República Bolivariana de Venezuela

Ministerio del Poder Popular para la Educación Universitaria

Universidad Rafael Urdaneta

Facultad de Ingeniería

Cátedra: Sistemas Operativos

Profesor: Mario Gonzalez

Algoritmos de Planificación de Procesos

Estudiantes:

Álvarez, Luis. C.I. 31.211.294

Maracaibo, agosto de 2025

Introducción

En los sistemas operativos multitarea, múltiples procesos compiten por recursos limitados como la CPU y dispositivos de E/S. La planificación de procesos es la función del sistema operativo que determina el orden y la duración del acceso a la CPU. El componente responsable de esta decisión es el planificador de corto plazo (también conocido como dispatcher).

Los algoritmos de planificación constituyen las estrategias que implementa el planificador para seleccionar el siguiente proceso a ejecutar desde la cola de procesos listos. Dado que el número de procesos generalmente supera la capacidad de ejecución simultánea del procesador, estos algoritmos establecen criterios que optimizan el rendimiento del sistema, mejoran los tiempos de respuesta y garantizan equidad entre los procesos.

Índice

Contenido

1. Algoritmos de Planificación

Los algoritmos de planificación representan las estrategias específicas que implementa el planificador para seleccionar el siguiente proceso a ejecutar desde la cola de procesos listos. En esencia, podemos conceptualizar un algoritmo de planificación de procesos como una técnica sofisticada utilizada por los sistemas operativos para determinar el orden prioritario en que los distintos procesos acceden al CPU. Esta necesidad surge de una realidad fundamental: el número de procesos activos casi invariablemente supera la capacidad de ejecución simultánea del procesador, creando una competencia por recursos que debe ser gestionada inteligentemente.

Para abordar este desafío, es imperativo establecer criterios de selección que optimicen el rendimiento global del sistema, mejoren los tiempos de respuesta ante las solicitudes del usuario y aseguren una distribución equitativa de recursos entre todos los procesos. Estos algoritmos constituyen elementos fundamentales en los sistemas multitarea contemporáneos, ya que determinan directamente la eficiencia con la cual se administran los recursos del hardware subyacente, actuando como el cerebro que coordina la ejecución múltiple de tareas.

2. Impacto en el Sistema y Experiencia de Usuario

La elección del algoritmo de planificación específico impacta directamente en el rendimiento global del sistema y en la experiencia percibida por el usuario. Un algoritmo bien seleccionado puede significar la diferencia entre un sistema responsivo y eficiente versus uno lento e impredecible. Esta decisión técnica representa un balance cuidadoso entre múltiples objetivos que con frecuencia entran en conflicto, requiriendo una adaptación específica según el tipo de sistema, la naturaleza de las aplicaciones y los patrones de uso predominantes. La evolución de estos algoritmos continúa siendo un área activa de investigación y desarrollo, reflejando la creciente complejidad de las cargas de trabajo modernas y las expectativas de desempeño de los usuarios contemporáneos.

3. Algoritmo Round Robin (RR)

El Round Robin es un algoritmo de planificación apropiativo que se caracteriza por su equidad en la distribución del tiempo de CPU. Su funcionamiento se basa en un quantum de tiempo, que es un intervalo fijo asignado a cada proceso. Los procesos se organizan en una cola circular, y cada uno ejecuta durante su quantum. Si un proceso no finaliza en ese tiempo, es interrumpido y reubicado al final de la cola, permitiendo que el siguiente proceso acceda a la CPU. Este enfoque garantiza que todos los procesos reciban atención periódica, evitando la inanición y ofreciendo tiempos de respuesta rápidos, ideal para sistemas interactivos o de tiempo compartido. Sin embargo, un quantum muy pequeño puede generar un alto overhead debido a los frecuentes cambios de contexto.

4. Algoritmo First In, First Out (FIFO).

El algoritmo FIFO, también conocido como First Come, First Served (FCFS), es un método no apropiativo que sigue un orden estricto de llegada. Los procesos se ejecutan en la misma secuencia en que arriban a la cola de listos, sin interrupciones hasta su finalización. Su principal ventaja radica en la simplicidad de implementación y el bajo overhead, ya que minimiza los cambios de contexto. No obstante, esta misma característica puede dar lugar al "efecto convoy", donde un proceso de larga duración retrasa a todos los procesos posteriores, afectando negativamente los tiempos de espera promedio y el rendimiento general en entornos con cargas de trabajo mixtas.

5. Algoritmo Last In, First Out (LIFO)

El algoritmo LIFO prioriza los procesos más recientes, ejecutando primero el último que llegó a la cola de listos. Al igual que FIFO, es no apropiativo, por lo que cada proceso se ejecuta hasta completarse una vez que obtiene la CPU. Este enfoque puede ser útil en contextos donde los procesos nuevos son más urgentes o presentan mayor relevancia. Sin embargo, su mayor desventaja es el alto riesgo de inanición para procesos antiguos, los cuales pueden permanecer indefinidamente en la cola si

continuamente llegan nuevos procesos. Esta falta de equidad limita su aplicabilidad en sistemas generales que requieren公平idad en la atención de procesos.

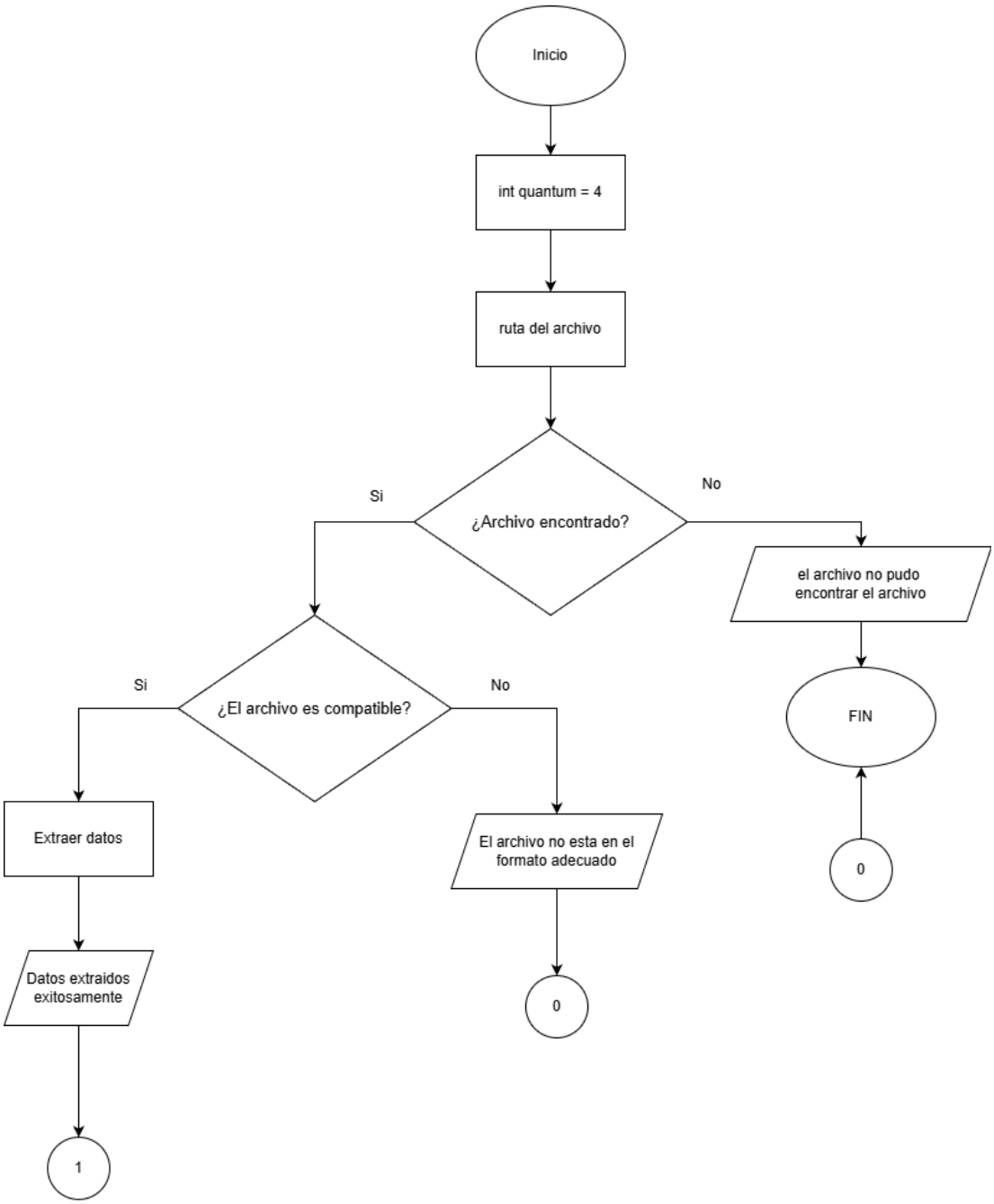
6. Proyecto sobre Algoritmos de Planificación

Se ha desarrollado un programa en Python que simula tres algoritmos de planificación de procesos: FIFO, LIFO y Round Robin. El sistema procesa una lista de actividades identificadas (A, B, C, ..., Z, A1, B1, ...) con sus respectivos tiempos iniciales (ti) y tiempos de ejecución (t). Funcionalidades principales:

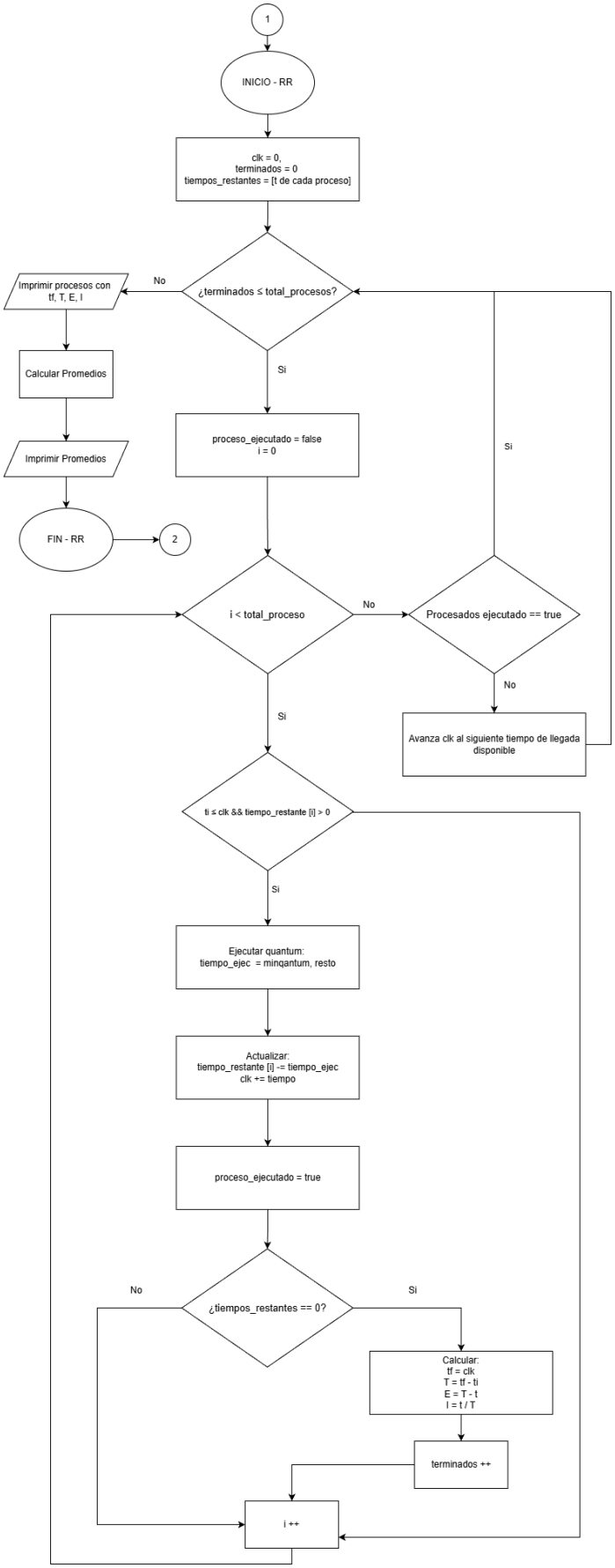
- Carga de datos desde archivo CSV con los procesos y sus tiempos
- Cálculo automático de métricas para cada proceso:
- Tiempo final (tf)
- Tiempo total (T = tf - ti)
- Tiempo de espera (E = T - t)
- Índice de servicio (I = t/T)
- Configuración de quantum para el algoritmo Round Robin
- Medición de tiempo de ejecución de cada algoritmo por separado
- Comparación automática entre los tres métodos para determinar el mejor
- Generación de reportes con tablas detalladas y promedios

El programa permite analizar el rendimiento de cada algoritmo y selecciona automáticamente el más eficiente según las métricas calculadas, proporcionando una herramienta completa para el estudio de planificación de procesos en sistemas operativos.

7. Diagrama de Flujo

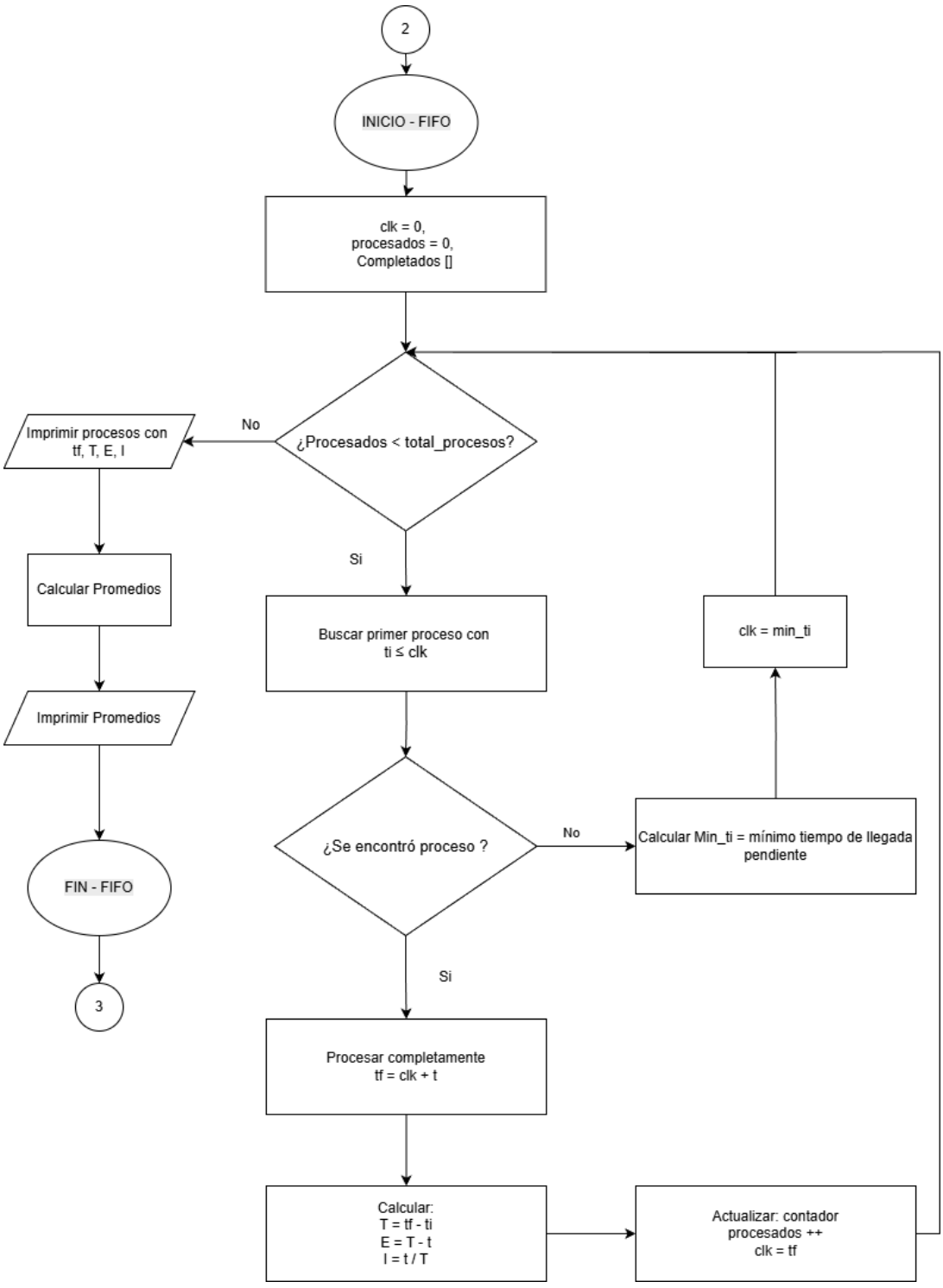


a. Diagrama de Flujo de Round Robin



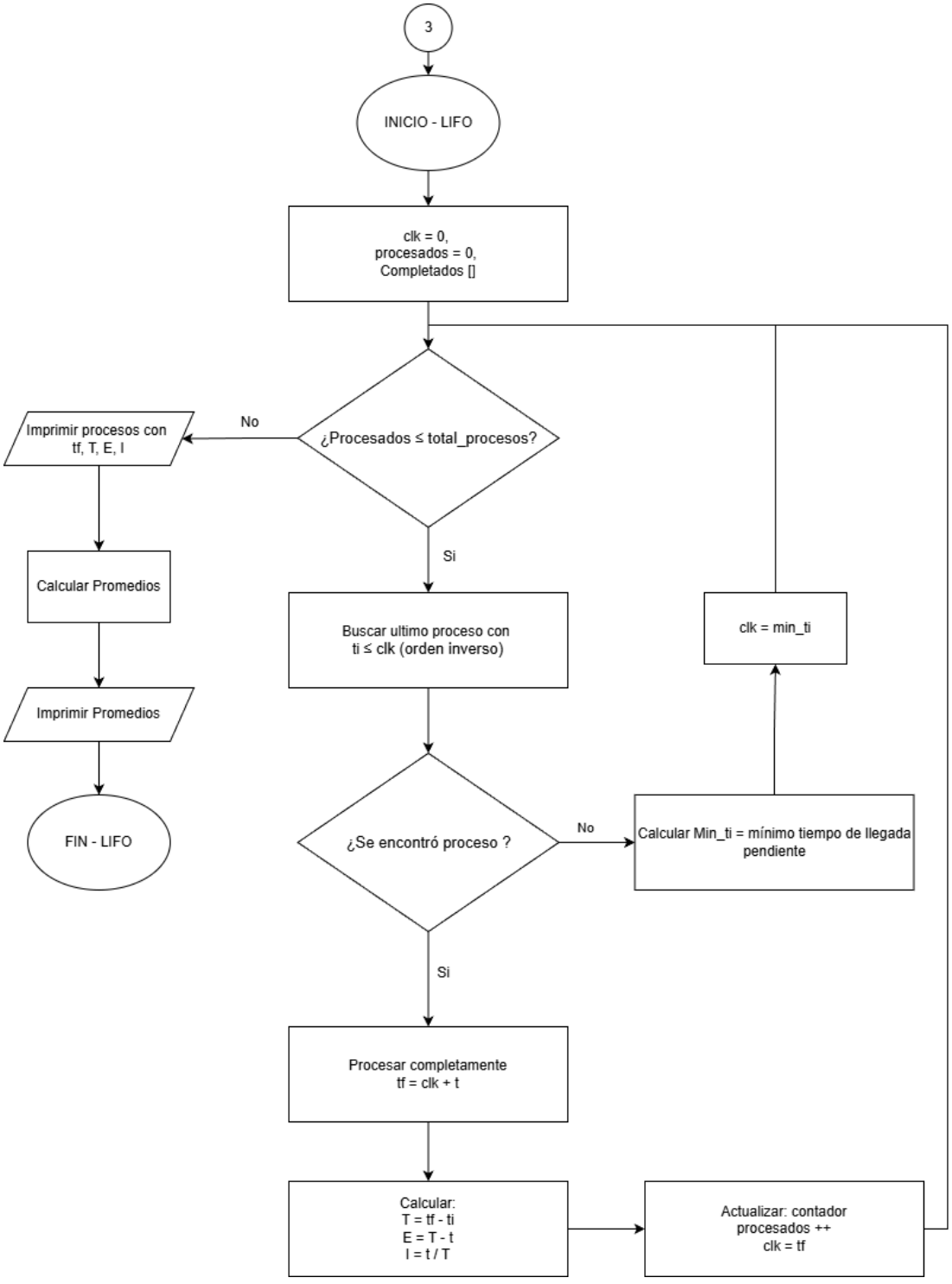
El diagrama de flujo del algoritmo Round Robin inicia con la configuración del reloj (clk) en cero, el contador de procesos terminados y un arreglo que almacena los tiempos restantes de cada proceso. El algoritmo opera mediante un bucle que continúa hasta que todos los procesos hayan finalizado. En cada iteración, recorre los procesos en orden, ejecutando cada uno disponible durante un quantum de tiempo. Si un proceso no ha terminado, se interrumpe y se recoloca en la cola, actualizando su tiempo restante. Cuando un proceso completa su ejecución, se calculan sus métricas: tiempo final (tf), tiempo total (T), tiempo de espera (E) e índice de servicio (I). Finalmente, se muestran los resultados y los promedios de las métricas.

b. Diagrama de Flujo de First In, First Out



El diagrama del algoritmo FIFO comienza inicializando el reloj (clk) y los contadores de procesos procesados. Su lógica principal busca en cada iteración el primer proceso que haya llegado (ti ≤ clk) y lo ejecuta por completo, sin interrupciones. Una vez finalizado, se calculan sus métricas: tiempo final (tf), tiempo total (T = tf - ti), tiempo de espera (E = T - t) e índice de servicio (I = t / T). Si no hay procesos disponibles, el reloj avanza al siguiente tiempo de llegada pendiente. Al terminar todos los procesos, se imprimen los resultados individuales y los promedios de las métricas calculadas.

c. Diagrama de Flujo de Last In, First Out



El flujo del algoritmo LIFO es similar al FIFO en estructura, pero difiere en la selección de procesos. Comienza con la inicialización del reloj (clk) y los contadores. En cada paso, busca el último proceso que haya llegado (en orden inverso) y que esté listo para ejecutarse (ti ≤ clk). Este proceso se ejecuta hasta su finalización, y luego se determinan sus métricas: tiempo final (tf), tiempo total (T), tiempo de espera (E) e índice de servicio (I). Si no hay procesos disponibles, el reloj avanza al menor tiempo de llegada pendiente. Al procesar todos los procesos, se muestran los resultados y los promedios de las métricas.

Conclusión

El desarrollo de este proyecto permitió implementar y analizar tres algoritmos fundamentales de planificación de procesos: Round Robin, FIFO y LIFO, demostrando cómo cada uno presenta ventajas y desventajas según el contexto de aplicación. A través de la simulación en Python, se evidenció que Round Robin ofrece la mayor equidad y tiempos de respuesta consistentes gracias a su naturaleza apropiativa, mientras que FIFO destaca por su simplicidad y bajo overhead, aunque sufre del efecto convoy. Por su parte, LIFO mostró ser útil en escenarios que priorizan los procesos más recientes, pero con un alto riesgo de inanición para procesos antiguos.

La comparación de métricas como tiempo total, tiempo de espera e índice de servicio reforzó la importancia de seleccionar el algoritmo adecuado según los requisitos del sistema, ya que incide directamente en el rendimiento y la experiencia del usuario. Este trabajo no sólo consolidó el entendimiento de los mecanismos de planificación, sino que también destacó la relevancia de optimizar el uso de recursos en entornos multitarea, proporcionando una base sólida para futuras implementaciones y mejoras en sistemas operativos modernos.