****

**UNIVERSIDAD PRIVADA DE TACNA**

**FACULTAD DE INGENIERÍA**

**Escuela Profesional de Ingeniería de Sistemas**

**INFORME TÉCNICO**

**“Planificación de CPU y gestión de memoria”**

Curso: Base de datos II

Docente: Msc. Ing. Hugo Manuel Barraza Vizcarra

**Podestá Condori, Richard Podestá (2022075743)**

**Tacna – Perú**

**2025**

**ÍNDICE**

[I. Introducción 4](#_heading=h.gjdgxs)

[Problema: 4](#_heading=h.k7v1nucrrofn)

[Objetivos: 4](#_heading=h.30j0zll)

[Alcance: 4](#_heading=h.1fob9te)

[Supuestos: 4](#_heading=h.a5qb15401ft7)

[II. Marco conceptual 5](#_heading=h.3znysh7)

[III. Diseño del simulador 6](#_heading=h.2et92p0)

[IV. Metodología de experimentos 9](#_heading=h.tyjcwt)

[V. Resultados 10](#_heading=h.vy25w4jnw41a)

[VI. Discusión 12](#_heading=h.8h58eg9sq0b)

[VII. Conclusiones 12](#_heading=h.o8u7gpxz79r4)

**ÍNDICE DE FIGURAS**

[Figura 1 6](#_heading=h.p6v03r9btphj)

[Figura 2 6](#_heading=h.i4iusyx30un4)

[Figura 3 7](#_heading=h.ox7geinsiloq)

[Figura 4 7](#_heading=h.829avwcihwka)

[Figura 5 8](#_heading=h.q26sq9hk3owi)

[Figura 6 9](#_heading=h.1w7g9d1q3t9m)

[Figura 7 10](#_heading=h.i96a8fmdri1)

[Figura 8 11](#_heading=h.doz9yvogv641)

[Figura 9 11](#_heading=h.z99ky6lp7qv2)

**Informe Técnico**

**TEMA: Planificación de CPU y gestión de memoria**

# Introducción

## *Problema:*

* En los sistemas operativos, la correcta gestión de procesos y memoria es esencial para garantizar un uso eficiente de los recursos de hardware. Sin embargo, comprender estos conceptos únicamente de forma teórica puede ser abstracto. Surge la necesidad de un simulador simple que permita observar de manera controlada el comportamiento de procesos, la planificación de CPU y la asignación de memoria.

## *Objetivos:*

* Diseñar e implementar un simulador de sistema operativo en C++ que permita gestionar procesos y memoria.
* Implementar los algoritmos de planificación FCFS, SPN y Round Robin para comparar su desempeño.
* Calcular métricas de rendimiento como tiempo de respuesta, espera, retorno y throughput.
* Simular un esquema de gestión de memoria con particiones variables usando First-Fit y Best-Fit.

## *Alcance:*

* El simulador procesa una lista de procesos definida por el usuario (PID, llegada, servicio), ejecuta los algoritmos de planificación seleccionados y calcula métricas de desempeño. Además, simula la asignación de bloques de memoria para cada proceso y muestra el estado de la memoria después de cada solicitud.

## *Supuestos:*

* Todos los tiempos de llegada y servicio son enteros y conocidos.
* El quantum para Round Robin es fijo y ≥ 2.
* La memoria es lineal y de tamaño configurable; se asigna en el momento de llegada del proceso.

# Marco conceptual

***Procesos y Estados***

* Un proceso es un programa en ejecución, compuesto por su código, datos y el contexto de CPU necesario para continuar su ejecución. Durante su ciclo de vida, un proceso puede encontrarse en distintos estados: nuevo, listo, ejecutando, bloqueado o terminado (Silberschatz, Galvin y Gagne, 2019).

***Planificación de CPU***

* La planificación es la decisión de cuál proceso se ejecutará cuando la CPU esté disponible.
  + FCFS (First Come, First Served): Atiende procesos en orden de llegada y no expropia la CPU. Puede generar el efecto convoy, donde procesos cortos esperan demasiado si hay procesos largos delante.
  + SPN (Shortest Process Next, no expropiativo): Atiende primero al proceso con menor tiempo de servicio conocido. Minimiza el tiempo promedio de espera, pero requiere conocer de antemano la duración del proceso, lo que no siempre es posible en sistemas reales.
  + Round Robin: Asigna la CPU por intervalos de tiempo (quantum) iguales. Si el proceso no termina, se vuelve a encolar. Evita la inanición, pero si el quantum es demasiado pequeño puede generar sobrecarga de cambio de contexto.

***Gestión de Memoria***

* Se utiliza un esquema de particiones variables: la memoria comienza como un único bloque libre y se divide dinámicamente a medida que los procesos solicitan espacio.
  + First-Fit: Asigna el primer bloque suficientemente grande que encuentre.
  + Best-Fit: Asigna el bloque que deje la menor cantidad de memoria libre sin usar (menor fragmentación interna).

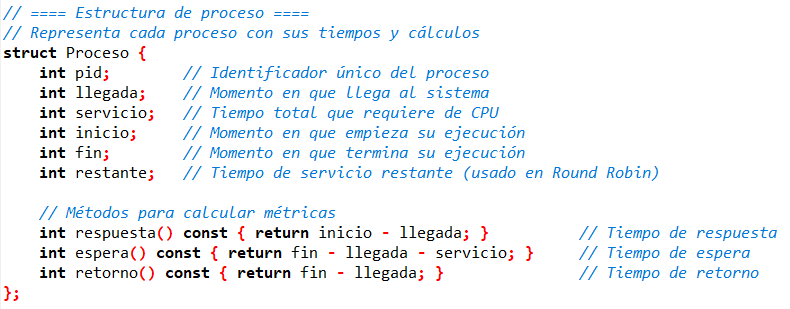
# Diseño del simulador

***Estructuras de datos:***

* Almacena toda la información de un proceso (PID, llegada, servicio, inicio, fin, tiempo restante).
* Tiene métodos internos (respuesta(), espera(), retorno()) que calculan métricas de rendimiento de forma automática.

###### Figura 1

*Estructura proceso con variables y métodos.*

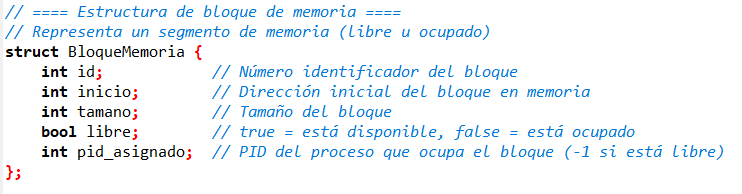


*Nota.* Elaboración Propia.

* struct BloqueMemoria: representa un bloque de memoria (inicio, tamaño, estado).

###### Figura 2

*Estructura BloqueMemoria con variables.*



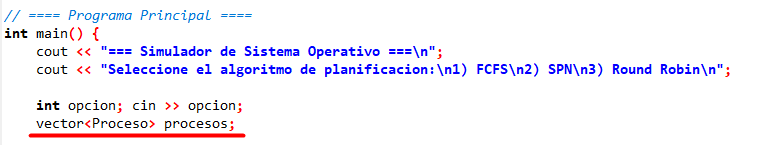
*Nota.* Elaboración Propia.

***Colas y vectores:***

* vector<Proceso> : almacena procesos pendientes y permite ordenarlos (sort) según llegada o tiempo de servicio.

###### Figura 3

*Vector Proceso.*

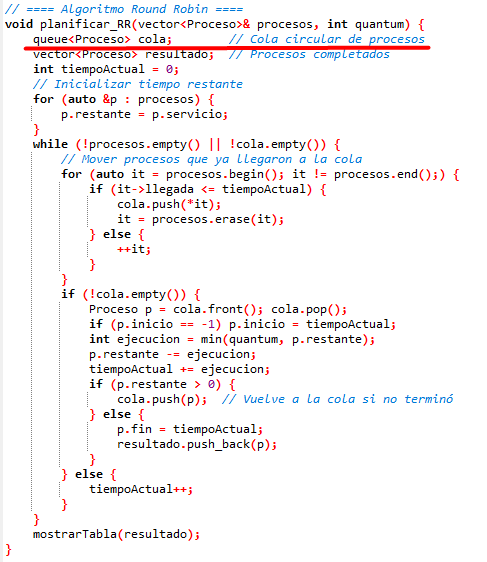


*Nota.* Elaboración Propia.

* queue<Proceso> : en Round Robin, se usa para rotar procesos según el quantum (es decir, simula la cola circular del CPU).

###### Figura 4

*Queue Proceso.*



*Nota.* Elaboración Propia.

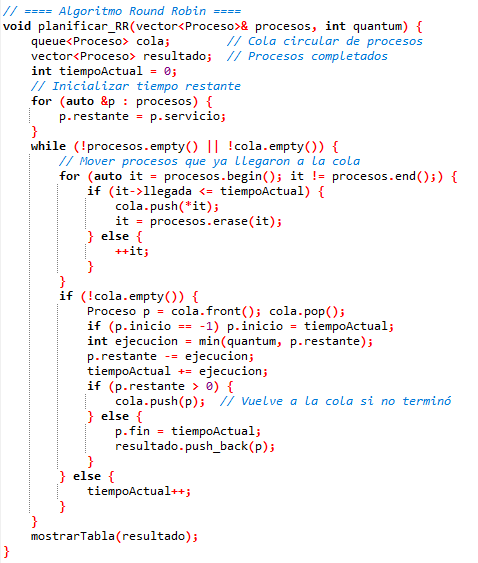
***Manejo de llegada y servicio:***

Los procesos se ordenan por llegada y se despachan según el algoritmo.

En RR, el campo restante indica cuánto tiempo de CPU le queda.

###### Figura 5

*Algoritmo Round Robin*



*Nota.* Elaboración Propia.

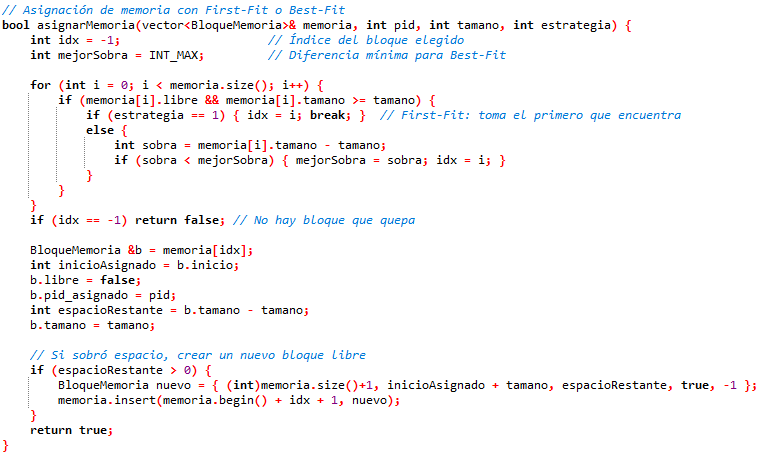
***Módulo de memoria:***

Busca bloques libres, asigna el espacio y divide si sobra.

Muestra visualmente el estado de la memoria tras cada asignación.

###### Figura 6

*Asignación de memoria (First fit-Best fit).*



*Nota.* Elaboración Propia.

# Metodología de experimentos

***Casos de prueba:***

Conjunto de 3 a 6 procesos con llegadas escalonadas y tiempos de servicio variados.

***Parámetros:***

Tamaño de memoria: 1 MiB.

Quantum: 4 unidades de tiempo.

Estrategias de asignación: First-Fit y Best-Fit.

***Repetibilidad:***

Se utilizan los mismos datos de entrada para comparar los tres algoritmos y ambas estrategias de memoria.

# Resultados

***Datos de prueba:***

Procesos: 3

Servicio 1: 12

Servicio 2: 8

Servicio 3: 5

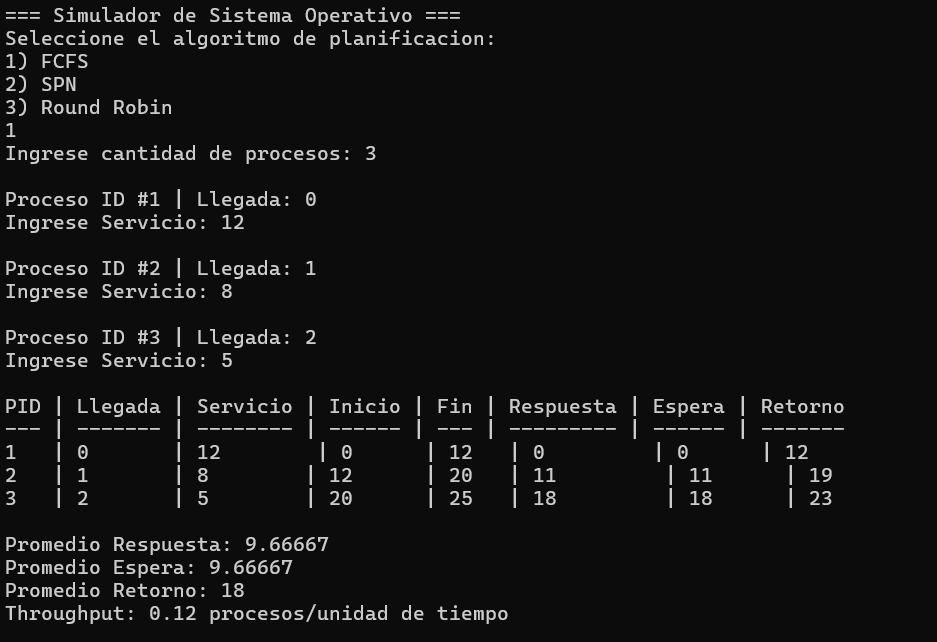
Quantum (RR): 4 unidades de tiempo.

Tamaño de memoria: 10 Mib

***Prueba en planificación FCFS:***

###### Figura 7

*Planificación FCFS.*

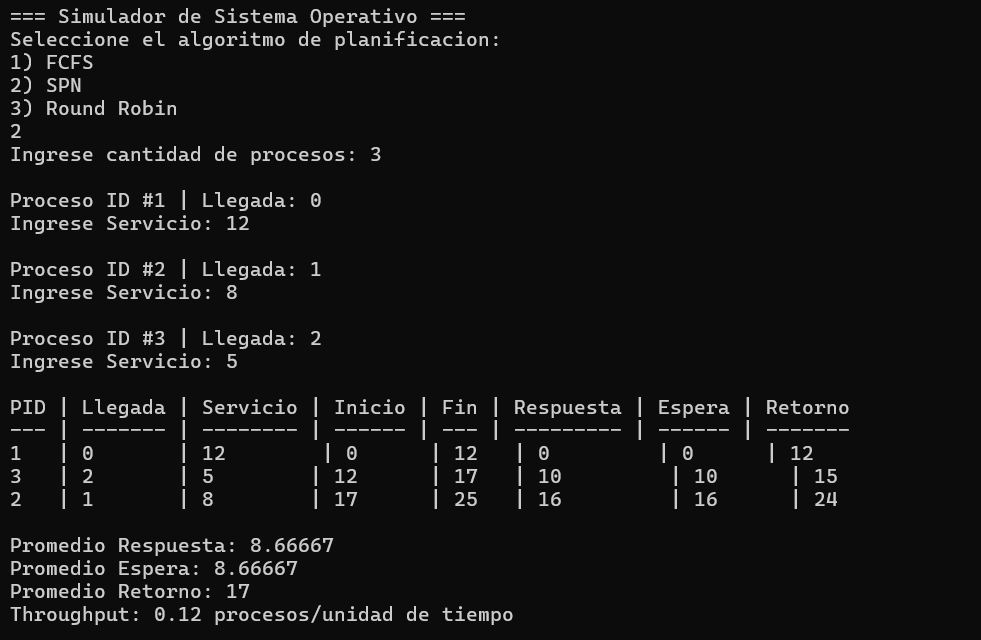
******

*Nota.* Elaboración Propia.

***Prueba en planificación SPN:***

###### Figura 8

*Planificación SPN.*



*Nota.* Elaboración Propia.

***Prueba en planificación RR:***

###### Figura 9

*Planificación RR.*

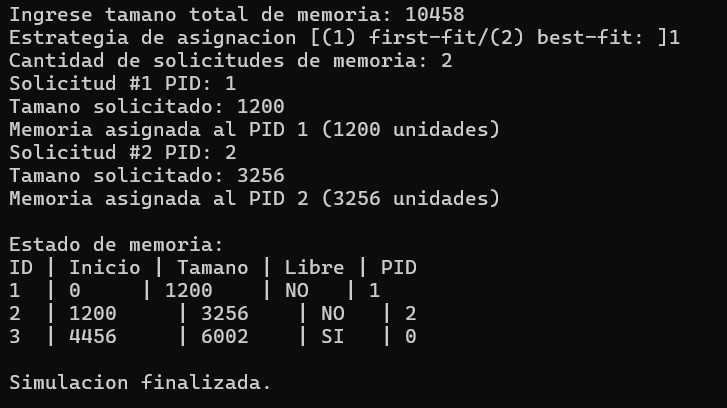


*Nota.* Elaboración Propia.

***Prueba en gestión de memoria:***

###### Figura 10

*Gestión de memoria.*



*Nota.* Elaboración Propia.

# Discusión

# Conclusiones