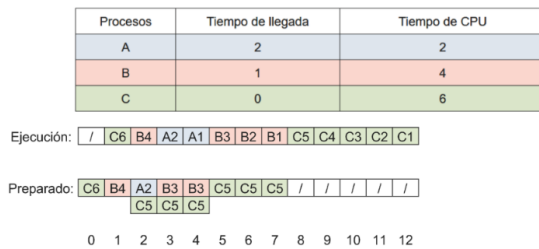


Figura 4. Ejemplo SRTF

intentado dar prioridad a los procesos, independientemente de si son largos o cortos. El algoritmo Round Robin (RR) consiste en asignar un tiempo (quantum) a cada proceso y si en este tiempo el proceso que se está ejecutando no ha terminado, vuelve de nuevo a la cola de preparados y entra en ejecución el siguiente proceso de la cola. Este algoritmo se realiza de forma repetida hasta que se ejecutan todos los procesos. El quantum asignado a los procesos depende del tipo de sistema y de las cargas que soporta y del número de procesos que hay en la cola. Características del algoritmo Round Robin es el algoritmo más utilizado para sistemas de tiempo compartido, es de planificación apropiativa, genera una sobrecarga baja puesto que los procesos siempre están en memoria si hay un cambio de contexto eficiente. Ejemplo de Round Robin suponiendo un quantum de 3:

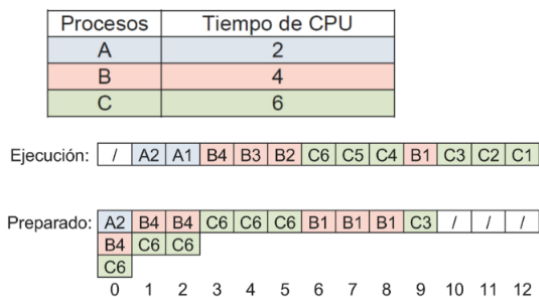


Figura 5. Ejemplo Round Robin

#### IV. DESARROLLO DE PROGRAMA Y PRUEBA DE SIMULACIÓN

##### IV-A. Código y Diseño

Se desarrolló el programa con el lenguaje Javascript, el código en su totalidad se encuentra en el siguiente repositorio : <https://github.com/Jperezchap/Simulador-Algoritmos-de-Planificaci-n.git> . El programa cuenta con un barra desplegable donde se selecciona el algoritmo de planeacion con el que quiera trabajar el usuario para ver como se gestionan los procesos. Entre los algoritmos a escoger tenemos: FCFS ,SJF ,SRTF y Round Robin.

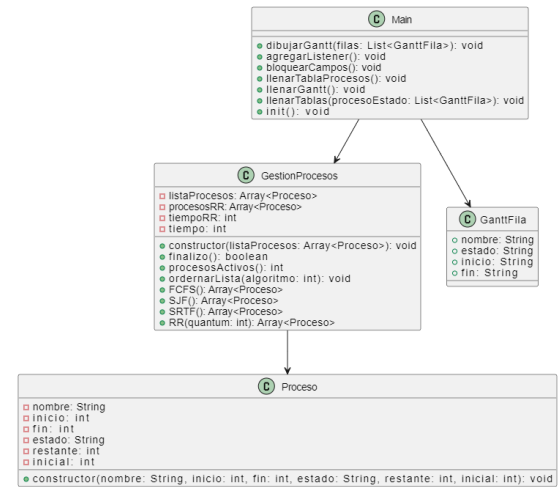


Figura 6. Diagrama de clases

El diagrama de clases modela un sistema de gestión de procesos con cuatro clases principales. La clase "Proceso" representa un proceso del sistema, almacenando información clave como nombre, tiempos de inicio y finalización, estado actual y tiempo restante. La clase "GestionProcesos" administra una lista de procesos, ofreciendo métodos para ordenarlos según diversos algoritmos de planificación y realizar el seguimiento del tiempo de ejecución. La clase "GanttFila" se utiliza para representar filas en un diagrama de Gantt, detallando la secuencia de ejecución de procesos en el tiempo. Finalmente, la clase "Main" actúa como la interfaz principal, coordinando la interacción con el usuario, dibujando el diagrama de Gantt, bloqueando campos en la interfaz y llenando tablas con información sobre procesos y estados. En términos de relaciones, la clase "Main" utiliza instancias de "GestionProcesos", "GanttFila", mientras que "GestionProcesos" emplea instancias de "Proceso" para gestionar la información detallada de cada proceso. En conjunto, estas clases colaboran para simular y visualizar la ejecución de procesos en un entorno de planificación.

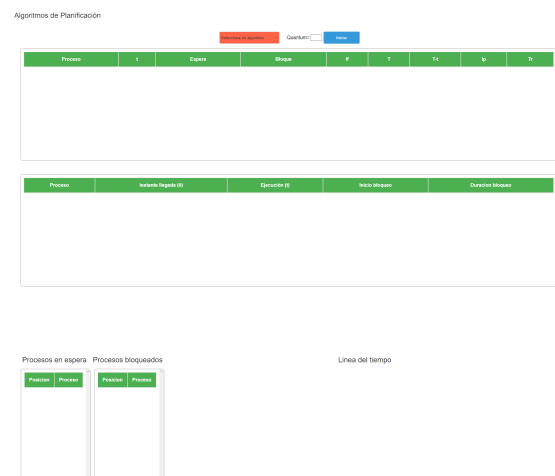


Figura 7. Interfaz de la aplicación

En la parte de la interfaz de usuario esta consiste en prin-

cipio de una barra de opciones desplegable que contiene los algoritmos de planificación que se pueden seleccionar y un boton al lado para iniciar la simulacion. Tiene una primera tabla donde se iran mostrando los procesos que van terminando con sus respectivos tiempos, la segunda tabla muestra la tabla de procesos con los datos iniciales (bloques, tiempo de ejecución, y de llegada) para iniciar la simulacion. Como ultima parte de la vista tenemos dos pequeñas tablas que nos mostraran los procesos en espera y bloqueados segun vaya pasando el tiempo y la linea del tiempo (Diagrama de Gantt) que nos mostrara graficamente por los estados por los que pasa cada proceso hasta terminar. Teniendo como caso aparte el algoritmo Round Robin que necesita un dato adicional para su ejecución, el cual debe ser ingresado en la caja de texto del Quantum ubicada a lado del boton de inicio.

#### IV-B. Simulación

En esta sección se comprobara el funcionamiento de la aplicación web, en el caso que se escogiera el algoritmo FCFS, por lo que se muestran las diferentes vistas que la componen y el resultado del escenario simulado.

**IV-B1. Algoritmos de planificación:** Se cuenta con cuatro algoritmos en principio, el usuario debe escoger para iniciar la simulación de gestión de procesos. Ya escogido se presiona "Iniciar" para comenzar la simulación.

En este caso de prueba escogeremos el algoritmo FCFS, si escogieramos el algoritmo Round Robin tenemos como condición inicial ingresar el Quantum que se dea usar, para así poder ejecutarlo.

Algoritmos de Planificación

Figura 8. Algoritmos de planeación

**IV-B2. Procesos:** Teniendo el algoritmo a usar e iniciada la simulación, en la segunda tabla aparecerán los procesos que serán gestionados, estos procesos con datos preestablecidos (los cuales podemos editar si deseamos cambiar los datos de cada proceso) se insertarán para que el algoritmo empiece a gestionar los procesos. En este caso dejaremos los datos preestablecidos.

Proceso	Instante Llegada (ti)	Ejecución (ti)	Inicio Bloqueo	Duración Bloqueo
A	5	4	2	2
B	8	7	3	3
C	7	11	1	5
D	3	17	5	5
E	2	20	9	5
F	9	18	4	5

Figura 9. Procesos Preestablecidos

**IV-B3. Paginación:** A continuación se observa la simulación del algoritmo FCFS (ver figuras 10, 10 y 11) en donde se toma el primer proceso que llega y se registra su estado de ejecución hasta bloque o tenga que entrar en espera, si ese el caso y el proceso entra en estado de bloqueo o de espera, se

va registrando estos estados en las dos tablas que acompañan a la línea del tiempo (Diagrama de Gantt).

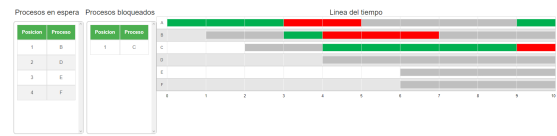


Figura 10. Algoritmo FCFS - Simulación

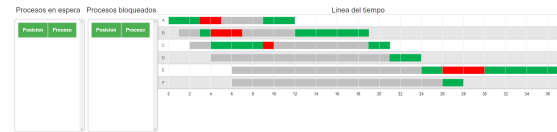


Figura 11. Algoritmo FCFS - Simulación

**IV-B4. Resultado Final:** A continuación se observa el resultado obtenido al gestionar los anteriores procesos por medio del algoritmo FCFS, se muestran los procesos en el orden en el que termino su tiempo de ejecución, con sus respectivos tiempos y valores calculados.

Proceso	t	Espera	Bloque	ti	T	Tt	tp	Tr
A	5	4	2	12	12	5	2.00	5
B	8	7	3	19	18	10	2.25	2
C	7	11	1	21	19	12	2.71	2
D	3	17	5	24	20	17	6.67	17
E	2	20	9	28	22	20	11.00	20
F	9	18	4	37	31	22	3.44	18

Figura 12. Paginación - Simulación

## V. CONCLUSIONES

Se obtienen las siguientes conclusiones:

En este estudio de los algoritmos de planificación de sistemas operativos, hemos explorado el funcionamiento y la eficiencia de los algoritmos FIFO, SJF, SRTF y Round Robin. La simulación a través de nuestro software ha proporcionado una valiosa visión práctica de cómo estos algoritmos gestionan los procesos en diversas condiciones.

En primer lugar, confirmamos que el algoritmo FIFO (First-In-First-Out) es sencillo pero puede no ser la mejor opción en situaciones donde la prioridad temporal no garantiza la eficiencia global del sistema. SJF (Shortest Job First) y SRTF (Shortest Remaining Time First) destacan por su enfoque en la minimización del tiempo de ejecución, siendo especialmente eficientes en entornos con tareas de duración variable. Sin embargo, es crucial considerar su susceptibilidad a la inanición de procesos más largos.

Por otro lado, el algoritmo Round Robin proporciona una solución equitativa al asignar un intervalo de tiempo a cada proceso, asegurando una distribución justa de recursos. Sin embargo, puede sufrir de latencias y tiempos de espera excesivos para procesos de duración variable.

En resumen, la elección del algoritmo de planificación debe basarse en las características específicas del sistema y las aplicaciones previstas. No existe un enfoque universalmente superior, ya que cada algoritmo tiene sus ventajas y desventajas.

## REFERENCIAS

- [1] “Sistemas Operativos Modernos”.Tanenbaum A.S. Pearson Edición 3ra. Accedido el 25 de noviembre de 2023. [En línea]. Disponible: <https://apps4two.com/cursodba/bibliografia/2-Sistemas/20operativos/20moderno/203ed/20Tanenbaum.pdf>
- [2] “Algoritmos de Planificación”.Instituto Consorcio Clavijero. Accedido el 25 de noviembre de 2023. [En línea]. Disponible: <https://cursos.clavijero.edu.mx/cursos/147poti/modulo2/contenidos/tema2.2.html>
- [3] “Sistemas y aplicaciones informáticas (F.P.). Volumen II”.Campos Monge E. M. , Campos Monge M. Accedido el 25 de noviembre de 2023. [En línea]. Disponible: [https://www.google.com.co/books/edition/Oposiciones\\_Cuerpo\\_de\\_Profesores\\_de\\_Ense/31W6EAAQBAJ?hl=es-419&gbpv=0](https://www.google.com.co/books/edition/Oposiciones_Cuerpo_de_Profesores_de_Ense/31W6EAAQBAJ?hl=es-419&gbpv=0)