



**UNIVERSIDAD PRIVADA DE TACNA**

**FACULTAD DE INGENIERÍA**

**Escuela Profesional de Ingeniería de Sistemas**

**PROYECTO DE UNIDAD  
“SIMULADOR DE UN SISTEMA OPERATIVO”**

**Curso: Sistemas Operativos I**

**Docente: MSc. Ing. Hugo Manuel Barraza Vizcarra**

**Mamani Cori, Cristhian Carlos (2023077282)**

**Tacna – Perú  
2025**



## ÍNDICE

<b>1. Introducción</b>	<b>3</b>
1.1. Problema	3
1.2. Objetivos	3
1.3. Alcance	3
1.4. Supuestos	3
<b>2. Marco Conceptual</b>	<b>3</b>
2.1. Concepto de Procesos	3
2.2. Estados de un Proceso	3
2.3. Planificación de Procesos	3
2.4. Gestión de Memoria	3
2.5. Citas y Referencias	3
<b>3. Diseño del Simulador</b>	<b>3</b>
3.1. Estructuras de Datos Utilizadas	3
3.2. Manejo de Colas de Procesos	3
3.3. Gestión de Llegada y Servicio	3
3.4. Implementación de Algoritmos de Planificación (FCFS, SPN, Round Robin)	3
3.5. Gestión de Memoria (First-Fit y Best-Fit)	3
<b>4. Metodología de Experimentos</b>	<b>3</b>
4.1. Casos de Prueba	3
4.2. Parámetros del Simulador	3
4.3. Repetibilidad de Experimentos	3
<b>5. Resultados</b>	<b>4</b>
5.1. Tablas de Resultados por Algoritmo	4
5.2. Interpretación de Métricas Calculadas	4
<b>6. Discusión</b>	<b>4</b>
6.1. Trade-offs y Consideraciones	4
6.2. Efecto Convoy en FCFS	4
6.3. Limitaciones en SPN (SJF)	4
6.4. Elección del Quantum en Round Robin	4
<b>7. Conclusiones</b>	<b>4</b>
7.1. Síntesis de Resultados	4
7.2. Recomendaciones para Uso y Mejoras	4
<b>8. Referencias</b>	<b>4</b>



## **1. Introducción**

### **1.1. Problema**

El desarrollo de sistemas operativos implica el manejo eficiente de procesos y la correcta asignación de recursos, como la CPU y la memoria. Comprender y comparar el comportamiento de distintos algoritmos de planificación y estrategias de gestión de memoria es fundamental para optimizar el rendimiento y la respuesta del sistema. Sin embargo, en un entorno educativo, el acceso directo a sistemas operativos reales para practicar estos conceptos es limitado. Por ello, surge la necesidad de un simulador que permita experimentar y analizar estos algoritmos en un entorno controlado y didáctico.

### **1.2. Objetivos**

#### **Objetivo General**

Diseñar e implementar un simulador de sistema operativo simple que permita la experimentación y análisis de los algoritmos de planificación de procesos y gestión de memoria vistos en la Unidad 01 de la asignatura Sistemas Operativos I.

#### **Objetivos Específicos**

- Implementar algoritmos de planificación de CPU: FCFS (no expropiativo), SPN (no expropiativo) y Round Robin (expropiativo) con un quantum configurable.
- Desarrollar la estructura mínima requerida para la gestión de procesos (PCB) que incluya PID, tiempo de llegada, servicio, inicio y fin de ejecución.
- Calcular y mostrar métricas por proceso: tiempo de respuesta, espera y retorno (turnaround), así como métricas globales como promedio de estos tiempos y throughput.
- Implementar las estrategias de asignación de memoria lineal First-Fit y Best-Fit para procesos o lotes pequeños con tamaño de memoria configurable.
- Validar el funcionamiento del simulador mediante casos de prueba que permitan analizar y comparar el desempeño de los algoritmos desarrollados.

### **1.3. Alcance**

El simulador contempla la creación y manejo de procesos definidos con sus atributos esenciales (PID, tiempo de llegada, tiempo de servicio, inicio y fin de ejecución). Permite seleccionar el algoritmo de planificación de CPU y la estrategia de gestión de memoria, ajustando parámetros como el quantum para Round Robin y el tamaño total de memoria. La salida incluye tablas detalladas de ejecución con métricas por proceso y resúmenes globales. Se enfoca en simular escenarios sencillos para fines



educativos, sin incluir otros aspectos avanzados de un SO real, como entrada/salida o sincronización de procesos.

#### **1.4. Supuestos**

- Los procesos ingresados tienen tiempos de llegada y servicio predefinidos y estáticos.
- El simulador no cuenta con información futura sobre la llegada de procesos (excepto para SPN que usa información local del servicio mínimo).
- La memoria es lineal y su tamaño configurable, sin fragmentación externa avanzada ni swapping.
- El costo de cambio de contexto es considerado opcional y no es modelado en detalle, salvo que se indique específicamente.
- El quantum para Round Robin es configurable y debe ser mayor o igual a 2 para su correcto funcionamiento.
- El sistema simula un entorno cerrado y controlado, ideal para propósitos didácticos.

### **2. Marco Conceptual**

#### **2.1. Concepto de Procesos**

Un proceso es un programa en ejecución que constituye la unidad básica de trabajo en un sistema operativo. Representa un conjunto de instrucciones que se encuentran en ejecución y requieren recursos del sistema para completarse, como utilización de CPU, memoria y dispositivos de E/S (Silberschatz, Galvin, & Gagne, 2018). La gestión eficiente de procesos es crucial para la óptima operación del sistema.

#### **2.2. Estados de un Proceso**

Los procesos atraviesan diferentes estados durante su ciclo de vida: nuevo, listo, en ejecución, bloqueado y terminado. Estos estados permiten al sistema operativo administrar correctamente la asignación de recursos y la coordinación de la ejecución (Stallings, 2020). La transición entre estados depende de eventos como la asignación de CPU o la disponibilidad de recursos.

#### **2.3. Planificación de Procesos**

La planificación de procesos implica decidir qué proceso será asignado a la CPU en un momento dado. Los algoritmos de planificación pueden ser expropiativos o no expropiativos y buscan optimizar métricas como el tiempo de espera y de respuesta (Tanenbaum & Bos, 2015). Entre los algoritmos comunes se encuentran FCFS



(First-Come, First-Served), SPN (Shortest Process Next) y Round Robin, cada uno con ventajas y limitaciones específicas.

## **2.4. Gestión de Memoria**

La gestión de memoria en sistemas operativos se encarga de asignar y liberar espacios de memoria para procesos activos, asegurando un uso eficiente y evitando conflictos (Silberschatz et al., 2018). Las estrategias First-Fit y Best-Fit son técnicas para asignar bloques de memoria contiguos a procesos, diferenciándose en la manera en que seleccionan el bloque adecuado para asignación.

## **3. Diseño del Simulador**

### **3.1. Estructuras de Datos Utilizadas**

El simulador utiliza estructuras de datos esenciales para representar los procesos y el manejo de planificación. Cada proceso se modela mediante un PCB (Process Control Block) que contiene los campos mínimos: PID, llegada (tiempo de arribo), servicio (CPU total requerido), inicio (primer despacho) y fin (terminación). Para la gestión de procesos, se utilizan colas para organizar los procesos en estado listo bajo los diferentes algoritmos de planificación.

### **3.2. Manejo de Colas de Procesos**

La planificación FCFS gestiona los procesos en una cola simple en orden de llegada. SPN selecciona el proceso con menor tiempo de servicio, manteniendo la cola ordenada para prioridad. Round Robin utiliza una cola circular para alternar la asignación de CPU en quantum configurables, permitiendo la preempción.

### **3.3. Gestión de Llegada y Servicio**

El simulador controla la llegada de procesos basados en su tiempo de arribo y registra el tiempo de servicio para determinar su ejecución completa. Se registra el instante del primer despacho (inicio) y el de terminación (fin), facilitando el cálculo de las métricas de desempeño por proceso.

### **3.4. Implementación de Algoritmos de Planificación (FCFS, SPN, Round Robin)**

Se implementaron tres algoritmos representativos:

- FCFS (First-Come, First-Served): No expropiativo, los procesos se atienden en orden de llegada.



- SPN (Shortest Process Next): No expropiativo, prioriza el proceso con el menor tiempo de servicio.
- Round Robin: Expropiativo, asigna un quantum configurable para atención cíclica entre procesos.

### **3.5. Gestión de Memoria (First-Fit y Best-Fit)**

El simulador incorpora una memoria lineal con tamaño configurable (por ejemplo, 1 MiB). La asignación de bloques para procesos se realiza mediante dos estrategias: First-Fit y Best-Fit. First-Fit busca el primer bloque libre que sea suficiente para el proceso, mientras que Best-Fit busca el bloque más pequeño que pueda acomodar al proceso. La salida muestra el ID del bloque asignado y su tamaño, o un mensaje indicando que no se encontró espacio disponible.

## **4. Metodología de Experimentos**

### **4.1. Casos de Prueba**

Para validar el funcionamiento del simulador, se diseñaron varios casos de prueba con diferentes conjuntos de procesos, definidos por sus tiempos de llegada y servicio. Se incluyen escenarios con procesos con tiempos de servicio variados y superposición en tiempos de llegada para evaluar la respuesta de los algoritmos bajo diversas condiciones.

### **4.2. Parámetros del Simulador**

Los parámetros configurables incluyen la selección del algoritmo de planificación CPU (FCFS, SPN, Round Robin), el quantum para Round Robin (mínimo 2), el tamaño total de memoria lineal (ejemplo 1 MiB) y la estrategia de asignación de memoria (First-Fit o Best-Fit). Estos parámetros permiten experimentar con distintas configuraciones para observar el impacto en las métricas calculadas.

### **4.3. Repetibilidad de Experimentos**

Los experimentos se realizan en un entorno controlado, utilizando archivos de configuración JSON o ingreso por consola que aseguran la reproducibilidad de los casos de prueba. Los resultados se registran para cada ejecución y se comparan las métricas obtenidas para evaluar la eficiencia y comportamiento de los algoritmos implementados.

## 5. Resultados

### 5.1. Tablas de Resultados por Algoritmo

Asignando memoria segun estrategia: first-fit

PID 1 => asignado inicio=0, tam=120000

PID 2 => asignado inicio=120000, tam=64000

PID	Llegada	Servicio	Inicio	Fin	Respuesta	Espera	Retorno
1	0	12	0	25	0	13	25
2	1	5	4	17	3	11	16
3	2	8	8	21	6	11	19

Resumen global:

- Promedio respuesta: 3,000
- Promedio espera: 11,667
- Promedio retorno: 20,000
- Throughput: 0,120000 procesos / unidad\_tiempo (n=3, tiempo\_total=25)

Estado bloques de memoria (ordenados por inicio):

Inicio	Tam	Libre	PID
0	120000	NO	1
120000	64000	NO	2
184000	864576	SI	-1

### 5.2. Interpretación de Métricas Calculadas

- Tiempo de Respuesta: Indica el tiempo que tarda un proceso en comenzar su ejecución tras su llegada. Procesos con menor tiempo de servicio tienden a recibir respuesta más rápida en algoritmos como SPN.
- Tiempo de Espera: Se calcula restando el tiempo de servicio y llegada al tiempo de finalización, reflejando el tiempo total que un proceso estuvo esperando.
- Tiempo de Retorno (Turnaround): Tiempo total desde la llegada hasta la finalización del proceso.
- Throughput: Se calcula como la cantidad de procesos completados por unidad de tiempo total de simulación.

Los resultados reflejan que las distintas políticas afectan el orden y rapidez de atención de procesos, impactando directamente en las métricas de eficiencia y equidad del sistema.



## **6. Discusión**

### **6.1. Trade-offs y Consideraciones**

Cada algoritmo de planificación presenta ventajas y desventajas que impactan el desempeño del sistema y la experiencia del usuario. La elección adecuada depende del contexto y objetivos específicos del sistema operativo.

### **6.2. Efecto Convoy en FCFS**

El algoritmo FCFS, al atender procesos en orden de llegada sin preempción, puede causar el "efecto convoy", donde un proceso largo retrasará a todos los procesos siguientes, aumentando significativamente el tiempo de espera de los procesos cortos (Silberschatz et al., 2018).

### **6.3. Limitaciones en SPN (SJF)**

SPN o SJF, aunque minimiza el tiempo promedio de espera al priorizar procesos más cortos, asume conocimiento del tiempo de servicio requerido, lo cual raramente es perfecto en la práctica. Esta suposición limita su aplicabilidad y puede llevar a la inanición de procesos largos (Tanenbaum & Bos, 2015).

### **6.4. Elección del Quantum en Round Robin**

Round Robin es eficaz para sistemas con múltiples procesos interactivos, ofreciendo tiempos de respuesta equitativos mediante la preempción cíclica. Sin embargo, la elección del tamaño de quantum es crítica: un quantum muy pequeño incrementa el overhead por cambios de contexto, mientras uno muy grande se asemeja a FCFS, perdiendo la equidad (Stallings, 2020).

## **7. Conclusiones**

### **7.1. Síntesis de Resultados**

El simulador implementado permitió poner en práctica y comprender los conceptos fundamentales de planificación de procesos y gestión de memoria. Se evidenció cómo diferentes algoritmos afectan las métricas de desempeño, observando que FCFS puede ocasionar altos tiempos de espera debido al efecto convoy, mientras que SPN mejora el tiempo promedio de espera pero con limitaciones prácticas por la falta de conocimiento exacto del servicio. Round Robin se mostró como una alternativa equilibrada para sistemas interactivos, dependiendo fuertemente de la configuración del quantum.





## 7.2. Recomendaciones para Uso y Mejoras

Se recomienda utilizar el simulador como herramienta educativa para experimentar con diferentes configuraciones y conocer la dinámica de los algoritmos. Para futuros trabajos, se sugiere implementar el cálculo y visualización del costo por cambio de contexto, así como añadir otros algoritmos de planificación. En gestión de memoria, ampliar la simulación a técnicas avanzadas como paginación o segmentación mejoraría su alcance.

## 8. Referencias

Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). *Operating System Concepts* (10th ed.). Wiley.

<https://os.ecci.ucr.ac.cr/slides/Abraham-Silberschatz-Operating-System-Concepts-10th-2018.pdf>

Stallings, W. (2020). *Operating Systems: Internals and Design Principles* (9th ed.). Pearson.

<https://www.pearson.com/en-us/search.html?aq=operating%20systems%20internals%20and%20design%20principles>

Tanenbaum, A. S., & Bos, H. (2015). *Modern Operating Systems* (4th ed.). Pearson.

<https://www.pearson.com/en-us/search.html?aq=modern%20operating%20systems>