

IMPLEMENTACIÓN DE UN SISTEMA OPERATIVO



Linux

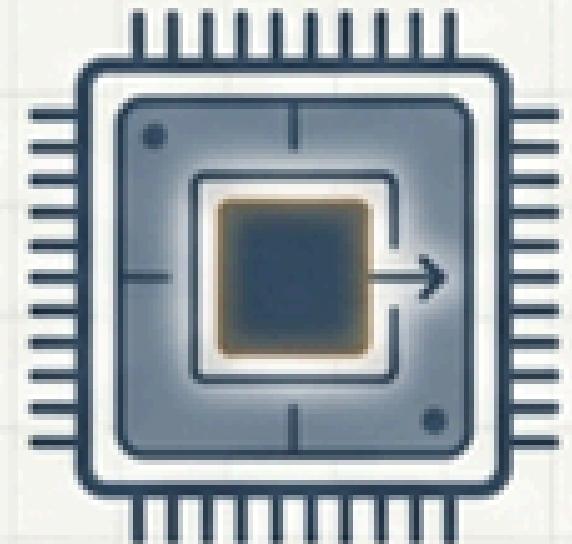
Asignatura:
Sistemas Operativos

Estudiantes:
Maykol David Espinoza Kquerare
Luz Marina Flores Carbajal

Universidad:
Universidad Nacional Jorge
Basadre Grohmann

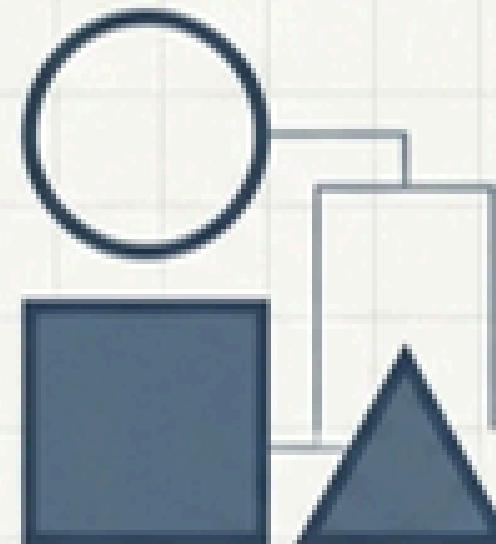
El Desafío Fundamental: Gestionar Recursos Finitos

En el núcleo de todo sistema operativo reside un dilema constante: cómo asignar recursos limitados de manera eficiente y justa. La CPU y la memoria son los campos de batalla donde esta decisión se libra continuamente.



Recursos Limitados

La CPU solo puede ejecutar una tarea a la vez; la memoria principal es finita.



Necesidades Diversas

Los procesos varían drásticamente en sus requerimientos de tiempo de cómputo y espacio de memoria.
simulador_colas.c



Impacto Directo

Las políticas de asignación determinan el rendimiento global, la equidad y la capacidad de multiprogramación del sistema.

Declaración del Problema de Investigación.

El objetivo es evaluar rigurosamente cómo diferentes algoritmos de planificación y técnicas de asignación de memoria influyen en la eficiencia del sistema bajo condiciones controladas.

Nuestro Enfoque: La Creación de un Entorno Controlado

Para comprender estos mecanismos, construimos un simulador. No es solo un programa, sino un laboratorio virtual que nos permite aislar variables y medir su impacto con precisión.

Parámetros del Laboratorio (Las Reglas del Experimento)

Entorno Monoprocesador

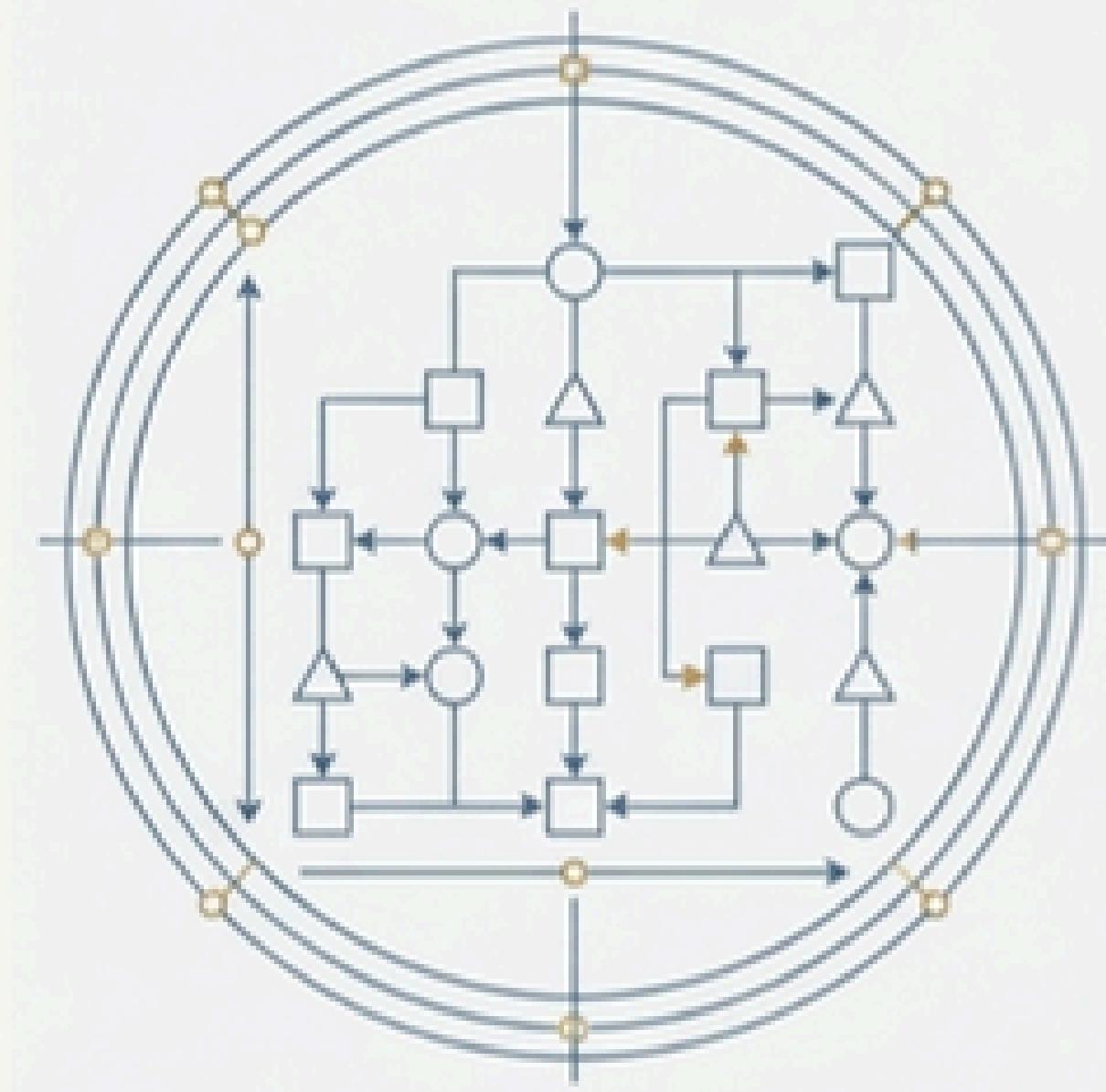
Simulamos una única CPU para enfocarnos en los algoritmos de planificación sin la complejidad de la concurrencia multinúcleo.

Simulación Discreta por Tiempo

Cada paso de tiempo representa un milisegundo, permitiendo modelar llegadas, expropiaciones y finalizaciones con exactitud.

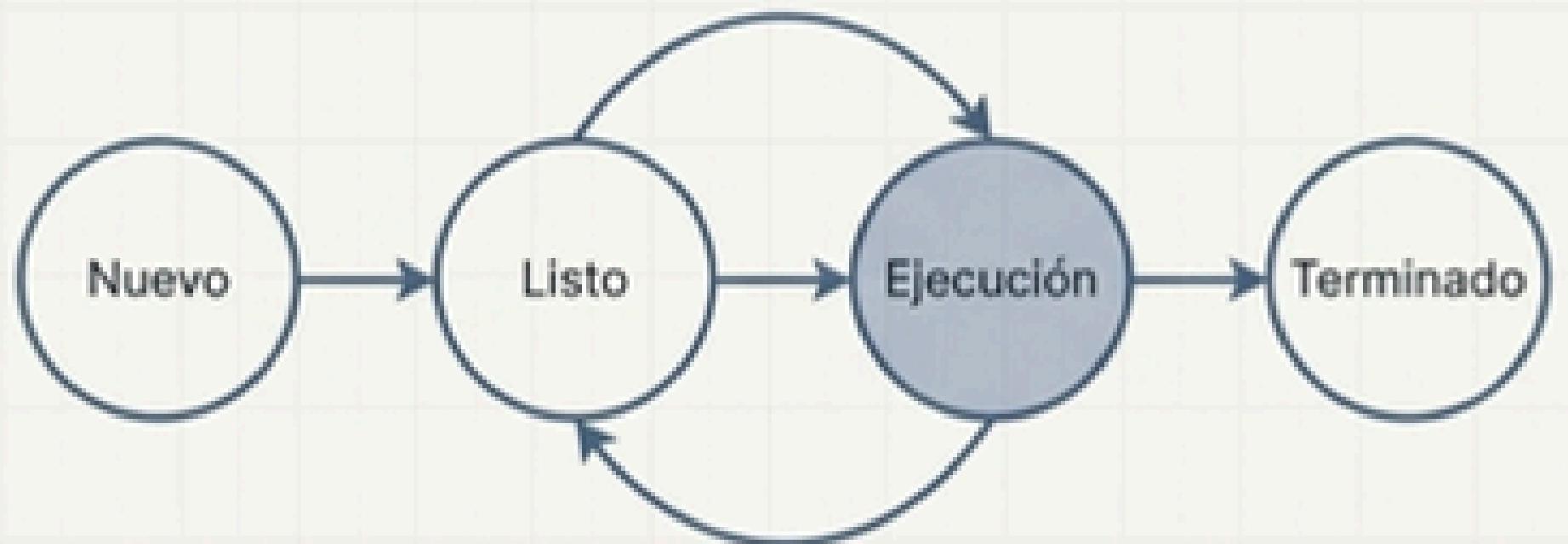
Asignación Contigua de Memoria

La memoria se asigna en bloques lineales. No se implementa paginación ni segmentación para estudiar directamente el efecto de la fragmentación externa.



Los Sujetos de Estudio: El Proceso y su Ciclo de Vida

Definición Clave: Un proceso es la entidad que requiere CPU y memoria para ejecutar un programa. Su información esencial se agrupa en un Bloque de Control de Proceso (PCB).



Nuevo (New): El proceso es leído desde el archivo de configuración (JSON) pero aún no compite por recursos.

Listo (Ready): En la cola de listos, esperando la asignación de la CPU.

Ejecución (Running): Utilizando activamente la CPU.

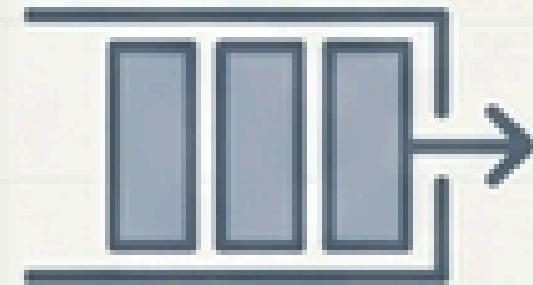
Terminado (Exit): Ha finalizado su ejecución y liberado sus recursos.

“El modelado de los estados facilita la abstracción de la multiprogramación y el análisis de métricas temporales.” — Tanenbaum A. y Bos H. (2015)

El Experimento: Tres Filosofías de Planificación en Competencia

Evaluamos tres algoritmos clásicos, cada uno con una estrategia fundamentalmente diferente para decidir “quién sigue”.

Los Contendientes



FCFS (First-Come, First-Served)

Filosofía: La simplicidad de la cola. El primero en llegar es el primero en ser atendido.

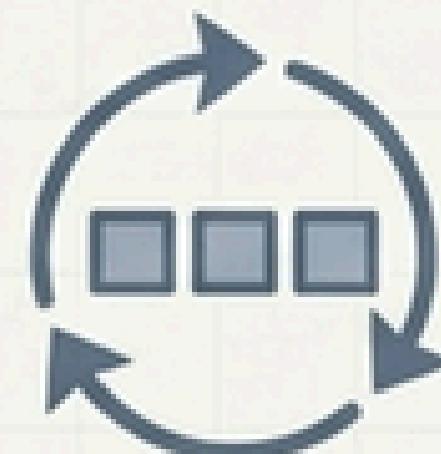
Tipo: No expropiativo.



SPN (Shortest Process Next)

Filosofía: La eficiencia ideal. Prioriza el trabajo más corto para minimizar la espera promedio.

Tipo: No expropiativo.



Round Robin (RR)

Filosofía: La democracia del tiempo compartido. Asigna a cada proceso una porción fija de tiempo (quantum) en un ciclo rotatorio.

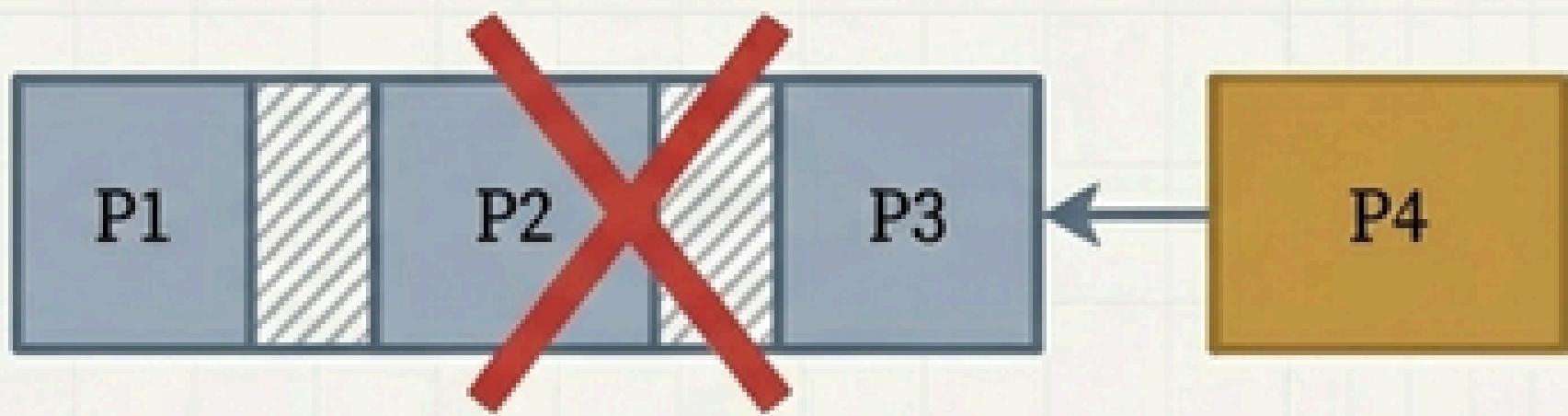
Tipo: Expropiativo.

El Terreno de Juego: La Gestión de un Espacio de Memoria Finito

El simulador utiliza particionamiento dinámico, donde la memoria se asigna a los procesos según su tamaño exacto.

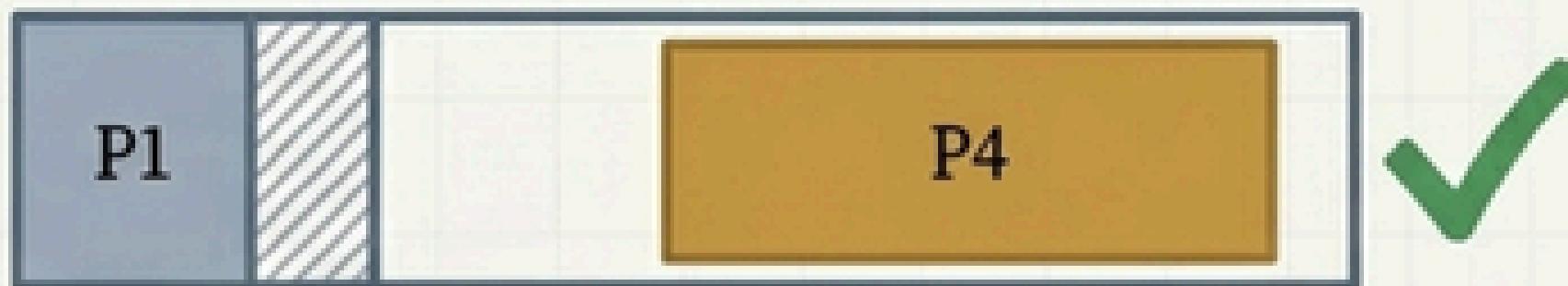
El Problema: Fragmentación Externa

Definición: Ocorre cuando hay suficiente memoria libre total, pero está dividida en pequeños bloques no contiguos, impidiendo alojar un proceso nuevo más grande.



La Solución: Coalescing (Compactación)

Definición: Cuando un proceso termina, su bloque de memoria se libera. Si los bloques vecinos también están libres, el sistema los fusiona en un único bloque mayor.



Diseñando los Instrumentos: Métricas Clave de Rendimiento

Para comparar objetivamente los algoritmos, medimos tres indicadores fundamentales del rendimiento, calculados para cada proceso.

Las Métricas

Tiempo de Respuesta

inicio - llegada

Qué Mide: La rapidez con la que el sistema reacciona a una nueva solicitud. Esencial para sistemas interactivos.

Tiempo de Espera

fin - llegada - servicio

Qué Mide: El “tiempo desperdiciado” que un proceso pasa en la cola de listos. Un indicador directo de la ineficiencia del planificador.

Tiempo de Retorno (Turnaround)

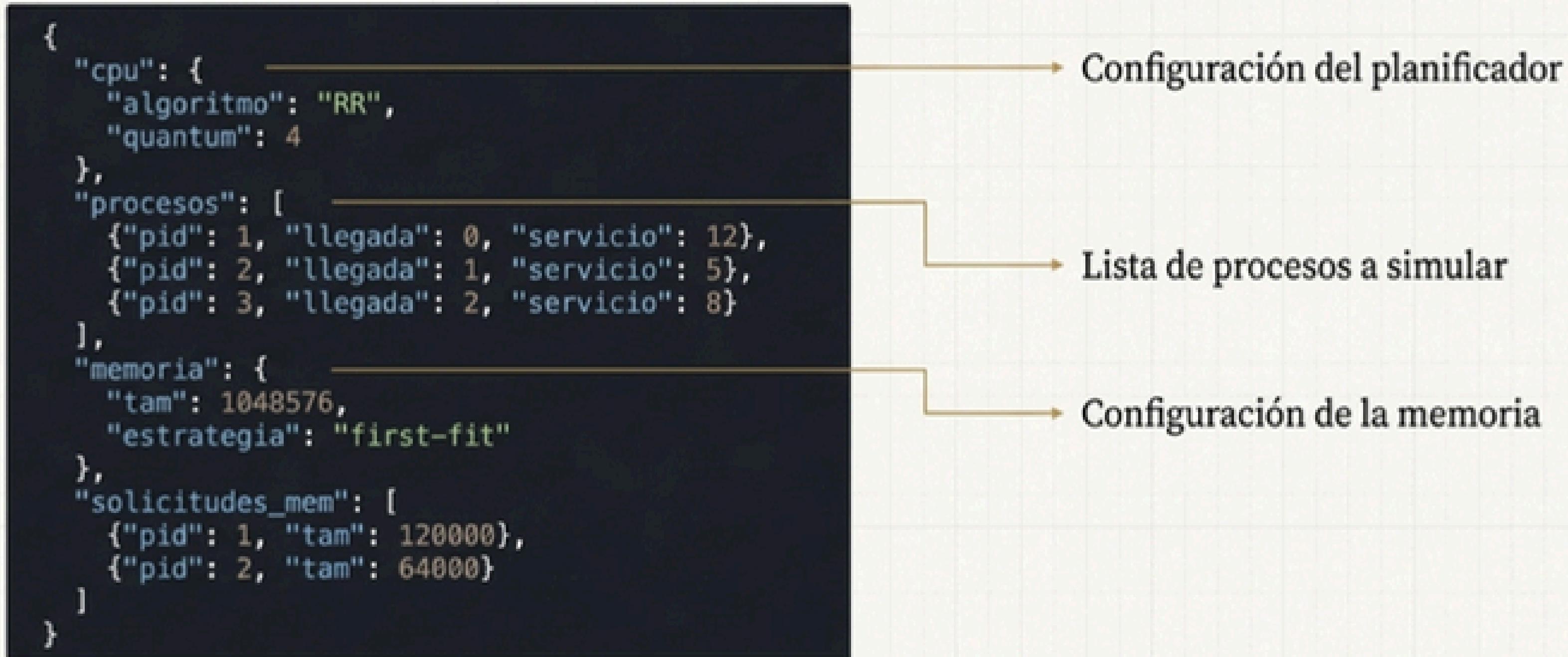
fin - llegada

Qué Mide: La eficiencia total del sistema, desde que un proceso llega hasta que se completa. Clave para sistemas de procesamiento por lotes (batch).

“Estas métricas permiten comparar los diferentes algoritmos en cargas distintas.” – Silberschatz et al. (2018)

El Plan del Experimento: Definiendo las Condiciones Iniciales

Cada simulación se inicia a partir de un archivo de configuración JSON. Esto nos permite definir escenarios de prueba exactos y garantiza la total repetibilidad de los resultados.



- Permite modificar parámetros aislados para observar efectos directos.
- Facilita comparaciones justas entre diferentes algoritmos bajo la misma carga de trabajo.

El Momento del Descubrimiento: Resultados de un Experimento Representativo

Contexto del Experimento: Algoritmo Round Robin con un quantum de 4 ms.

Tabla de Resultados							
Proceso	Llegada	Servicio	Inicio	Fin	Respuesta	Espera	Retorno
P1	0	12	0	25	0	13	25
P2	1	5	4	17	3	11	16
P3	2	8	8	21	6	11	19

Observaciones Clave

- Respuesta Rápida**
P2, aunque llega después, obtiene la CPU rápidamente gracias a la rotación, evitando una larga espera inicial.
- Equidad en la Espera**
A diferencia de FCFS, ningún proceso monopoliza la CPU. P3, con un servicio mayor, no es penalizado y participa equitativamente en la rotación.
- Compromiso Razonable**
Las métricas promedio reflejan un equilibrio entre rapidez y equidad, demostrando la fortaleza de RR en cargas mixtas.

Conclusión del Experimento: La simulación demuestra que Round Robin previene la inanición y reduce el "efecto convoy".

Análisis de los Contendientes: Los Trade-Offs Revelados

FCFS: El Costo de la Simplicidad - El Efecto Convoy

Si un proceso largo llega primero, obliga a todos los procesos cortos posteriores a esperar, degradando drásticamente el tiempo de respuesta promedio.



RPFI: El Costo de la Irradicación - El Efecto Convoy

Si un proceso largo llega primero, obliga a todos los procesos cortos posteriores a esperar, degradando drásticamente el tiempo de respuesta promedio, porque llenan completamente por cortos de rezagarse a la procenta de contenido.

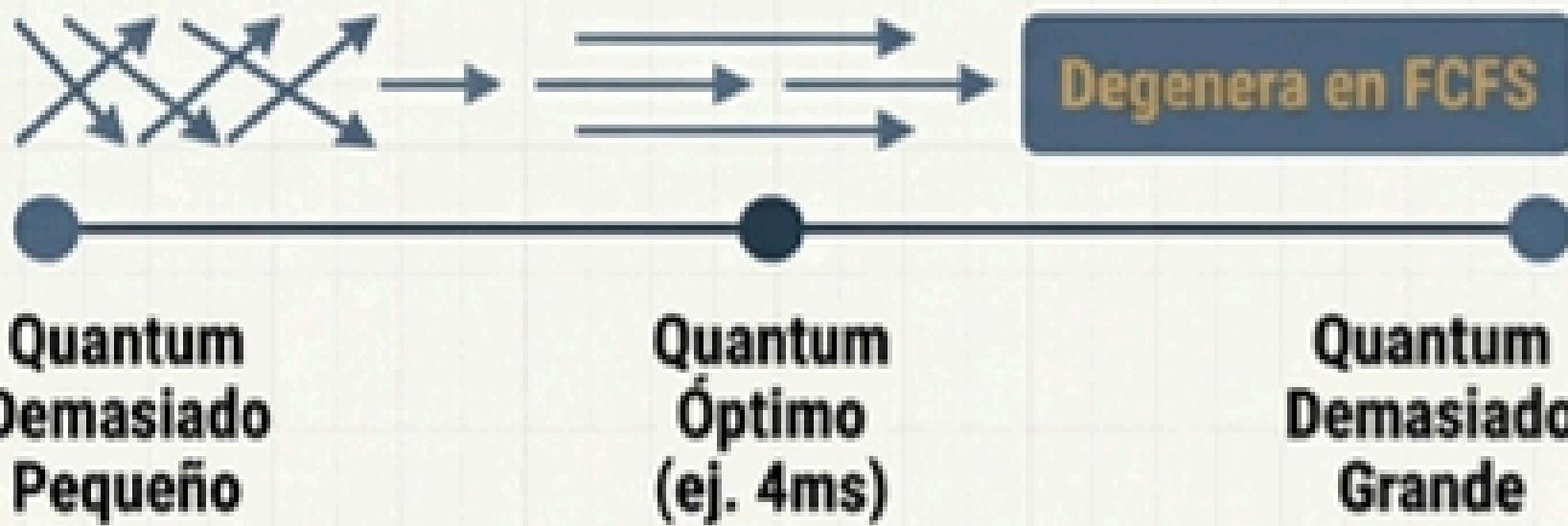
SPN: La Falla del Conocimiento Perfecto

SPN ofrece un rendimiento ideal en nuestra simulación porque conoce la duración del servicio de antemano. En sistemas reales, este dato debe estimarse, y una mala estimación puede llevar a la inanición de procesos largos.



Round Robin: El Dilema del Quantum

El rendimiento de RR depende críticamente de la elección del quantum.



Principios Validados por Nuestro Laboratorio

Nuestro simulador reproduce con éxito los comportamientos descritos en la teoría clásica de sistemas operativos, permitiéndonos validar los siguientes principios:

1. La Equidad es Clave para Cargas Mixtas.

Round Robin es la elección superior cuando la equidad y la capacidad de respuesta son prioritarias.

3. La Simplicidad Tiene un Límite.

FCFS es fácil de implementar, pero es ineficiente y propenso al “efecto convoy” ante procesos de tamaños diversos.

2. La Optimización Teórica Requiere Omnipotencia.

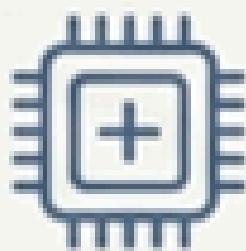
SPN minimiza el tiempo de espera promedio, pero su dependencia del conocimiento futuro lo hace impráctico en su forma pura.

4. La Higiene de la Memoria es Esencial.

El uso de Coalescing es fundamental para combatir la fragmentación y mantener la eficiencia en la asignación de memoria contigua.

Avenidas para la Investigación Futura: Expandiendo el Laboratorio

Este simulador es una base sólida. Las siguientes extensiones permitirían explorar fenómenos más complejos y realistas:



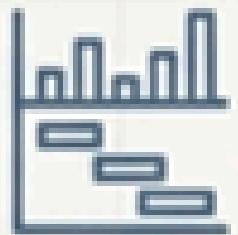
Implementar un Nuevo Contendiente:

Añadir el algoritmo SRTF (Shortest Remaining Time First), la versión expropiativa de SPN, para una comparación más rica entre planificadores preemptivos.



Modelar Comportamiento de E/S:

Permitir que los procesos tengan ráfagas de CPU y de espera por Entrada/Salida para simular cargas de trabajo más realistas.



Mejorar la Instrumentación:

Desarrollar visualizaciones gráficas, como líneas de tiempo de la CPU (diagramas de Gantt) y mapas de ocupación de memoria en tiempo real.



Ampliar los Planes de Experimento:

Extender el formato JSON para permitir la definición de cargas de trabajo más complejas y definición de cargas de trabajo más complejas y dinámicas.

Referencias Bibliográficas

Shore, J. E. (1975). On the external storage fragmentation produced by first-fit and best-fit allocation strategies. *Communications of the ACM*, 18(1), 54–58.

Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). *Operating System Concepts* (10th ed.). Wiley.

Stallings, W. (2018). *Operating Systems: Internals and Design Principles* (9th ed.). Pearson.

Tanenbaum, A. S., & Bos, H. (2015). *Modern Operating Systems* (4th ed.). Pearson.