

# Documentación del Simulador de Sistemas Operativos

Nombre: Andy Fuentes  
Curso: Sistemas Operativos  
Docentes: Juan Luis García  
Fecha de entrega: 30 de mayo de 2025

## Índice

|  |          |
|--|----------|
| <b>1. Introducción</b>                             | <b>2</b> |
| <b>2. Simulador de Calendarización</b>             | <b>2</b> |
| 2.1. Algoritmos de planificación . . . . .         | 2        |
| 2.2. Visualización dinámica . . . . .              | 2        |
| 2.3. Métricas . . . . .                            | 3        |
| <b>3. Simulador de Sincronización</b>              | <b>3</b> |
| 3.1. Modos de sincronización . . . . .             | 3        |
| 3.2. Visualización del acceso a recursos . . . . . | 4        |
| <b>4. Conclusiones</b>                             | <b>5</b> |

# 1 Introducción

Este documento describe el funcionamiento de una aplicación visual desarrollada en Python utilizando las librerías Streamlit y Plotly. El simulador permite ilustrar dinámicamente dos componentes fundamentales en los sistemas operativos: la **calendarización de procesos** y la **sincronización por acceso a recursos compartidos** usando mecanismos como **mutex** y **semáforos**.

## 2 Simulador de Calendarización

### 2.1. Algoritmos de planificación

El simulador soporta los siguientes algoritmos de planificación:

- FIFO (First In First Out)
- SJF (Shortest Job First)
- SRT (Shortest Remaining Time)
- RR (Round Robin, con quantum configurable)
- Priority Scheduling

### 2.2. Visualización dinámica

La visualización de la planificación se realiza mediante diagramas de Gantt generados en tiempo real. Cada bloque representa un proceso ejecutándose en un determinado ciclo.

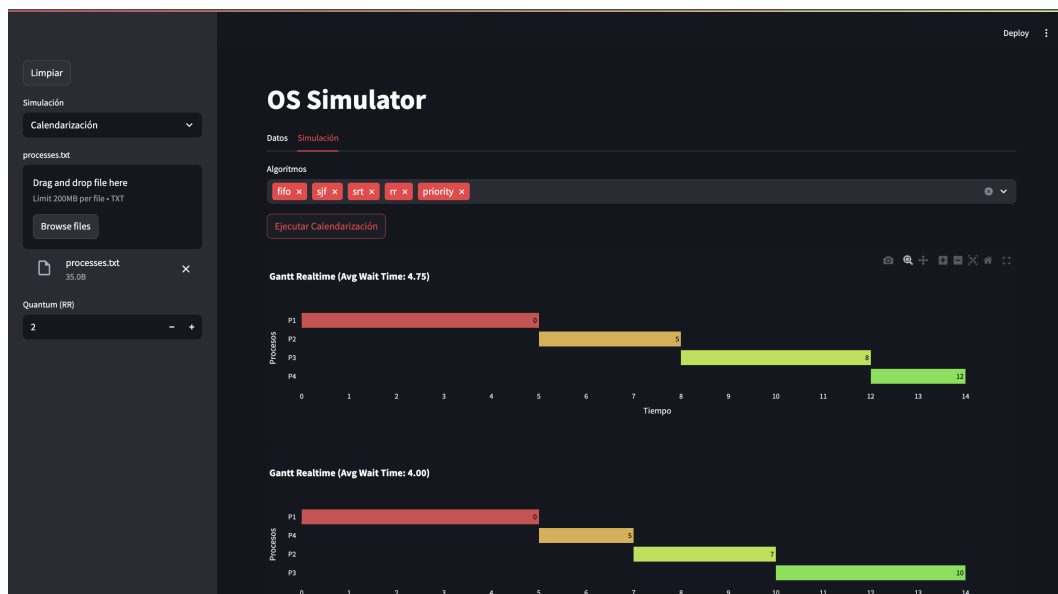


Figura 1: Planificación FIFO, SJF y SRT



Figura 2: Planificación RR (Quantum = 1)

|   | Algoritmo | AvgWaitingTime |
|---|-----------|----------------|
| 0 | fifo      | 4.7500         |
| 1 | sjf       | 4.0000         |
| 2 | srt       | 3.5000         |
| 3 | rr        | 7.0000         |
| 4 | priority  | 4.7500         |

Figura 3: Planificación Priority Scheduling

## 2.3. Métricas

Se calcula automáticamente el **Average Waiting Time** (tiempo de espera promedio) para cada algoritmo, permitiendo comparar su eficiencia bajo distintos escenarios.

## 3 Simulador de Sincronización

### 3.1. Modos de sincronización

Se implementan los siguientes mecanismos:

- **Mutex:** permite un solo acceso al recurso compartido.
- **Semáforo:** permite múltiples accesos simultáneos si el contador lo permite.

### 3.2. Visualización del acceso a recursos

El acceso a recursos se representa por estados:

- **ACCESSED (verde):** acceso exitoso al recurso.
- **WAITING (rojo):** el recurso no estaba disponible.



Figura 4: Visualización con semáforos

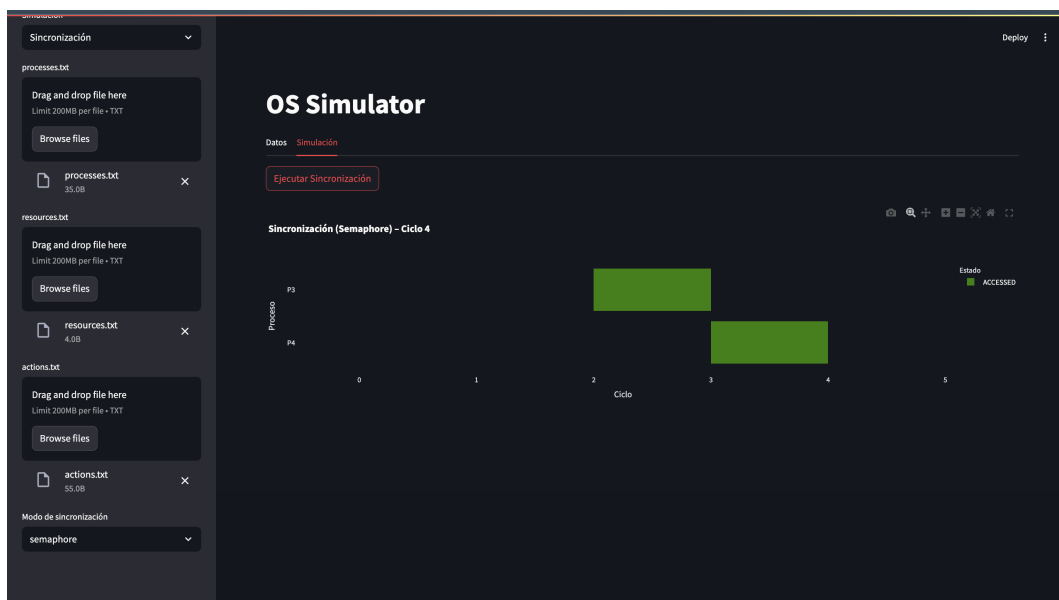


Figura 5: Visualización con mutex

## Repositorio

**GitHub:** <https://github.com/Andyfer004/P2-S0>

## 4 Conclusiones

Este simulador permite observar de manera intuitiva cómo se comportan distintos algoritmos de planificación y mecanismos de sincronización. Refuerza el entendimiento práctico sobre el manejo de procesos, tiempos de ejecución y acceso a recursos críticos en sistemas operativos.