

The background is a light beige or off-white color with a subtle texture. It features several large, organic, wavy shapes in muted colors: a light green shape in the top left, a light orange shape in the top right, a teal shape in the bottom left, and another light green shape in the bottom right. Scattered throughout these shapes and the background are numerous small, yellowish-orange circular splatters and dots, giving it a hand-drawn or watercolor-like appearance. Thin black lines also trace some of the boundaries of the colored shapes.

# **Entrega Final**

## **SISTEMAS OPERATIVOS**

Alexandra Hurtado  
Camilo Salazar  
Mariana Valderrama

# INFORME TECNICO

Link de github: <https://github.com/Marianavc02/sistemasOperativos>

## 1. Arquitectura General del Sistema

El sistema desarrollado simula un núcleo de sistema operativo simplificado, implementado en C++ y basado en ejecución concurrente con `std::thread`. El sistema está diseñado de forma modular, agrupando las funcionalidades principales en bloques lógicos accesibles a través de una interfaz por consola (CLI). Los componentes clave del sistema son:

Gestor de procesos: permite crear, suspender, reanudar, terminar y listar procesos simulados.

Planificador de procesos: permite aplicar dos algoritmos de planificación: Round Robin y Shortest Job First (SJF).

Gestor de memoria: simula paginación con algoritmos FIFO y LRU.

Mecanismos de sincronización: resuelve problemas clásicos como productor-consumidor y la cena de los filósofos mediante `std::mutex` y `std::condition_variable`.

Simulación de E/S: maneja recursos como impresoras compartidas con acceso sincronizado.

Planificación de disco: implementa algoritmos FCFS y SSTF.

Cada módulo está diseñado para trabajar de forma autónoma pero puede integrarse en una simulación completa, reflejando el comportamiento de un sistema operativo multitarea.

## 2. Algoritmos Implementados y Justificación

### 2.1 Planificación de Procesos

#### ● Round Robin

Descripción: Asigna un "quantum" de tiempo fijo a cada proceso. Si el proceso no termina en ese tiempo, se reenvía al final de la cola.

Justificación: Ideal para sistemas interactivos con tiempos de respuesta equitativos. Evita inanición y permite compartir CPU entre múltiples tareas.

#### ● Shortest Job First (SJF)

Descripción: Selecciona el proceso con menor tiempo de ejecución estimado.

Justificación: Minimiza el tiempo promedio de espera, mejorando el rendimiento del sistema cuando los tiempos de CPU son conocidos.

### 2.2 Gestión de Memoria Virtual

#### ● FIFO (First-In First-Out)

Descripción: Las páginas se almacenan en orden de llegada; la primera que entra es la primera en ser reemplazada.

Justificación: Fácil de implementar, pero propenso a la anomalía de Belady. Se incluye por su valor pedagógico.

#### ● LRU (Least Recently Used)

Descripción: Reemplaza la página que no ha sido utilizada por el mayor tiempo.

Justificación: Mejora el rendimiento al adaptarse al uso reciente de páginas. Refleja un comportamiento más realista en entornos modernos.

### **2.3 Sincronización de Procesos**

#### ● Productor-Consumidor

Descripción: Un hilo productor inserta elementos en un buffer compartido, mientras que el consumidor los retira.

Justificación: Modelo clásico para ilustrar sincronización en acceso a recursos compartidos. Se implementa con `std::mutex` y `std::condition_variable` para evitar condiciones de carrera.

#### ● Cena de los Filósofos

Descripción: Cada filósofo alterna entre pensar y comer. Para comer, debe adquirir dos tenedores (recursos compartidos).

Justificación: Demuestra la gestión de interbloqueo y exclusión mutua. Se aplica una estrategia de orden de adquisición para evitar deadlocks.

### **2.4 Entrada/Salida y Manejo de Recursos**

#### ● Impresora Compartida

Descripción: Varios empleados (hilos) preparan documentos y esperan su turno para imprimir.

Justificación: Refleja la exclusión mutua en un entorno compartido. Se emplea una cola de espera y mutex para permitir que solo un empleado imprima a la vez. El sistema registra quién está esperando y quién imprime.

### **2.5 Planificación de Disco**

#### ● FCFS (First-Come, First-Served)

Descripción: Las solicitudes de acceso a disco se atienden en orden de llegada.

Justificación: Justo y simple, aunque ineficiente si hay saltos grandes entre pistas.

#### ● SSTF (Shortest Seek Time First)

Descripción: Atiende la solicitud más cercana a la posición actual del cabezal.

Justificación: Reduce el tiempo de búsqueda y mejora la eficiencia respecto a FCFS.

## 2.6 Capturas o gráficas del comportamiento de los subsistemas

```
===== SIMULADOR DE KERNEL =====  
1. Gestión de Procesos  
2. Ejecutar  
3. Memoria  
4. Simular E/S (Impresora)  
5. Planificación Disco (FCFS/SSTF)  
6. Simular Productor-Consumidor  
7. Simular Cena de Filósofos  
8. Salir  
Seleccione opción: 1
```

```
=== Gestión de Procesos ===  
1. Crear Proceso  
2. Suspender Proceso  
3. Reanudar Proceso  
4. Terminar Proceso  
5. Ver Procesos  
6. Regresar  
Seleccione opción: 1  
ID del proceso: 1  
Tiempo de CPU: 2  
Proceso 1 creado en estado NUEVO.
```

=== Gestión de Procesos ===

1. Crear Proceso
2. Suspender Proceso
3. Reanudar Proceso
4. Terminar Proceso
5. Ver Procesos
6. Regresar

Seleccione opción: 1

ID del proceso: 1

Tiempo de CPU: 2

Proceso 1 creado en estado NUEVO.

=== Gestión de Procesos ===

1. Crear Proceso
2. Suspender Proceso
3. Reanudar Proceso
4. Terminar Proceso
5. Ver Procesos
6. Regresar

Seleccione opción: 1

ID del proceso: 2

Tiempo de CPU: 3

Proceso 2 creado en estado NUEVO.

T

```
=== Gestión de Procesos ===
1. Crear Proceso
2. Suspender Proceso
3. Reanudar Proceso
4. Terminar Proceso
5. Ver Procesos
6. Regresar
Seleccione opción: 1
ID del proceso: 3
Tiempo de CPU: 3
Proceso 3 creado en estado NUEVO.
```

```
=== Gestión de Procesos ===
1. Crear Proceso
2. Suspender Proceso
3. Reanudar Proceso
4. Terminar Proceso
5. Ver Procesos
6. Regresar
Seleccione opción: 5
```

ID	Estado	Tiempo restante
1	NUEVO	2
2	NUEVO	3
3	NUEVO	3

```
===== SIMULADOR DE KERNEL =====
1. Gestión de Procesos
2. Ejecutar
3. Memoria
4. Simular E/S (Impresora)
5. Planificación Disco (FCFS/SSTF)
6. Simular Productor-Consumidor
7. Simular Cena de Filósofos
8. Salir
Seleccione opción: 2
```



=== Ejecutar Planificación ===

1. Planificación Básica
2. Planificación Round Robin
3. Planificación SJF
4. Regresar

Seleccione opción: 2

Ejecutando Round Robin con quantum = 2

Ejecutando 1 por 2 unidades de tiempo.

Ejecutando 2 por 2 unidades de tiempo.

Ejecutando 3 por 2 unidades de tiempo.

Ejecutando 2 por 1 unidades de tiempo.

Ejecutando 3 por 1 unidades de tiempo.

Proceso	Llegada	Ejecución	Finalización	Retorno	Espera
1	0	2	2	2	0
2	0	3	7	7	4
3	0	3	8	8	5

Tiempo promedio de espera: 3 unidades

Tiempo promedio de retorno: 5.66667 unidades

=== Ejecutar Planificación ===

1. Planificación Básica
2. Planificación Round Robin
3. Planificación SJF
4. Regresar

Seleccione opción: 3

Ejecutando SJF (Shortest Job First)

Ejecutando 1 por 2 unidades.

Ejecutando 2 por 2 unidades.

Ejecutando 3 por 3 unidades.

Proceso	Llegada	Ejecución	Finalización	Retorno	Espera
1	0	2	2	2	0
2	0	2	4	4	2
3	0	3	7	7	4

Tiempo promedio de espera: 2 unidades

Tiempo promedio de retorno: 4.33333 unidades

===== SIMULADOR DE KERNEL =====

1. Gestión de Procesos
2. Ejecutar
3. Memoria
4. Simular E/S (Impresora)
5. Planificación Disco (FCFS/SSTF)
6. Simular Productor-Consumidor
7. Simular Cena de Filósofos
8. Salir

Seleccione opción: 3

=== Memoria ===

1. Memoria FIFO
2. Memoria LRU
3. Regresar

Seleccione opción: 1



Página 7 cargada en un marco vacío.  
Memoria: 7 | Fallos: 1

Estado actual de memoria: 7  
Página 0 cargada en un marco vacío.  
Memoria: 7 0 | Fallos: 2

Estado actual de memoria: 7 0  
Página 1 cargada en un marco vacío.  
Memoria: 7 0 1 | Fallos: 3

Estado actual de memoria: 7 0 1  
Página 2 cargada en un marco vacío.  
Memoria: 7 0 1 2 | Fallos: 4

Estado actual de memoria: 7 0 1 2  
Página 0 ya en memoria.  
Memoria: 7 0 1 2 | Fallos: 4

Estado actual de memoria: 7 0 1 2  
Página 7 reemplazada por 3.  
Memoria: 0 1 2 3 | Fallos: 5

Estado actual de memoria: 0 1 2 3  
Página 0 ya en memoria.  
Memoria: 0 1 2 3 | Fallos: 5

Estado actual de memoria: 0 1 2 3  
Página 0 reemplazada por 4.  
Memoria: 1 2 3 4 | Fallos: 6

Estado actual de memoria: 1 2 3 4  
Página 2 ya en memoria.

```
Estado actual de memoria: 1 2 3 4
P gina 2 ya en memoria.
Memoria: 1 2 3 4 | Fallos: 6
```

```
Estado actual de memoria: 1 2 3 4
P gina 3 ya en memoria.
Memoria: 1 2 3 4 | Fallos: 6
```

```
Estado actual de memoria: 1 2 3 4
P gina 1 reemplazada por  .
Memoria: 2 3 4   | Fallos: 7
```

```
Estado actual de memoria: 2 3 4  
P gina 3 ya en memoria.
Memoria: 2 3 4   | Fallos: 7
```

```
Estado actual de memoria: 2 3 4  
P gina 2 ya en memoria.
Memoria: 2 3 4   | Fallos: 7
```

```
Total de fallos de p gina: 7
```

```
=== Memoria ===
```

```
1. Memoria FIFO
```

```
2. Memoria LRU
```

```
3. Regresar
```

```
Seleccione opci n: 2
```

```
Procesando p|ígina: 7
Estado previo del cach|®:
P|ígina 7 cargada en memoria.
Estado actual del cach|®: 7 | Fallos: 1

Procesando p|ígina: 0
Estado previo del cach|®: 7
P|ígina 0 cargada en memoria.
Estado actual del cach|®: 0 7 | Fallos: 2

Procesando p|ígina: 1
Estado previo del cach|®: 0 7
P|ígina 1 cargada en memoria.
Estado actual del cach|®: 1 0 7 | Fallos: 3

Procesando p|ígina: 2
Estado previo del cach|®: 1 0 7
P|ígina 2 cargada en memoria.
Estado actual del cach|®: 2 1 0 7 | Fallos: 4

Procesando p|ígina: 0
Estado previo del cach|®: 2 1 0 7
P|ígina 0 ya estaba en memoria (actualizada a m|ís reciente).
Estado actual del cach|®: 0 2 1 7 | Fallos: 4
```

```
Procesando página: 3
Estado previo del caché: 0 2 1 7
Página 7 reemplazada por 3.
Página 3 cargada en memoria.
Estado actual del caché: 3 0 2 1 | Fallos: 5

Procesando página: 0
Estado previo del caché: 3 0 2 1
Página 0 ya estaba en memoria (actualizada a más reciente).
Estado actual del caché: 0 3 2 1 | Fallos: 5

Procesando página: 4
Estado previo del caché: 0 3 2 1
Página 1 reemplazada por 4.
Página 4 cargada en memoria.
Estado actual del caché: 4 0 3 2 | Fallos: 6

Procesando página: 2
Estado previo del caché: 4 0 3 2
Página 2 ya estaba en memoria (actualizada a más reciente).
Estado actual del caché: 2 4 0 3 | Fallos: 6

Procesando página: 3
Estado previo del caché: 2 4 0 3
Página 3 ya estaba en memoria (actualizada a más reciente).
Estado actual del caché: 3 2 4 0 | Fallos: 6

Procesando página: 0
Estado previo del caché: 3 2 4 0
Página 0 ya estaba en memoria (actualizada a más reciente).
Estado actual del caché: 0 3 2 4 | Fallos: 6
```



```
Procesando p gina: 3
Estado previo del cach : 0 3 2 4
P gina 3 ya estaba en memoria (actualizada a m s reciente).
Estado actual del cach : 3 0 2 4 | Fallos: 6
```

```
Procesando p gina: 2
Estado previo del cach : 3 0 2 4
P gina 2 ya estaba en memoria (actualizada a m s reciente).
Estado actual del cach : 2 3 0 4 | Fallos: 6
```

```
Total de fallos de p gina: 6
```

```
===== SIMULADOR DE KERNEL =====
1. Gest n de Procesos
2. Ejecutar
3. Memoria
4. Simular E/S (Impresora)
5. Planificaci n Disco (FCFS/SSTF)
6. Simular Productor-Consumidor
7. Simular Cena de Fil sofos
8. Salir
Seleccione opci n: 4
```

```
=== SISTEMA DE IMPRESION COMPARTIDA ===
[IMPRESION] Empleado 3 est|í esperando para imprimir.
  >> Empleados esperando: 3
[IMPRESION] Empleado 3 est|í imprimiendo...
  >> Empleados esperando:
[IMPRESION] Empleado 1 est|í esperando para imprimir.
  >> Empleados esperando: 1
[IMPRESION] Empleado 4 est|í esperando para imprimir.
  >> Empleados esperando: 1 4
[IMPRESION] Empleado 5 est|í esperando para imprimir.
  >> Empleados esperando: 1 4 5
[IMPRESION] Empleado 2 est|í esperando para imprimir.
  >> Empleados esperando: 1 4 5 2
[IMPRESION] Empleado 3 ha terminado de imprimir.
  >> Empleados esperando: 1 4 5 2
[IMPRESION] Empleado 1 est|í imprimiendo...
  >> Empleados esperando: 4 5 2
[IMPRESION] Empleado 1 ha terminado de imprimir.
  >> Empleados esperando: 4 5 2
[IMPRESION] Empleado 4 est|í imprimiendo...
  >> Empleados esperando: 5 2
[IMPRESION] Empleado 4 ha terminado de imprimir.
  >> Empleados esperando: 5 2
[IMPRESION] Empleado 5 est|í imprimiendo...
  >> Empleados esperando: 2
[IMPRESION] Empleado 5 ha terminado de imprimir.
  >> Empleados esperando: 2
[IMPRESION] Empleado 2 est|í imprimiendo...
  >> Empleados esperando:
[IMPRESION] Empleado 2 ha terminado de imprimir.
  >> Empleados esperando:
```



===== SIMULADOR DE KERNEL =====

1. Gestión de Procesos
2. Ejecutar
3. Memoria
4. Simular E/S (Impresora)
5. Planificación Disco (FCFS/SSTF)
6. Simular Productor-Consumidor
7. Simular Cena de Filósofos
8. Salir

Seleccione opción: 5

Simulación FCFS:

Secuencia: 95 180 34 119 11 123 62 64

Simulación SSTF:

Secuencia: 62 64 34 11 95 119 123 180

```
===== SIMULADOR DE KERNEL =====  
1. Gestión de Procesos  
2. Ejecutar  
3. Memoria  
4. Simular E/S (Impresora)  
5. Planificación Disco (FCFS/SSTF)  
6. Simular Productor-Consumidor  
7. Simular Cena de Filósofos  
8. Salir  
Seleccione opción: 6
```

```
Simulando Productor-Consumidor...
```

```
Productor produce: 1  
Productor produce: 2  
Productor produce: 3  
Productor produce: 4  
Productor produce: 5  
Consumidor consume: 1  
Consumidor consume: 2  
Consumidor consume: 3  
Consumidor consume: 4  
Consumidor consume: 5  
Productor produce: 6  
Productor produce: 7  
Productor produce: 8  
Productor produce: 9  
Productor produce: 10  
Consumidor consume: 6  
Consumidor consume: 7  
Consumidor consume: 8  
Consumidor consume: 9  
Consumidor consume: 10
```

===== SIMULADOR DE KERNEL =====

1. Gestión de Procesos
2. Ejecutar
3. Memoria
4. Simular E/S (Impresora)
5. Planificación Disco (FCFS/SSTF)
6. Simular Productor-Consumidor
7. Simular Cena de Filósofos
8. Salir

Seleccione opción: 7

```
Simulando Cena de los Fil|sofos...
Fil|sofo Fil|sofo Fil|sofo 1Fil|sofo 4 est|í pensando...
  est|í pensando...
0 est|í pensando...
Fil|sofo 2 est|í pensando...
3 est|í pensando...
Fil|sofo 1Fil|sofo  est|í comiendo...
4 est|í comiendo...
Fil|sofo Fil|sofo 4 ha terminado de comer.
1 ha terminado de comer.
Fil|sofo 1 est|í pensando...
Fil|sofo 2 est|í comiendo...
Fil|sofo 4 est|í pensando...
Fil|sofo 0 est|í comiendo...
Fil|sofo 2 ha terminado de comer.
Fil|sofo 2 est|í pensando...
Fil|sofo Fil|sofo 0 ha terminado de comer.
Fil|sofo 3 est|í comiendo...
1 est|í comiendo...
Fil|sofo 0 est|í pensando...
Fil|sofo 3 ha terminado de comer.
Fil|sofo 3Fil|sofo 4 est|í comiendo...
  est|í pensando...
Fil|sofo 2 est|í comiendo...
Fil|sofo 1 ha terminado de comer.
Fil|sofo 1 est|í pensando...
Fil|sofo 4Fil|sofo 0 est|í comiendo...
  ha terminado de comer.
Fil|sofo 4 est|í pensando...
Fil|sofo 2Fil|sofo 3 est|í comiendo...
  ha terminado de comer.
```

```

Fil| sofo 2 est|í pensando...
Fil| sofo 0 ha terminado de comer.
Fil| sofo 0 est|í pensando...
Fil| sofo 1 est|í comiendo...
Fil| sofo 3 ha terminado de comer.
Fil| sofo 3 est|í pensando...
Fil| sofo 4 est|í comiendo...
Fil| sofo 1 ha terminado de comer.
Fil| sofo 2 est|í comiendo...
Fil| sofo 4Fil| sofo 0 est|í comiendo...
    ha terminado de comer.
Fil|sofo Fil|sofo 3 est|í comiendo...
2 ha terminado de comer.
Fil|sofo 0 ha terminado de comer.
Fil|sofo 3 ha terminado de comer.

```

## **2.7 Resultados y análisis de simulación.**

**Nota : debemos tener presente que ara ejecutar la opción 2 del menú de el simulador de kernel debes crear los procesos con anterioridad en la opción 1 del menú y si quieres volver a ejecutar la opción 2 con un proceso diferente ten en cuenta que al ya haber terminado este proceso que realizaste con anterioridad todos los procesos crados antes se dan por terminados y no se ejecutarn , asi que debes repetir el proceso de creación de procesos si quieres volver a ejecutar la opción 2 del menú de kernel.**

Planificación de Procesos: Round Robin logra distribución justa; SJF mejora tiempos de respuesta global.

Gestión de Memoria: LRU genera menos fallos de página que FIFO en acceso repetido. Las simulaciones muestran los cambios de marcos y la cantidad total de fallos.

Sincronización: El productor-consumidor y los filósofos operan sin bloqueos, gracias a las estructuras de sincronización correctamente aplicadas.

Impresora compartida: Solo un empleado imprime a la vez. Se muestran en consola los estados de espera y acceso.

Planificación de Disco: SSTF reduce la distancia recorrida por el cabezal en comparación con FCFS.

## **3. Diagramas.**

### **Diagrama de módulos**

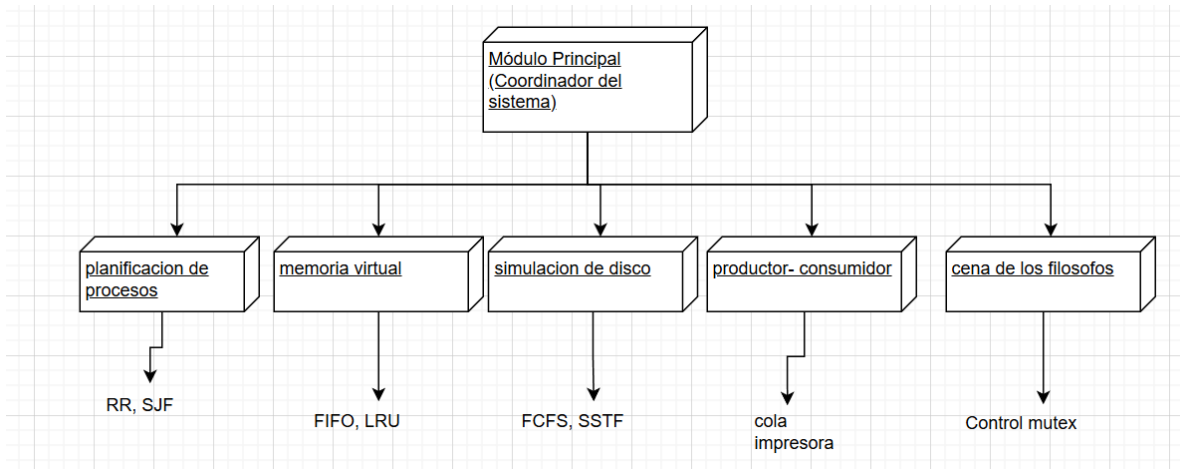


Diagrama de flujo general del sistema

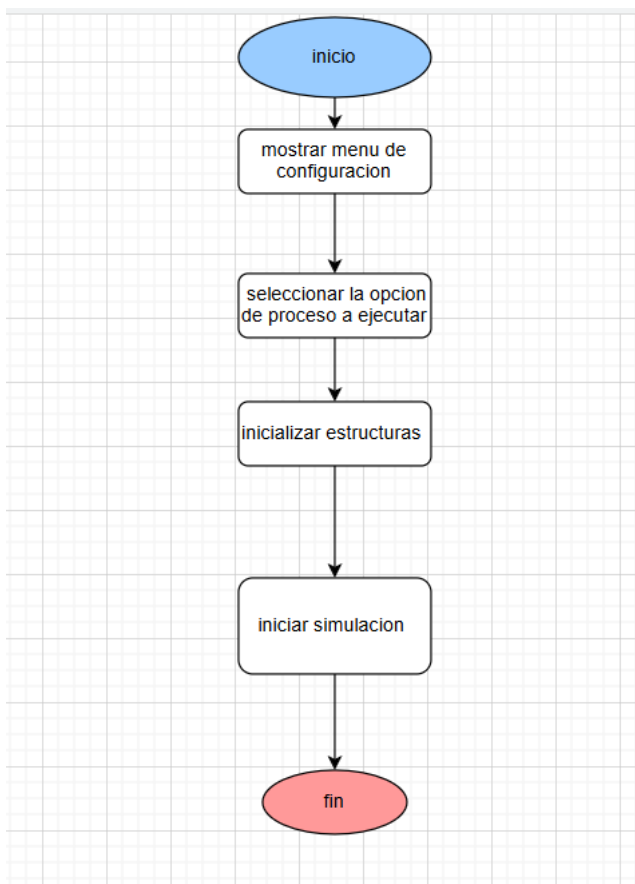


Diagrama de flujo secuencia de paginas



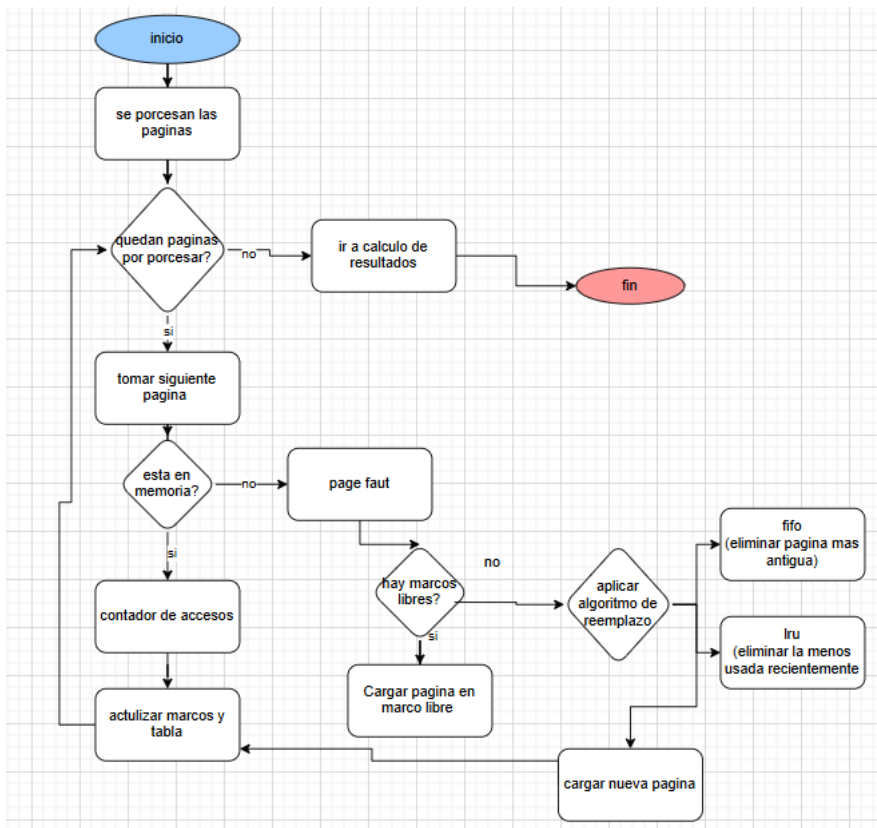


diagrama de flujo planificación de procesos

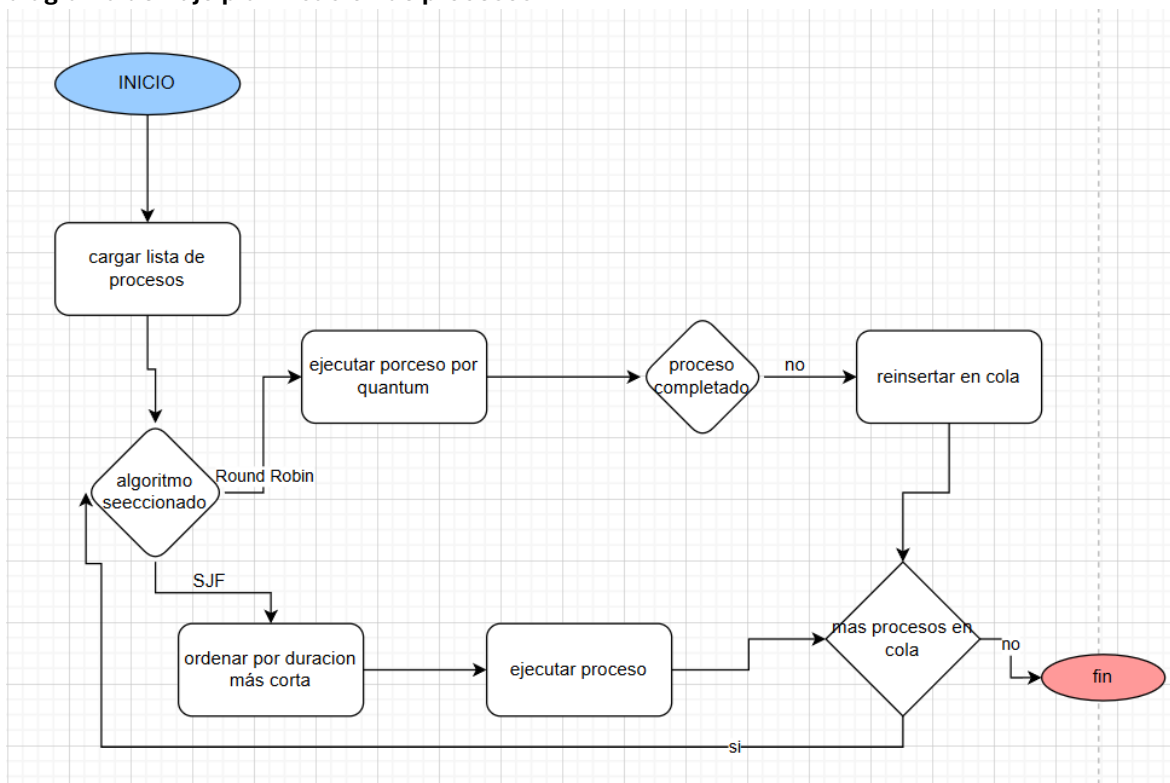
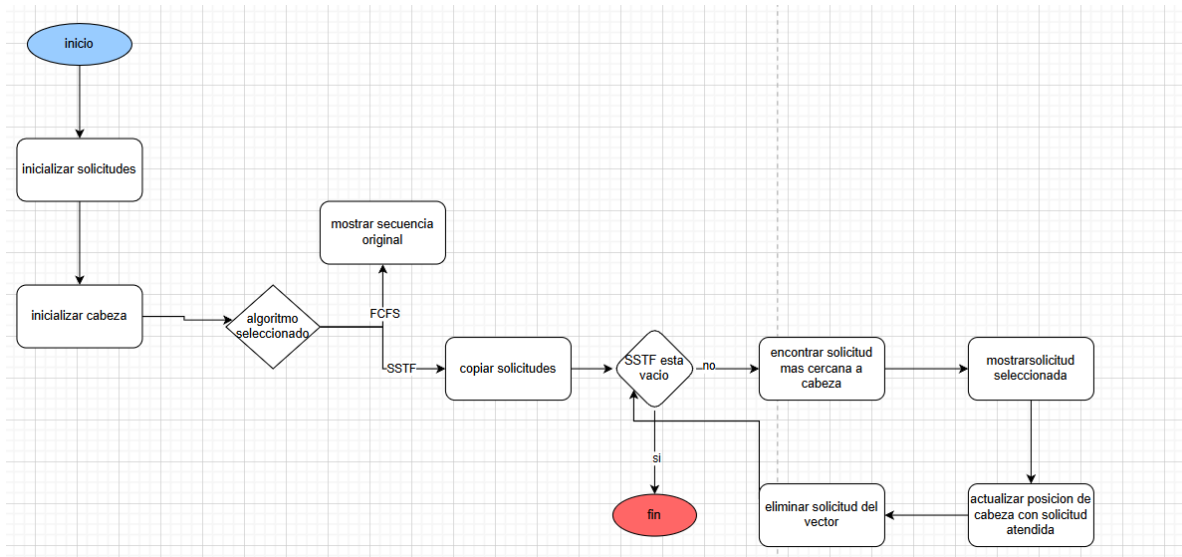


diagrama de flujo planificación de disco



**diagrama de flujo cena de los filósofos**

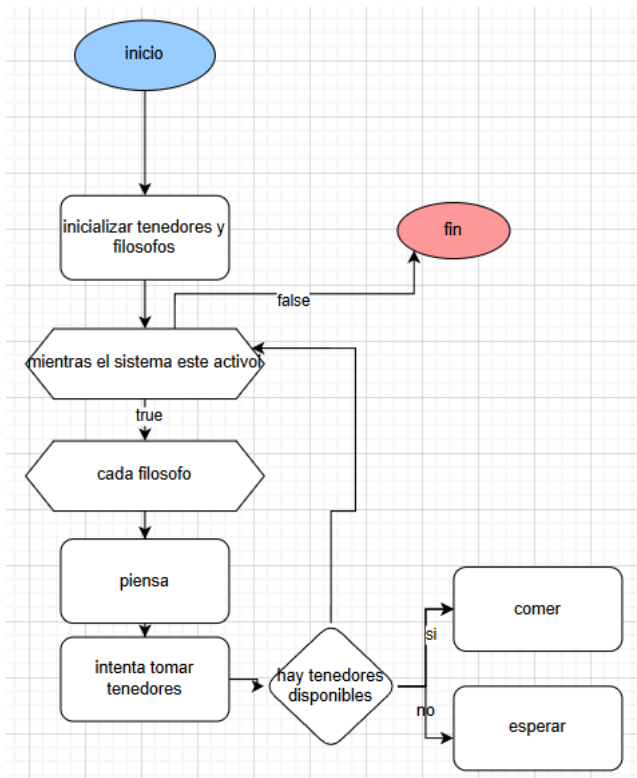


diagrama de flujo simulación impresora

