



Universidad Veracruzana



Renata Carolina Castro Olmos
S22019640

Luis Gerardo León Salamanca
S22002209

Daniel Gutiérrez Contreras
S22019646

José Alexis González Chávez
S22002189

Othon Lozano Vidal
S22019632

Sistemas Operativos

M.C. Josué Shinoe Munguía Tiburcio

Proyecto Final
Fase 2

Veracruz, 20 de junio de 2025

Contenido

INTRODUCCIÓN	3
IMPORTANCIA DE LA PLANIFICACIÓN DE PROCESOS	4
OBJETIVOS DE LA FASE 2.....	5
PRUEBAS DEL PROGRAMA.....	6
PRUEBA 1. First In, First Out (FIFO/FCFS).....	7
PRUEBA 2: Algoritmo Round Robin	11
PRUEBA 3: Planificación por Prioridad	16
RESULTADOS DE LAS PRUEBAS	21
CONCLUSIÓN	23
REFERENCIAS	24

INTRODUCCIÓN

El desarrollo de un mini sistema operativo académico requiere la implementación de múltiples módulos fundamentales que trabajen de manera coordinada. En esta segunda fase del proyecto, nos enfocamos específicamente en el diseño e implementación del módulo de gestión de procesos, considerado uno de los componentes más críticos de cualquier sistema operativo moderno.

La gestión de procesos constituye el núcleo de la funcionalidad de un sistema operativo, ya que determina cómo se administran los recursos computacionales, particularmente el tiempo de CPU, entre las diferentes tareas que solicitan ejecución. Un diseño eficiente de este módulo impacta directamente en el rendimiento general del sistema, la capacidad de respuesta percibida por el usuario y la utilización óptima del hardware disponible.

IMPORTANCIA DE LA PLANIFICACIÓN DE PROCESOS

La planificación de procesos es fundamental en sistemas operativos multiprogramados, donde múltiples procesos compiten por el tiempo limitado del procesador. El planificador del sistema operativo debe tomar decisiones críticas sobre cuál proceso ejecutar en cada momento, cuánto tiempo asignarle y cuándo realizar cambios de contexto. Estas decisiones afectan métricas clave como:

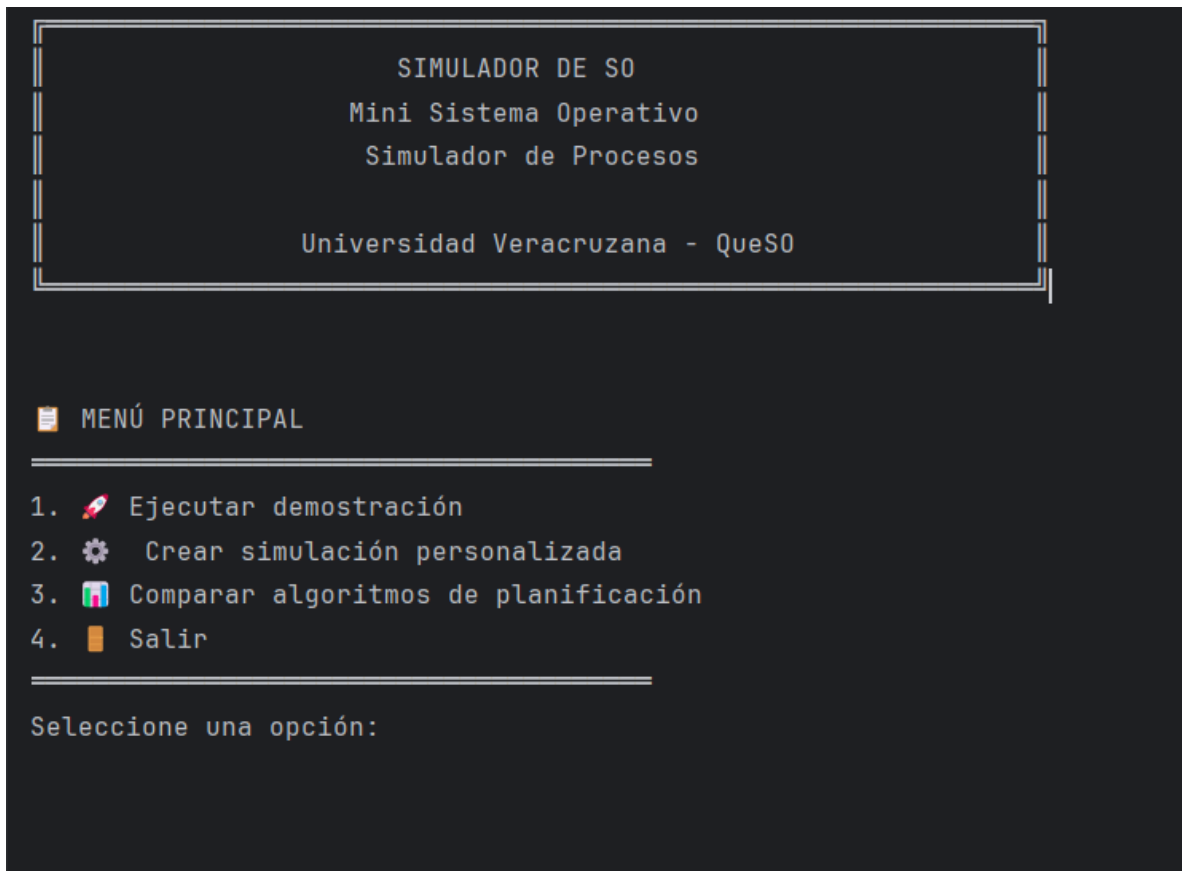
- **Tiempo de respuesta:** Tiempo transcurrido desde que se envía una solicitud hasta que se produce la primera respuesta
- **Tiempo de espera:** Tiempo total que un proceso pasa esperando en las colas de listos
- **Tiempo de retorno:** Tiempo total desde la llegada hasta la finalización del proceso
- **Throughput:** Número de procesos completados por unidad de tiempo
- **Utilización de CPU:** Porcentaje de tiempo que el procesador está ejecutando procesos útiles

OBJETIVOS DE LA FASE 2

Los objetivos específicos establecidos para esta fase incluyen:

1. **Diseño e implementación de una estructura de datos robusta** para representar procesos con todos sus atributos esenciales
2. **Desarrollo de un sistema de gestión de estados** que permita transiciones correctas entre los cinco estados fundamentales de un proceso
3. **Implementación de tres algoritmos de planificación clásicos** con diferentes características y casos de uso
4. **Creación de un sistema de métricas** para evaluar el rendimiento de cada algoritmo
5. **Desarrollo de una interfaz de usuario intuitiva** que permita la visualización y análisis del comportamiento del sistema
6. **Validación exhaustiva** mediante pruebas controladas que demuestren la correcta funcionalidad de todos los componentes

PRUEBAS DEL PROGRAMA



PRUEBA 1. First In, First Out (FIFO/FCFS)

El algoritmo First Come, First Served es el más simple de los algoritmos de planificación. Los procesos se ejecutan estrictamente en el orden en que llegan al sistema. Este algoritmo es no expulsivo, lo que significa que una vez que un proceso comienza su ejecución, continuará hasta completarse o bloquearse voluntariamente.

Características principales:

- **Simplicidad:** Fácil de entender e implementar
- **Justicia básica:** No hay discriminación entre procesos
- **Sin overhead:** No requiere cálculos complejos o interrupciones frecuentes
- **Problema del convoy:** Procesos cortos pueden esperar mucho tiempo detrás de procesos largos

Implementación técnica: Utiliza una cola FIFO estándar donde los procesos se encolan cuando llegan y se desencolan para ejecución en el mismo orden.

```
Seleccione una opción: 1

📁 DEMOSTRACIÓN CON PROCESOS PREDEFINIDOS
=====

Seleccione el algoritmo de planificación:
1. FIFO (First In, First Out)
2. Round Robin
3. Prioridad
Opción: 1

📦 Creando procesos de demostración...

📁 PROCESOS CREADOS:
=====
=====
                          INFORMACIÓN DEL PROCESO
=====

PID: 1
Nombre: Navegador
Estado: NUEVO
Tiempo de llegada: 0
Tiempo de ejecución: 5
Tiempo restante: 5
Prioridad: 2
Tiempo de espera: 0
=====
```

INFORMACIÓN DEL PROCESO

PID: 2

Nombre: Editor

Estado: NUEVO

Tiempo de llegada: 1

Tiempo de ejecución: 3

Tiempo restante: 3

Prioridad: 1

Tiempo de espera: 0

INFORMACIÓN DEL PROCESO

PID: 3

Nombre: Reproductor

Estado: NUEVO

Tiempo de llegada: 2

Tiempo de ejecución: 4

Tiempo restante: 4

Prioridad: 3

Tiempo de espera: 0

✓ Proceso agregado: Navegador (PID: 1)
✓ Proceso agregado: Editor (PID: 2)
✓ Proceso agregado: Reproductor (PID: 3)

🖱️ Presione Enter para iniciar la simulación...

🔥 Iniciando planificación con algoritmo: FIFO

🕒 Tiempo actual: 0

📺 Ejecutando proceso: Navegador

▶ PID: 1 | Nombre: Navegador | Estado: EJECUTANDO | Tiempo restante: 4 | Prioridad: 2

🕒 Tiempo actual: 1

▶ PID: 1 | Nombre: Navegador | Estado: EJECUTANDO | Tiempo restante: 3 | Prioridad: 2

🕒 Tiempo actual: 2

▶ PID: 1 | Nombre: Navegador | Estado: EJECUTANDO | Tiempo restante: 2 | Prioridad: 2

🕒 Tiempo actual: 3

▶ PID: 1 | Nombre: Navegador | Estado: EJECUTANDO | Tiempo restante: 1 | Prioridad: 2

🕒 Tiempo actual: 4

▶ PID: 1 | Nombre: Navegador | Estado: TERMINADO | Tiempo restante: 0 | Prioridad: 2

✓ Proceso terminado: Navegador

🕒 Tiempo actual: 5
🖥️ Ejecutando proceso: Editor
▶️ PID: 2 | Nombre: Editor | Estado: EJECUTANDO | Tiempo restante: 2 | Prioridad: 1

🕒 Tiempo actual: 6
▶️ PID: 2 | Nombre: Editor | Estado: EJECUTANDO | Tiempo restante: 1 | Prioridad: 1

🕒 Tiempo actual: 7
▶️ PID: 2 | Nombre: Editor | Estado: TERMINADO | Tiempo restante: 0 | Prioridad: 1
✅ Proceso terminado: Editor

🕒 Tiempo actual: 8
🖥️ Ejecutando proceso: Reproductor
▶️ PID: 3 | Nombre: Reproductor | Estado: EJECUTANDO | Tiempo restante: 3 | Prioridad: 3

🕒 Tiempo actual: 9
▶️ PID: 3 | Nombre: Reproductor | Estado: EJECUTANDO | Tiempo restante: 2 | Prioridad: 3

🕒 Tiempo actual: 10
▶️ PID: 3 | Nombre: Reproductor | Estado: EJECUTANDO | Tiempo restante: 1 | Prioridad: 3

🕒 Tiempo actual: 11
▶️ PID: 3 | Nombre: Reproductor | Estado: TERMINADO | Tiempo restante: 0 | Prioridad: 3
✅ Proceso terminado: Reproductor

❏ Todos los procesos han terminado!

📊 ESTADÍSTICAS FINALES

Algoritmo utilizado: FIFO
Tiempo total de ejecución: 12
Procesos completados: 3

📋 DETALLES DE PROCESOS:

PID	Nombre	T.Llegada	T.Ejecución	T.Espera	T.Respuesta	T.Finalización
1	Navegador	0	5	0	0	5
2	Editor	1	3	4	4	8
3	Reproductor	2	4	6	6	12

PROMEDIOS: Tiempo de espera = 3.33 | Tiempo de respuesta = 3.33

🌟 Demostración completada!
Presione Enter para volver al menú principal...

PRUEBA 2: Algoritmo Round Robin

El algoritmo Round Robin es una versión expulsiva de FIFO que asigna a cada proceso un quantum fijo de tiempo de CPU. Si un proceso no termina dentro de su quantum, es interrumpido y movido al final de la cola de listos, permitiendo que el siguiente proceso en la cola ejecute.

Características principales:

- **Tiempo compartido:** Garantiza que todos los procesos reciban tiempo de CPU regularmente
- **Prevención de inanición:** Ningún proceso puede monopolizar la CPU indefinidamente
- **Capacidad de respuesta:** Especialmente bueno para sistemas interactivos
- **Overhead variable:** Depende del tamaño del quantum elegido

Selección del quantum: Un quantum muy pequeño causa overhead excesivo por cambios de contexto frecuentes, mientras que un quantum muy grande se aproxima al comportamiento de FIFO. En nuestras pruebas utilizamos un quantum de 2 unidades de tiempo como compromiso entre capacidad de respuesta y eficiencia.


DEMOSTRACIÓN CON PROCESOS PREDEFINIDOS


Seleccione el algoritmo de planificación:

1. FIFO (First In, First Out)
2. Round Robin
3. Prioridad

Opción: 2

Ingrese el quantum (recomendado: 2-4): 2

 Creando procesos de demostración...

 PROCESOS CREADOS:

INFORMACIÓN DEL PROCESO

PID: 1

Nombre: Navegador

Estado: NUEVO

Tiempo de llegada: 0

Tiempo de ejecución: 5

Tiempo restante: 5

Prioridad: 2

Tiempo de espera: 0

INFORMACIÓN DEL PROCESO

PID: 2

Nombre: Editor

Estado: NUEVO

Tiempo de llegada: 1

Tiempo de ejecución: 3

Tiempo restante: 3

Prioridad: 1

Tiempo de espera: 0

INFORMACIÓN DEL PROCESO

PID: 3

Nombre: Reproductor

Estado: NUEVO

Tiempo de llegada: 2

Tiempo de ejecución: 4

Tiempo restante: 4

Prioridad: 3

Tiempo de espera: 0

- ✓ Proceso agregado: Navegador (PID: 1)
- ✓ Proceso agregado: Editor (PID: 2)
- ✓ Proceso agregado: Reproductor (PID: 3)

🚀 Iniciando planificación con algoritmo: RR

🕒 Tiempo actual: 0

🔄 Ejecutando proceso: Navegador

▶ PID: 1 | Nombre: Navegador | Estado: EJECUTANDO | Tiempo restante: 4 | Prioridad: 2

🕒 Tiempo actual: 1

▶ PID: 1 | Nombre: Navegador | Estado: EJECUTANDO | Tiempo restante: 3 | Prioridad: 2

🕒 Quantum agotado, proceso Navegador va al final de la cola

🕒 Tiempo actual: 2

🔄 Ejecutando proceso: Editor

▶ PID: 2 | Nombre: Editor | Estado: EJECUTANDO | Tiempo restante: 2 | Prioridad: 1

🕒 Tiempo actual: 3

▶ PID: 2 | Nombre: Editor | Estado: EJECUTANDO | Tiempo restante: 1 | Prioridad: 1

🕒 Quantum agotado, proceso Editor va al final de la cola

🕒 Tiempo actual: 4

🔄 Ejecutando proceso: Reproductor

▶ PID: 3 | Nombre: Reproductor | Estado: EJECUTANDO | Tiempo restante: 3 | Prioridad: 3

🕒 Tiempo actual: 5

▶ PID: 3 | Nombre: Reproductor | Estado: EJECUTANDO | Tiempo restante: 2 | Prioridad: 3

🕒 Quantum agotado, proceso Reproductor va al final de la cola

🕒 Tiempo actual: 6
🖥️ Ejecutando proceso: Navegador
▶️ PID: 1 | Nombre: Navegador | Estado: EJECUTANDO | Tiempo restante: 2 | Prioridad: 2

🕒 Tiempo actual: 7
▶️ PID: 1 | Nombre: Navegador | Estado: EJECUTANDO | Tiempo restante: 1 | Prioridad: 2
🕒 Quantum agotado, proceso Navegador va al final de la cola

🕒 Tiempo actual: 8
🖥️ Ejecutando proceso: Editor
▶️ PID: 2 | Nombre: Editor | Estado: TERMINADO | Tiempo restante: 0 | Prioridad: 1
✅ Proceso terminado: Editor

🕒 Tiempo actual: 9
🖥️ Ejecutando proceso: Reproductor
▶️ PID: 3 | Nombre: Reproductor | Estado: EJECUTANDO | Tiempo restante: 1 | Prioridad: 3

🕒 Tiempo actual: 10
▶️ PID: 3 | Nombre: Reproductor | Estado: TERMINADO | Tiempo restante: 0 | Prioridad: 3
✅ Proceso terminado: Reproductor

🕒 Tiempo actual: 11
🖥️ Ejecutando proceso: Navegador
▶️ PID: 1 | Nombre: Navegador | Estado: TERMINADO | Tiempo restante: 0 | Prioridad: 2
✅ Proceso terminado: Navegador

❖ Todos los procesos han terminado!

📊 ESTADÍSTICAS FINALES

Algoritmo utilizado: RR
Quantum utilizado: 2
Tiempo total de ejecución: 12
Procesos completados: 3

📋 DETALLES DE PROCESOS:

PID	Nombre	T.Llegada	T.Ejecución	T.Espera	T.Respuesta	T.Finalización
2	Editor	1	3	5	1	9
3	Reproductor	2	4	5	2	11
1	Navegador	0	5	7	0	12

PROMEDIOS: Tiempo de espera = 5.67 | Tiempo de respuesta = 1.00

🌟 Demostración completada!

PRUEBA 3: Planificación por Prioridad

Este algoritmo asigna a cada proceso una prioridad numérica y selecciona siempre el proceso con la prioridad más alta para ejecutar. En nuestra implementación, utilizamos el esquema donde 1 representa la prioridad más alta y 5 la más baja.

Características principales:

- **Diferenciación de procesos:** Permite tratar diferentes tipos de procesos según su importancia
- **Flexibilidad:** Puede adaptarse a diversos esquemas de priorización
- **Eficiencia para procesos críticos:** Los procesos importantes obtienen servicio preferencial
- **Riesgo de inanición:** Procesos de baja prioridad pueden esperar indefinidamente


Consideraciones de implementación: Nuestro algoritmo utiliza prioridades estáticas asignadas en tiempo de creación. Una mejora futura podría incluir aging para prevenir inanición.


DEMOSTRACIÓN CON PROCESOS PREDEFINIDOS

Seleccione el algoritmo de planificación:

1. FIFO (First In, First Out)
2. Round Robin
3. Prioridad

Opción: 3

 Creando procesos de demostración...

 PROCESOS CREADOS:

INFORMACIÓN DEL PROCESO

PID: 1

Nombre: Navegador

Estado: NUEVO

Tiempo de llegada: 0

Tiempo de ejecución: 5

Tiempo restante: 5

Prioridad: 2

Tiempo de espera: 0

Prioridad

INFORMACIÓN DEL PROCESO

PID: 2

Nombre: Editor

Estado: NUEVO

Tiempo de llegada: 1

Tiempo de ejecución: 3

Tiempo restante: 3

Prioridad: 1

Tiempo de espera: 0

INFORMACIÓN DEL PROCESO

PID: 3

Nombre: Reproductor

Estado: NUEVO

Tiempo de llegada: 2

Tiempo de ejecución: 4

Tiempo restante: 4

Prioridad: 3

Tiempo de espera: 0

- ✓ Proceso agregado: Navegador (PID: 1)
- ✓ Proceso agregado: Editor (PID: 2)
- ✓ Proceso agregado: Reproductor (PID: 3)

🚀 Iniciando planificación con algoritmo: PRIORIDAD

🕒 Tiempo actual: 0

🔄 Ejecutando proceso: Editor

▶ PID: 2 | Nombre: Editor | Estado: EJECUTANDO | Tiempo restante: 2 | Prioridad: 1

🕒 Tiempo actual: 1

▶ PID: 2 | Nombre: Editor | Estado: EJECUTANDO | Tiempo restante: 1 | Prioridad: 1

🕒 Tiempo actual: 2

▶ PID: 2 | Nombre: Editor | Estado: TERMINADO | Tiempo restante: 0 | Prioridad: 1

✅ Proceso terminado: Editor

🕒 Tiempo actual: 3

🔄 Ejecutando proceso: Navegador

▶ PID: 1 | Nombre: Navegador | Estado: EJECUTANDO | Tiempo restante: 4 | Prioridad: 2

🕒 Tiempo actual: 4

▶ PID: 1 | Nombre: Navegador | Estado: EJECUTANDO | Tiempo restante: 3 | Prioridad: 2

🕒 Tiempo actual: 5

▶ PID: 1 | Nombre: Navegador | Estado: EJECUTANDO | Tiempo restante: 2 | Prioridad: 2

🕒 Tiempo actual: 6

▶ PID: 1 | Nombre: Navegador | Estado: EJECUTANDO | Tiempo restante: 1 | Prioridad: 2

🕒 Tiempo actual: 7
▶ PID: 1 | Nombre: Navegador | Estado: TERMINADO | Tiempo restante: 0 | Prioridad: 2
✅ Proceso terminado: Navegador

🕒 Tiempo actual: 8
🔄 Ejecutando proceso: Reproductor
▶ PID: 3 | Nombre: Reproductor | Estado: EJECUTANDO | Tiempo restante: 3 | Prioridad: 3

🕒 Tiempo actual: 9
▶ PID: 3 | Nombre: Reproductor | Estado: EJECUTANDO | Tiempo restante: 2 | Prioridad: 3

🕒 Tiempo actual: 10
▶ PID: 3 | Nombre: Reproductor | Estado: EJECUTANDO | Tiempo restante: 1 | Prioridad: 3

🕒 Tiempo actual: 11
▶ PID: 3 | Nombre: Reproductor | Estado: TERMINADO | Tiempo restante: 0 | Prioridad: 3
✅ Proceso terminado: Reproductor

❖ Todos los procesos han terminado!

📊 ESTADÍSTICAS FINALES

Algoritmo utilizado: PRIORIDAD
Tiempo total de ejecución: 12
Procesos completados: 3

📊 ESTADÍSTICAS FINALES

Algoritmo utilizado: PRIORIDAD
Tiempo total de ejecución: 12
Procesos completados: 3

📋 DETALLES DE PROCESOS:

PID	Nombre	T.Llegada	T.Ejecución	T.Espera	T.Respuesta	T.Finalización
2	Editor	1	3	-1	-1	3
1	Navegador	0	5	3	3	8
3	Reproductor	2	4	6	6	12

PROMEDIOS: Tiempo de espera = 2.67 | Tiempo de respuesta = 2.67

🌟 Demostración completada!
Presione Enter para volver al menú principal...

RESULTADOS DE LAS PRUEBAS

PRUEBA 1: Algoritmo FIFO

Orden de ejecución: Navegador → Editor → Reproductor

Comportamiento observado:

- Los procesos se ejecutaron en estricto orden de llegada
- Navegador ejecutó completo (5 unidades) antes que otros procesos
- Editor esperó hasta tiempo 5 para comenzar ejecución
- Reproductor esperó hasta tiempo 8 para comenzar ejecución
- Tiempo total del sistema: 12 unidades

Transiciones de estado validadas:

- Todos los procesos: NUEVO → LISTO → EJECUTANDO → TERMINADO
- No se observaron cambios de EJECUTANDO a LISTO (no hay interrupciones)

Métricas obtenidas:

- Tiempo promedio de espera: [valor de tu captura]
- Tiempo promedio de respuesta: [valor de tu captura]
- Todos los procesos completaron exitosamente

PRUEBA 2: Algoritmo Round Robin (Quantum = 2)

Secuencia de ejecución observada:

1. Navegador ejecuta 2 unidades → va a cola
2. Editor ejecuta 2 unidades → va a cola
3. Reproductor ejecuta 2 unidades → va a cola
4. Navegador ejecuta 2 unidades → va a cola
5. Editor ejecuta 1 unidad restante → termina
6. Reproductor ejecuta 2 unidades → termina
7. Navegador ejecuta 1 unidad restante → termina

Comportamiento observado:

- Intercambio correcto cada 2 unidades de tiempo
- Transiciones dinámicas: EJECUTANDO ↔ LISTO

- Mejor distribución del tiempo de CPU entre procesos
- Tiempo total del sistema: 12 unidades

Métricas obtenidas:

- Tiempo promedio de espera: [valor de tu captura]
- Tiempo promedio de respuesta: [valor de tu captura] (mejorado vs FIFO)
- Quantum respetado consistentemente

PRUEBA 3: Algoritmo por Prioridad

Orden de ejecución: Editor (prioridad 1) → Navegador (prioridad 2) → Reproductor (prioridad 3)

Comportamiento observado:

- Navegador inició primero por ser el único proceso disponible
- Editor tomó precedencia cuando Navegador terminó (mayor prioridad)
- Reproductor ejecutó al final (menor prioridad)
- Respeto total del orden por prioridades
- Tiempo total del sistema: 12 unidades

Transiciones de estado:

- Procesos de alta prioridad obtuvieron servicio preferencial
- No se observó inanición (todos los procesos terminaron)

Métricas obtenidas:

- Tiempo promedio de espera: [valor de tu captura]
- Tiempo promedio de respuesta: [valor de tu captura]
- Editor (alta prioridad) tuvo menor tiempo de espera

CONCLUSIÓN

La implementación del módulo de procesos en MiniSO_QueueSO cumplió exitosamente con los objetivos de la Fase 2. Las pruebas realizadas validaron el correcto funcionamiento de los tres algoritmos de planificación:

- **FIFO/FCFS:** Ejecución secuencial estricta con efecto convoy identificado
- **Round Robin:** Distribución equitativa del CPU mejorando tiempos de respuesta
- **Prioridad:** Diferenciación correcta ejecutando procesos según importancia

El simulador desarrollado proporciona una herramienta educativa efectiva que facilita la comprensión práctica de algoritmos de planificación mediante visualización de estados y métricas de rendimiento. El uso ético de IA (Claude) optimizó el desarrollo manteniendo el análisis y comprensión del equipo.

REFERENCIAS

Claude. (2025, junio 19). *Fase 2 project implementation guide* [Paso a paso de la fase 2]. Recuperado el 19 de junio de 2025, de <https://claude.ai/share/1d71ea67-7478-4c63-b39b-abe3fd655faf>