

Taller práctico: “Administrador de Memoria RAM”

1. Contexto del Taller

Un sistema operativo debe administrar de manera eficiente la memoria RAM disponible para permitir la ejecución de diferentes procesos. Para ello, el gestor de memoria debe **registrar, reservar, liberar y gestionar** los espacios de memoria utilizados por los procesos.

En este reto, los estudiantes asumirán el rol de **administradores de memoria de un sistema operativo** y deberán diseñar una solución que permita simular la asignación y liberación de procesos en una memoria RAM.

2. Propósito

Diseñar e implementar una **simulación de gestión de memoria RAM** que permita representar la asignación y liberación de procesos, identificar la fragmentación de memoria y comparar diferentes algoritmos de ubicación.

3. Situación problema

Un computador dispone de una memoria RAM de **1.000 MB** para la ejecución de procesos.

Durante una jornada de trabajo se solicitan los siguientes procesos:

Proceso	Tamaño
P1	120 MB
P2	250 MB
P3	80 MB
P4	180 MB
P5	150 MB
P6	70 MB

El sistema operativo debe decidir **dónde ubicar cada proceso**, teniendo en cuenta los espacios disponibles.

Posteriormente, algunos procesos finalizarán y dejarán espacios libres. El administrador deberá determinar cómo aprovechar esos espacios para cargar nuevos procesos.

4. Fases del reto

Fase 1. Representación inicial de la memoria

Representar gráficamente una memoria RAM de **1.000 MB**.

Reservar inicialmente **100 MB para el sistema operativo**.

Por lo tanto:

Memoria disponible para procesos = 900 MB

Los estudiantes deberán elaborar una representación indicando:

- Sistema operativo.
- Procesos.
- Espacios libres.
- Dirección inicial.
- Tamaño de cada proceso.
- Espacio ocupado.
- Espacio disponible.

La presentación explica que la reubicación debe considerar inicialmente el espacio ocupado por el sistema operativo.

Fase 2. Asignación mediante Primer ajuste First Fit

Aplicar el algoritmo **Primero en Ajustarse (First Fit)**.

El algoritmo debe recorrer los espacios disponibles hasta encontrar el **primer hueco suficientemente grande** para almacenar el proceso.

Registrar cada operación en una tabla:

Proceso	Tamaño	Hueco seleccionado	Memoria restante
P1	120 MB		

Proceso	Tamaño	Hueco seleccionado	Memoria restante
P2	250 MB		
P3	80 MB		
P4	180 MB		
P5	150 MB		
P6	70 MB		

Fase 3. Liberación de procesos

Ahora deberán simular la finalización de:

- **P2**
- **P4**
- **P3**

Cada proceso liberado debe convertirse en un **hueco disponible**.

Representar nuevamente la memoria.

Identificar:

Fragmentación externa

Determinar si existen varios espacios libres separados entre procesos.

La presentación define la fragmentación externa como el espacio no utilizado que queda entre particiones o al final de la memoria.

Fase 4. Compactación

Realizar una **compactación de memoria**.

El objetivo será reorganizar los procesos para disminuir los espacios libres dispersos y generar un espacio disponible de mayor tamaño. La presentación plantea la compactación precisamente como una estrategia ante la existencia de múltiples fragmentos externos.

Los estudiantes deberán presentar:

Antes de compactar → Después de compactar

y calcular:

- Memoria ocupada.
- Memoria libre.
- Número de huecos.
- Tamaño del hueco más grande.
- Fragmentación externa.

Fase 5. Comparación de algoritmos

Repetir la asignación de los procesos utilizando:

A. First Fit (Primero en ajustarse)

Primer hueco disponible que tenga capacidad suficiente.

B. Best Fit (La mejor opción)

Recorrer la lista completa para encontrar el hueco **más adecuado** para el proceso.

C. Worst Fit (Peor ajuste)

Investigar e implementar la estrategia del **peor ajuste**.

D. Next Fit (Siguiendo ajuste)

Investigar e implementar la estrategia del **siguiendo ajuste**.

La presentación plantea estas dos últimas alternativas como variaciones adicionales para la ubicación de procesos.

6. Producto tecnológico

El reto deberá desarrollarse mediante una aplicación sencilla en **Python, PSeInt, Java, C++ o Excel**, según las herramientas disponibles en el curso.

Como mínimo, el programa debe permitir:

1. Crear la memoria RAM.
2. Registrar procesos.
3. Indicar el tamaño de cada proceso.

4. Asignar procesos.
5. Liberar procesos.
6. Mostrar los espacios ocupados.
7. Mostrar los espacios libres.
8. Calcular la fragmentación.
9. Ejecutar compactación.
10. Comparar los algoritmos de asignación.

7. Representación mediante lista ligada

Como extensión del reto, representar la memoria utilizando una **lista ligada**.

Cada nodo deberá contener:

Tipo	Inicio	Tamaño	Siguiente
SO	0	100 MB	
P	100	120 MB	
H	220	...	
P	...	...	

La presentación plantea precisamente la lista ligada como una forma de registrar procesos y huecos, incluyendo dónde comienza cada elemento, cuánto ocupa y el apuntador al siguiente nodo.

Reto adicional: sistemas asociados

Para profundizar, utilizar una memoria de **1 GB** y simular la ubicación de un proceso de **95 MB** mediante el esquema de sistemas asociados.

Los estudiantes deberán demostrar:

1 GB → 512 MB → 256 MB → 128 MB → proceso de 95 MB

y calcular la fragmentación interna:

128 MB – 95 MB = 33 MB

Este procedimiento corresponde al ejemplo desarrollado en la presentación.

9. Entregables

Cada equipo deberá presentar:

Entregable 1 — Informe técnico

- Introducción.
- Situación problema.
- Objetivo.
- Conceptos utilizados.
- Metodología.
- Resultados.
- Conclusiones.

Entregable 2 — Simulador

Código o archivo funcional de la solución.

Entregable 3 — Evidencias

Capturas de pantalla de:

- Memoria inicial.
- Asignación de procesos.
- Liberación.
- Fragmentación.
- Compactación.
- Ejecución de los algoritmos.

Entregable 4 — Análisis comparativo

Algoritmo	Procesos ubicados	Memoria desperdiciada	Huecos generados	Observaciones
First Fit				
Best Fit				
Worst Fit				

Algoritmo	Procesos ubicados	Memoria desperdiciada	Huecos generados	Observaciones
Next Fit				

Desafío final

¿Cómo puede un sistema operativo administrar una cantidad limitada de memoria RAM de manera que reduzca el desperdicio de espacio y permita cargar la mayor cantidad posible de procesos?

El equipo deberá justificar técnicamente las decisiones tomadas a partir de los resultados obtenidos en su simulación.