

UNIVERSIDAD PEDAGÓGICA Y TECNOLÓGICA DE
COLOMBIA

SECCIONAL SOGAMOSO

Facultad de Ingeniería

Escuela de Ingeniería de Sistemas y Computación

Simulador de Sistema Operativo

Gestión de Procesos, Memoria y Planificación

Manual de Usuario

Versión 1.0

Proyecto Final de Sistemas Operativos

Presentado por:

Andryw Yesid Barrera Camargo
Henry Leonardo Rodriguez Paez

Asignatura:

Sistemas Operativos
Período Académico: 2025-2

Sogamoso, Boyacá
17 de noviembre de 2025

Resumen

Este manual describe el uso del **Simulador de Sistema Operativo**, una herramienta educativa con interfaz gráfica que permite visualizar y comprender los conceptos fundamentales de:

- Planificación de procesos (RR, SJF, Priority)
- Gestión de memoria con paginación
- Sistema de archivos con control de concurrencia

Índice

1. Introducción	3
1.1. Objetivo del Simulador	3
1.2. Requisitos del Sistema	3
2. Descripción de la Interfaz	3
2.1. Distribución General	3
3. Parámetros de Entrada	4
3.1. Configuración de Planificación	4
3.1.1. Algoritmos de Planificación	4
3.2. Valores Recomendados por Escenario	5
4. Datos de Salida y Visualización	6
4.1. Pestaña 1: Procesos	6
4.1.1. Gráfico de Barras	6
4.1.2. Tabla de Procesos	6
4.2. Pestaña 2: Memoria	6
4.2.1. Visualización de Frames	6
4.2.2. Algoritmo de Reemplazo	7
4.3. Pestaña 3: Archivos	7
4.3.1. Log de Accesos	7
5. Métricas del Sistema	8
5.1. Panel de Métricas en Tiempo Real	8
5.2. Fórmulas de Cálculo	8
6. Ejemplos de Uso	9
6.1. Ejemplo 1: Comparación de Algoritmos	9
6.2. Ejemplo 2: Análisis de Memoria	9
6.3. Ejemplo 3: Conflictos de Archivos	10
7. Guía de Operación Paso a Paso	11
7.1. Inicio de Simulación	11
7.2. Interpretación de Resultados	11
8. Solución de Problemas	12
8.1. Problemas Comunes	12
9. Glosario de Términos	12
10. Conclusión	13

1. Introducción

1.1. Objetivo del Simulador

El simulador permite comprender de forma visual e interactiva cómo un sistema operativo gestiona múltiples procesos, memoria y archivos simultáneamente. Es ideal para estudiantes y profesionales que deseen reforzar conceptos teóricos con práctica visual.

1.2. Requisitos del Sistema

- Python 3.7 o superior
- Bibliotecas: `tkinter` (incluida en Python)
- Opcional: `customtkinter` para interfaz mejorada

Instalación opcional:

```
pip install customtkinter
```

2. Descripción de la Interfaz

2.1. Distribución General

La interfaz está dividida en dos secciones principales:

Sección	Descripción
Panel Izquierdo	Controles de configuración, botones de control y métricas en tiempo real
Panel Derecho	Visualización en 3 pestañas: Procesos, Memoria y Archivos

Tabla 1: Distribución de la interfaz principal

3. Parámetros de Entrada

3.1. Configuración de Planificación

Parámetro	Valores	Descripción
Algoritmo	RR / SJF / PRIORITY	Algoritmo de planificación de CPU
Nº Procesos	1 - 20	Cantidad de procesos a simular
Quantum	1 - 10	Tiempo de CPU para Round Robin (unidades)
Frames Memoria	5 - 20	Marcos de página disponibles

Tabla 2: Parámetros de configuración del simulador

3.1.1. Algoritmos de Planificación

Round Robin (RR)

Los procesos se ejecutan por turnos. Cada uno recibe un tiempo máximo (*quantum*) antes de pasar al siguiente proceso en la cola.

- **Ventaja:** Equitativo, todos los procesos avanzan
- **Uso:** Sistemas interactivos y de tiempo compartido

Shortest Job First (SJF)

Ejecuta primero el proceso con menor tiempo de ejecución restante.

- **Ventaja:** Minimiza tiempo promedio de espera
- **Desventaja:** Puede causar inanición en procesos largos

Priority Scheduling

Ejecuta procesos según su prioridad (1=baja, 10=alta).

- **Ventaja:** Tareas críticas se ejecutan primero
- **Uso:** Sistemas en tiempo real

3.2. Valores Recomendados por Escenario

Escenario	Algoritmo	Procesos	Quantum	Frames
Prueba básica	RR	5	2	10
Carga media	SJF	8	3	15
Carga alta	PRIORITY	15	1	10
Análisis memoria	RR	10	2	8

Tabla 3: Configuraciones recomendadas según escenario

4. Datos de Salida y Visualización

4.1. Pestaña 1: Procesos

4.1.1. Gráfico de Barras

Muestra el progreso de ejecución de cada proceso mediante barras de colores:

- Verde/Azul/Rojo: Proceso activo
- Gris: Proceso terminado
- Altura de barra = Porcentaje completado

4.1.2. Tabla de Procesos

PID	Estado	Prioridad	Burst	Restante	Espera
0	EJECUTANDO	7	10	3	2.50
1	LISTO	5	8	8	0.00
2	ESPERANDO	9	12	6	4.20
3	TERMINADO	4	6	0	1.80
4	LISTO	8	15	15	0.00

Tabla 4: Ejemplo de tabla de procesos durante la simulación

Descripción de columnas:

- **PID:** Identificador único del proceso (0, 1, 2, ...)
- **Estado:** NUEVO, LISTO, EJECUTANDO, ESPERANDO, TERMINADO
- **Prioridad:** Valor entre 1 (baja) y 10 (alta)
- **Burst:** Tiempo total de ejecución requerido
- **Restante:** Tiempo de CPU que falta por ejecutar
- **Espera:** Tiempo total esperado en cola

4.2. Pestaña 2: Memoria

4.2.1. Visualización de Frames

Frames de Memoria (Ejemplo con 10 frames)				
P0-Pg3	P1-Pg7	Vacío	P2-Pg12	P0-Pg5
Vacío	P3-Pg9	P1-Pg14	Vacío	P2-Pg2

Tabla 5: Representación visual de frames de memoria

Leyenda:

- Azul: Frame ocupado (P#=Proceso, Pg#=Página)
- Gris: Frame disponible

4.2.2. Algoritmo de Reemplazo

El simulador utiliza **LRU (Least Recently Used)** por defecto:

- Reemplaza la página menos recientemente usada
- Minimiza fallos de página en accesos con localidad temporal

4.3. Pestaña 3: Archivos

4.3.1. Log de Accesos

Estado	Proceso	Archivo	Resultado
✓	P2	archivo1.txt	SUCCESS
×	P5	archivo2.txt	CONFLICT
✓	P1	archivo3.txt	SUCCESS
✓	P3	archivo1.txt	SUCCESS
×	P4	archivo2.txt	CONFLICT

Tabla 6: Ejemplo de registro de acceso a archivos

Interpretación:

- ✓ **SUCCESS:** Acceso exitoso (archivo disponible)
- × **CONFLICT:** Conflicto de concurrencia (archivo bloqueado)

5. Métricas del Sistema

5.1. Panel de Métricas en Tiempo Real

Métrica	Ejemplo	Significado
Procesos Completados	5/10	Procesos terminados de total
Tiempo Promedio Espera	3.45	Tiempo medio en cola (unidades)
Tiempo Promedio Retorno	12.80	Tiempo desde llegada hasta fin
Uso de Memoria	75.0 %	Porcentaje de frames ocupados
Page Faults	45	Número de fallos de página
Page Hits	123	Número de aciertos de página
Conflictos de Archivos	8	Intentos fallidos de acceso

Tabla 7: Métricas del sistema y su interpretación

5.2. Fórmulas de Cálculo

Tiempo de Espera (Waiting Time):

$$T_{espera} = T_{finalizacion} - T_{llegada} - T_{burst}$$

Tiempo de Retorno (Turnaround Time):

$$T_{retorno} = T_{finalizacion} - T_{llegada}$$

Uso de Memoria:

$$Uso \% = \frac{Frames\ Ocupados}{Total\ Frames} \times 100$$

Tasa de Aciertos de Página:

$$Hit\ Rate = \frac{Page\ Hits}{Page\ Hits + Page\ Faults} \times 100$$

6. Ejemplos de Uso

6.1. Ejemplo 1: Comparación de Algoritmos

Objetivo: Comparar el desempeño de RR vs SJF con la misma carga.

Configuración:

- Número de procesos: 10
- Quantum: 2
- Frames: 10

Procedimiento:

1. Ejecutar con algoritmo RR y anotar métricas
2. Reiniciar simulador
3. Ejecutar con algoritmo SJF y anotar métricas
4. Comparar tiempos de espera promedio

Resultados Esperados:

Métrica	RR	SJF
Tiempo Promedio Espera	5.80	3.20
Tiempo Promedio Retorno	14.50	11.40

Tabla 8: Comparación RR vs SJF

Conclusión: SJF generalmente produce menores tiempos de espera, pero puede causar inanición en procesos largos.

6.2. Ejemplo 2: Análisis de Memoria

Objetivo: Observar el impacto del número de frames en los page faults.

Configuración Base:

- Algoritmo: RR
- Procesos: 8
- Quantum: 2

Experimento:

Prueba	Frames	Page Faults	Hit Rate
1	5	78	45 %
2	10	52	63 %
3	15	34	76 %

Tabla 9: Impacto del número de frames en rendimiento

Conclusión: A mayor cantidad de frames, menos page faults y mejor rendimiento de memoria.

6.3. Ejemplo 3: Conflictos de Archivos

Objetivo: Analizar la concurrencia en acceso a archivos.

Configuración:

- Algoritmo: RR
- Procesos: 15 (alta concurrencia)
- Quantum: 1
- Frames: 10

Observaciones Esperadas:

- Mayor número de procesos → Más conflictos de archivos
- Archivos populares (archivo1.txt) tendrán más conflictos
- Los conflictos generan retrasos en los procesos

7. Guía de Operación Paso a Paso

7.1. Inicio de Simulación

Paso 1: Ejecutar el programa Python:

```
python simulador_so.py
```

Paso 2: Configurar parámetros en el panel izquierdo:

- Seleccionar algoritmo de planificación
- Ingresar número de procesos (ej: 8)
- Ajustar quantum si usa RR (ej: 2)
- Definir frames de memoria (ej: 10)

Paso 3: Presionar el botón **→ INICIAR**

Paso 4: Observar la simulación en las 3 pestañas

Paso 5: Usar **||PAUSAR** para analizar un momento específico

Paso 6: Presionar **■ DETENER** al finalizar

7.2. Interpretación de Resultados

Bueno

- Tiempo promedio de espera < 5 unidades
- Uso de memoria entre 60-80 %
- Hit rate de página > 70 %
- Pocos conflictos de archivos (< 10)

Requiere Atención

- Tiempo promedio de espera > 10 unidades
- Uso de memoria > 95 % (sobrecarga)
- Hit rate de página < 50 %
- Muchos conflictos de archivos (> 20)

8. Solución de Problemas

8.1. Problemas Comunes

Problema	Solución
La interfaz no se ve moderna	Instalar customtkinter: <code>pip install customtkinter</code>
Simulación muy rápida	Aumentar el <code>time.sleep()</code> en el código (línea 621)
Muchos page faults	Aumentar número de frames de memoria
Procesos muy lentos	Reducir número de procesos o usar algoritmo SJF
Error al iniciar	Verificar que todos los valores sean numéricos válidos

Tabla 10: Problemas comunes y sus soluciones

9. Glosario de Términos

Definiciones Clave

Burst Time	Tiempo total de CPU que requiere un proceso para completarse
Frame	Marco de página en memoria física
Page Fault	Fallo al acceder una página que no está en memoria
Page Hit	Acceso exitoso a una página ya cargada en memoria
Quantum	Intervalo de tiempo asignado a cada proceso en Round Robin
Waiting Time	Tiempo que un proceso pasa esperando en la cola de listos
Turnaround Time	Tiempo total desde que llega un proceso hasta que termina
LRU	Least Recently Used - Algoritmo que reemplaza la página menos usada

10. Conclusión

Este simulador es una herramienta educativa completa que permite:

- **Visualizar** conceptos abstractos de sistemas operativos
- **Experimentar** con diferentes algoritmos y configuraciones
- **Comparar** desempeño de estrategias de planificación
- **Analizar** gestión de memoria y concurrencia

Recomendaciones Finales

1. Comience con configuraciones simples (5 procesos, algoritmo RR)
2. Experimente cambiando un parámetro a la vez
3. Anote y compare métricas entre ejecuciones
4. Use la pausa para analizar estados específicos
5. Pruebe escenarios extremos (muchos procesos, pocos frames)

Para más información, consulte la documentación del código fuente o contacte al equipo de desarrollo.