

# ***Informe del Proyecto: Simulador de Planificación de Procesos***

30.142.725 - Ricardo Baeta  
28.155.240 - Aida Cárdenas  
30.142.817 - René Chamorro

Caracas, 25 de Octubre de 2025

# Índice

1. Resumen	3
2. Introducción	3
3. Objetivos	4
4. Marco Teórico	4
5. Arquitectura y Diseño	5
6. Implementación	6
7. Algoritmos de Planificación Implementados	6
8. Interfaz de Usuario (GUI)	7
9. Metodología de Pruebas	7
10. Resultados y Análisis (Cualitativos)	8
11. Limitaciones del Proyecto	8
12. Trabajo Futuro	9
13. Conclusiones	9
14. Referencias	9

**Palabras Clave:** procesos; PCB; planificación; Round Robin; SJF; prioridad; MLFQ; métricas; simulación.

# 1. Resumen

Este proyecto implementa un simulador didáctico diseñado para la planificación de procesos, inspirado en sistemas operativos reales. El simulador ofrece la capacidad de crear procesos de dos tipos: ligados a CPU (CPU-bound) o ligados a E/S (I/O-bound). Permite aplicar y comparar visualmente diversos algoritmos de planificación, incluyendo FCFS, SJF/SRTF, Round Robin, Prioridad y Colas Multinivel con Retroalimentación (MLFQ).

A través de una interfaz gráfica (GUI), los usuarios pueden observar el flujo de procesos en tiempo real. El sistema recolecta métricas clave de rendimiento, tales como throughput, utilización de CPU, tiempos promedio (espera, respuesta, retorno) y equidad. La arquitectura del proyecto está organizada por capas (modelo, planificación, sistema, GUI y persistencia), diseñada con un enfoque en la claridad, extensibilidad y facilidad de experimentación.

# 2. Introducción

La planificación de procesos es un pilar fundamental en los sistemas operativos modernos. Este trabajo presenta un simulador que materializa los conceptos teóricos de la planificación, facilitando su observación práctica y permitiendo la comparación directa del rendimiento de diferentes algoritmos bajo escenarios controlados.

### 3. Objetivos

El proyecto se define por los siguientes objetivos:

- **General:** Diseñar e implementar un simulador interactivo para comprender y comparar algoritmos de planificación.
- **Específicos:**
  - Modelar los estados de un proceso y su Bloque de Control de Proceso (PCB) de forma trazable.
  - Implementar los algoritmos FCFS, SJF/SRTF, Round Robin, Prioridad y MLFQ.
  - Visualizar en tiempo real las colas de procesos, los PCBs, un log de eventos y las métricas de rendimiento.
  - Medir *throughput*, utilización de CPU, y tiempos promedio de espera, respuesta y retorno.
  - Permitir la experimentación mediante la configuración de la duración del ciclo, el *quantum*, la generación aleatoria de procesos y la persistencia de escenarios.

### 4. Marco Teórico

- **Proceso y PCB:** Un proceso es un programa en ejecución. Atraviesa varios estados (NUEVO, LISTO, EJECUCIÓN, BLOQUEADO, SUSPENDIDO, TERMINADO) y se describe mediante un PCB, que almacena sus registros y metadatos.
- **Tipos de Procesos:** Los procesos pueden ser CPU-bound (uso intensivo del procesador) o I/O-bound (pasan tiempo esperando operaciones de E/S), lo que influye en las políticas de planificación.
- **Algoritmos Implementados:**
  - **FCFS (First-Come, First-Served):** No preemptivo, atiende en orden de llegada.
  - **SJF/SRTF (Shortest Job First / Shortest Remaining Time First):** Selecciona el

proceso con menor tiempo restante.

- **Round Robin (RR):** Asigna un *quantum* (intervalo de tiempo) a cada proceso de forma rotativa.
- **Prioridad:** Asigna procesos basado en un nivel de prioridad.
- **MLFQ (Colas Multinivel con Retroalimentación):** Utiliza múltiples colas con diferentes niveles de prioridad y *quanta*.

## 5. Arquitectura y Diseño

El código del simulador se organiza en los siguientes paquetes:

- **estructuras:** Contiene listas y colas enlazadas.
- **modelo:** Define las entidades PCB, Proceso y Estado.
- **planificacion:** Incluye la interfaz del planificador y las implementaciones de los algoritmos.
- **sistema:** Contiene el núcleo (Simulador), el calculador de Métricas y el Listener de eventos.
- **gui:** Componentes de la interfaz gráfica (ventana y paneles).
- **persistencia:** Maneja el guardado y carga de datos en formato CSV.

### Flujo del Sistema:

La GUI permite al usuario crear procesos y configurar el planificador. El Simulador avanza en ciclos discretos ("ticks"), selecciona el proceso a ejecutar según el algoritmo activo, simula la E/S y emite eventos para que la GUI refresque la información visual. El componente Metricas contabiliza los ciclos y tiempos para los indicadores de desempeño.

## 6. Implementación

- **Proceso y PCB:** La clase Proceso simula el avance de instrucciones y genera excepciones periódicas de E/S. El PCB almacena datos vitales como el PC (Contador de Programa), MAR (Registro de Dirección de Memoria), estado, prioridad y los *timestamps* de tiempos (llegada, inicio, fin, espera, respuesta).
- **Simulador:** El Simulador coordina el ciclo principal: procesa eventos de E/S, selecciona un proceso, ejecuta un paso, registra métricas, maneja el agotamiento del *quantum* o la degradación de nivel (en MLFQ) y publica los eventos resultantes.
- **Persistencia:** El sistema utiliza archivos CSV para guardar y recuperar configuraciones (como la duración del ciclo) y las listas de procesos, permitiendo reproducir escenarios.

## 7. Algoritmos de Planificación Implementados

- **FCFS:** Utiliza una cola FIFO simple, sin preempción ni *quantum*.
- **SJF/SRTF:** Selecciona el proceso con el menor tiempo restante; SRTF permite preempción.
- **Round Robin:** Usa un *quantum* configurable. Al agotarse, el proceso actual es movido al final de la cola de listos.
- **Prioridad:** Elige el proceso con el número de prioridad más bajo (mayor prioridad). La selección ocurre cuando la CPU queda libre.
- **MLFQ:** Implementa tres niveles con *quanta* escalonados. Los procesos nuevos entran al nivel superior. Si un proceso no termina su ráfaga de CPU en su *quantum*, baja de nivel.

## 8. Interfaz de Usuario (GUI)

La interfaz gráfica proporciona:

- **Controles de Ejecución:** Botones para Iniciar, Pausar y Detener la simulación.
- **Gestión de Procesos:** Opciones para agregar procesos manualmente o generar 20 procesos aleatorios.
- **Configuración:** Selectores para el algoritmo de planificación y la duración del ciclo.
- **Paneles de Visualización:**
  - **Colas:** Muestra los procesos en las distintas colas.
  - **PCB:** Detalla la información del PCB del proceso actual.
  - **Log:** Un registro de eventos en tiempo real.
  - **Métricas:** Tablero con los indicadores de rendimiento.
- **Indicadores Globales:** Muestra el ciclo global, el modo (SO/Usuario) y el proceso actual en ejecución.

## 9. Metodología de Pruebas

Se define un escenario reproducible para la experimentación:

1. **Carga:** Generar 20 procesos aleatorios.
2. **Parámetros:** Fijar la duración del ciclo entre 500 y 1000 ms (para facilitar la observación).
3. **Comparación:** Ejecutar la simulación completa con FCFS, SJF/SRTF y Round Robin (probando quanta de 2, 3 y 5).
4. **Registro:** Al finalizar cada ejecución, registrar las métricas de *throughput*, utilización, tiempos promedio y equidad.

## 10. Resultados y Análisis (Cualitativos)

Las pruebas cualitativas arrojan las siguientes observaciones:

- **FCFS:** Es estable, pero sufre del "efecto convoy", donde un proceso largo puede bloquear a procesos cortos que llegaron después.
- **SJF/SRTF:** Reduce significativamente los tiempos promedio de retorno al favorecer trabajos cortos. Sin embargo, esto puede afectar la equidad.
- **Round Robin:** Mejora notablemente el tiempo de respuesta interactiva, especialmente con *quanta* pequeños. Un *quantum* demasiado pequeño puede empeorar la eficiencia por el exceso de cambios de contexto.
- **Prioridad:** Respeta la criticidad de los procesos, pero requiere políticas adicionales para evitar la inanición de procesos de baja prioridad.
- **MLFQ:** Logra un buen equilibrio entre tiempo de respuesta y *throughput* al degradar las cargas largas a colas de menor prioridad.

## 11. Limitaciones del Proyecto

- La preempción se modela de forma cooperativa (la selección ocurre al liberar la CPU).
- Las operaciones de E/S están simplificadas como un patrón de excepciones con duración fija.
- Ausencia de mecanismos de *aging* (envejecimiento) o *boosting* periódico para los algoritmos de Prioridad y MLFQ.



## 12. Trabajo Futuro

Se proponen las siguientes mejoras:

- Implementar preempción estricta mediante interrupciones temporizadas.
- Añadir *aging* y *boosting* a los algoritmos de Prioridad y MLFQ.
- Incorporar nuevas métricas (ej. p95/p99), exportación de resultados y experimentación automatizada.
- Añadir soporte para simulación de múltiples CPUs (SMP).

## 13. Conclusiones

El simulador desarrollado permite observar eficazmente, tanto visual como cuantitativamente, el impacto de distintas estrategias de planificación de procesos. La arquitectura modular y la definición de una interfaz clara para el planificador facilitan la extensión del proyecto con nuevas políticas. La combinación de la GUI, las métricas en tiempo real y la persistencia de escenarios lo convierten en una herramienta valiosa para la docencia y la experimentación.

## 14. Referencias

- Tanenbaum, A. S., & Bos, H. (2015). *Modern Operating Systems* (4th ed.). Pearson.
- Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). *Operating System Concepts* (10th ed.). Wiley.