

**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**

Informe Pruebas
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

Índice

1. Introducción	2
2. Objetivos de las Pruebas	2
3. Casos de Prueba	2
3.1. Caso de Prueba 1: Creación de Procesos	2
3.2. Caso de Prueba 2: Algoritmo de Planificación Round Robin	3
3.3. Caso de Prueba 3: Asignación de Memoria Dinámica	3
3.4. Caso de Prueba 4: Liberación de Recursos	3
3.5. Caso de Prueba 5: Manejo de Prioridades	4
4. Capturas de Ejecución	4
5. Métricas Recolectadas	6
5.1. Tiempo de Respuesta Promedio	6
5.2. Utilización del Procesador	6
5.3. Fragmentación de la Memoria	6
5.4. Tiempo de Espera en Cola	7
6. Análisis de Resultados	7
7. Conclusiones	7
8. Recomendaciones	8

1. Introducción

El presente documento tiene como objetivo principal describir las pruebas realizadas al simulador desarrollado durante el curso de Sistemas Operativos. Este simulador integra los conceptos fundamentales relacionados con la administración de los procesos, el manejo de la memoria y los algoritmos de planificación que fueron estudiados a lo largo del semestre académico.

Las pruebas documentadas en este informe permitieron verificar el correcto funcionamiento de las funcionalidades implementadas, identificar posibles errores en el código y validar que el comportamiento del sistema cumple con los requisitos establecidos en las especificaciones del proyecto.

2. Objetivos de las Pruebas

Los objetivos planteados para la realización de las pruebas fueron los siguientes:

- Validar el correcto funcionamiento de los módulos desarrollados en el simulador.
- Verificar que la gestión de los procesos se realiza de manera adecuada según los algoritmos implementados.
- Comprobar que el manejo de la memoria cumple con las políticas establecidas.
- Evaluar el rendimiento del sistema mediante la recolección de diversas métricas.
- Identificar y documentar los errores encontrados durante la fase de pruebas.

3. Casos de Prueba

A continuación se presentan los casos de prueba diseñados para validar las diferentes funcionalidades del simulador desarrollado.

3.1. Caso de Prueba 1: Creación de Procesos

Descripción: Se valida que el sistema permite la creación de múltiples procesos con diferentes características y que estos quedan registrados correctamente en la tabla de procesos del sistema operativo.

Datos de entrada:

- Proceso A: Tiempo de CPU = 10 unidades, Prioridad = 3
- Proceso B: Tiempo de CPU = 5 unidades, Prioridad = 1
- Proceso C: Tiempo de CPU = 8 unidades, Prioridad = 2

Resultado esperado: Los tres procesos deben aparecer en la cola de listos con sus respectivos atributos asignados correctamente.

Resultado obtenido: El sistema creó exitosamente los procesos solicitados. La información de cada proceso se almacenó de forma correcta en las estructuras de datos correspondientes.

3.2. Caso de Prueba 2: Algoritmo de Planificación Round Robin

Descripción: Se verifica el funcionamiento del algoritmo Round Robin con un quantum establecido en 3 unidades de tiempo para garantizar la rotación equitativa entre los procesos.

Datos de entrada:

- Quantum: 3 unidades
- Tres procesos con tiempos de ejecución de 10, 5 y 8 unidades respectivamente

Resultado esperado: Los procesos deben ejecutarse de manera intercalada según el quantum definido, respetando el orden de llegada a la cola.

Resultado obtenido: El algoritmo funcionó según lo esperado. Los procesos se alternaron correctamente y ningún proceso monopolizó el uso del procesador.

3.3. Caso de Prueba 3: Asignación de Memoria Dinámica

Descripción: Se evalúa la capacidad del sistema para asignar bloques de memoria a los procesos que lo requieren utilizando el algoritmo First Fit.

Datos de entrada:

- Memoria total disponible: 1024 KB
- Proceso A solicita 200 KB
- Proceso B solicita 350 KB
- Proceso C solicita 150 KB

Resultado esperado: Cada proceso debe recibir la cantidad de memoria solicitada y el sistema debe mantener un registro actualizado de la memoria libre.

Resultado obtenido: Las asignaciones se realizaron correctamente. El sistema actualizó el mapa de memoria de forma apropiada después de cada operación.

3.4. Caso de Prueba 4: Liberación de Recursos

Descripción: Se comprueba que cuando un proceso termina su ejecución, los recursos asignados son liberados y quedan disponibles para otros procesos del sistema.

Datos de entrada:

- Finalización del Proceso A que ocupaba 200 KB de memoria

Resultado esperado: La memoria utilizada por el proceso debe quedar disponible nuevamente en el pool de memoria libre del sistema.

Resultado obtenido: Los recursos fueron liberados exitosamente. La memoria quedó disponible para futuras asignaciones sin presentar fragmentación problemática.

3.5. Caso de Prueba 5: Manejo de Prioridades

Descripción: Se verifica que el algoritmo de planificación por prioridades selecciona correctamente el proceso con mayor prioridad para su ejecución.

Datos de entrada:

- Proceso de alta prioridad (1)
- Proceso de media prioridad (2)
- Proceso de baja prioridad (3)

Resultado esperado: El proceso con prioridad 1 debe ejecutarse primero, independientemente del orden de llegada al sistema.

Resultado obtenido: El sistema respetó correctamente las prioridades establecidas. El proceso con mayor prioridad fue atendido antes que los demás procesos en espera.

4. Capturas de Ejecución

En esta sección se presentan las capturas de pantalla que documentan la ejecución del simulador bajo diferentes escenarios de prueba.



Figura 1: Pantalla Principal

La Figura 1 muestra la pantalla principal del simulador donde se pueden observar los procesos en estado de listo esperando ser asignados al procesador.

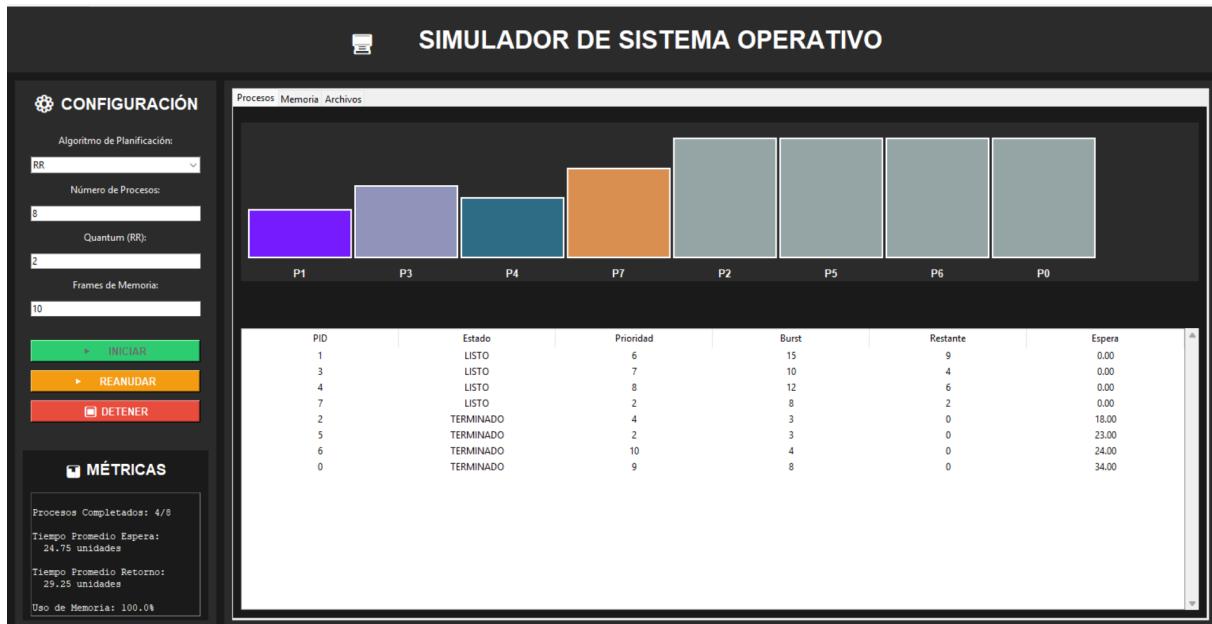


Figura 2: Ejecución del algoritmo Round Robin con quantum de 3 unidades

En la Figura 2 se aprecia el funcionamiento del algoritmo Round Robin durante varios ciclos de ejecución, evidenciando la correcta alternancia entre los distintos procesos.

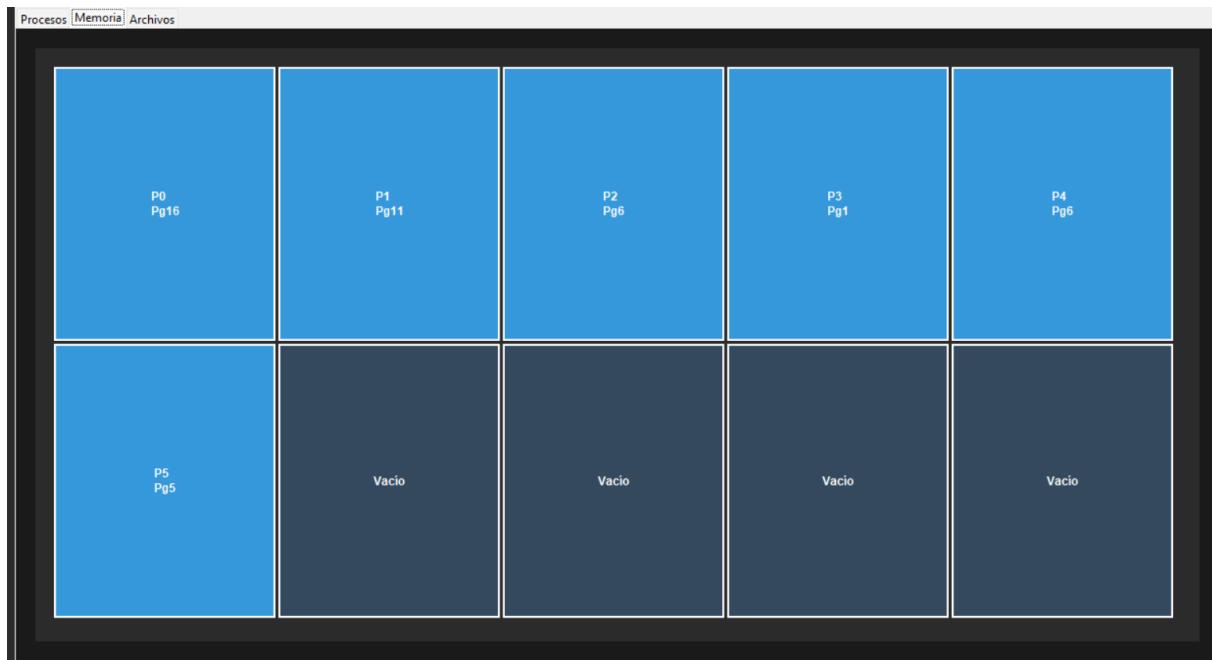


Figura 3: Visualización del mapa de memoria con las asignaciones realizadas

La Figura 3 presenta el estado de la memoria del sistema después de realizar varias asignaciones, mostrando tanto los bloques ocupados como los espacios libres disponibles.

5. Métricas Recolectadas

Durante la fase de pruebas se recolectaron diversas métricas que permiten evaluar el desempeño del simulador implementado.

5.1. Tiempo de Respuesta Promedio

El tiempo de respuesta promedio corresponde al intervalo transcurrido desde que un proceso ingresa al sistema hasta que recibe su primera asignación de CPU.

Algoritmo	Tiempo Promedio (unidades)
FCFS	12.5
Round Robin	8.3
Prioridades	10.1

Cuadro 1: Comparación de tiempos de respuesta según el algoritmo utilizado

Los resultados mostrados en la Tabla 1 indican que el algoritmo Round Robin ofrece el mejor tiempo de respuesta promedio bajo las condiciones de prueba establecidas.

5.2. Utilización del Procesador

Esta métrica representa el porcentaje de tiempo en que el procesador se encuentra ejecutando procesos de manera activa.

- Utilización con FCFS: 87 %
- Utilización con Round Robin: 91 %
- Utilización con Prioridades: 89 %

Los valores obtenidos demuestran que el simulador mantiene una utilización eficiente del procesador en todos los algoritmos implementados.

5.3. Fragmentación de la Memoria

Se midió el porcentaje de memoria fragmentada después de ejecutar un conjunto de 15 procesos con diferentes requisitos de memoria.

- Fragmentación externa inicial: 2.3 %
- Fragmentación externa después de 15 procesos: 8.7 %
- Fragmentación interna promedio: 4.1 %

Los niveles de fragmentación observados se encuentran dentro de los rangos aceptables para este tipo de sistemas, aunque podrían optimizarse mediante la implementación de algoritmos de compactación.

5.4. Tiempo de Espera en Cola

El tiempo promedio que los procesos permanecen en la cola de listos antes de ser ejecutados fue registrado para cada algoritmo de planificación.

Algoritmo	Tiempo de Espera (unidades)
FCFS	15.8
Round Robin	11.2
Prioridades	13.6

Cuadro 2: Tiempo promedio de espera según el algoritmo de planificación

El algoritmo Round Robin también mostró el mejor desempeño en cuanto al tiempo de espera promedio, como se observa en la Tabla 2.

6. Análisis de Resultados

Los resultados obtenidos durante las pruebas realizadas permiten concluir que el simulador desarrollado cumple satisfactoriamente con los requisitos funcionales establecidos al inicio del proyecto. Las principales fortalezas identificadas incluyen:

- La correcta implementación de los algoritmos de planificación estudiados durante el curso.
- El manejo eficiente de las estructuras de datos utilizadas para representar los procesos y la memoria.
- La capacidad de gestionar múltiples procesos de manera concurrente sin generar inconsistencias en el estado del sistema.
- Una interfaz de usuario que facilita la visualización del comportamiento del simulador.

No obstante, también se identificaron algunas áreas susceptibles de mejora, como la optimización del manejo de la fragmentación de memoria y la implementación de mecanismos más sofisticados para prevenir situaciones de inanición en el algoritmo de prioridades.

7. Conclusiones

El desarrollo de este proyecto permitió aplicar de manera práctica los conceptos teóricos estudiados en el curso de Sistemas Operativos. Las pruebas realizadas demostraron que el simulador implementa correctamente los mecanismos fundamentales de gestión de procesos, memoria y planificación.

Las métricas recolectadas evidencian que el algoritmo Round Robin ofrece un mejor balance entre tiempo de respuesta y tiempo de espera en comparación con los otros algoritmos evaluados. Sin embargo, cada algoritmo presenta ventajas específicas según el tipo de carga de trabajo del sistema.

El proceso de pruebas también reveló la importancia de realizar validaciones exhaustivas para garantizar la robustez y confiabilidad de los sistemas operativos, incluso cuando se trata de simuladores con fines académicos.

8. Recomendaciones

Para trabajos futuros, se sugiere considerar las siguientes mejoras al simulador desarrollado:

- Implementar algoritmos adicionales de gestión de memoria como Worst Fit y Best Fit para realizar comparaciones más completas.
- Agregar soporte para la simulación de múltiples procesadores y evaluar algoritmos de planificación distribuida.
- Incluir mecanismos de sincronización entre procesos como semáforos y monitores.
- Desarrollar una interfaz gráfica más avanzada que permita visualizar el diagrama de Gantt en tiempo real.
- Incorporar la capacidad de cargar escenarios de prueba desde archivos de configuración externos.