

Investigación (%)

Fecha de Entrega: noviembre, 2025

# Administración de Memoria

## Descripción de la tarea

En esta tarea, los estudiantes implementarán un programa en C que **simule** la gestión dinámica de memoria utilizando las funciones `malloc`, `calloc`, `realloc` y `free`, a partir de un archivo de entrada que contiene una secuencia de operaciones. Además, demostrarán problemas como fragmentación de memoria y fugas de memoria.

Para ello, deben solicitar un bloque relativamente grande de memoria al Sistema Operativo y realizar las operaciones correspondientes en ese mismo bloque. Deben llevar control de las asignaciones de memoria que se realizan.

Además, deben implementar los siguientes algoritmos de asignación de memoria:

- Best-fit
- First-fit
- Worst-fit

## Formato del Archivo de Entrada

El archivo debe especificar una secuencia de operaciones de memoria, siguiendo el siguiente formato:

- `ALLOC <variable_nombre> <tamaño>`: Asigna un bloque de memoria de `<tamaño>` bytes y lo asocia a `<variable_nombre>`.
- `REALLOC <variable_nombre> <nuevo_tamaño>`: Reasigna el bloque de memoria de `<variable_nombre>` a un nuevo tamaño.
- `FREE <variable_nombre>`: Libera el bloque de memoria asociado a `<variable_nombre>`.
- `PRINT`: Muestra el estado actual de las asignaciones de memoria.
- Líneas que comienzan con `#` son comentarios y deben ser ignoradas.

Tanto en el `ALLOC` como en el `REALLOC` se debe rellenar toda la memoria con el nombre de la variable.

## Ejemplo de Archivo de Entrada

```
# Simulación de fragmentación
ALLOC A 100
ALLOC B 200
FREE A
ALLOC C 50
PRINT

# Simulación de fuga de memoria
ALLOC D 300
ALLOC E 400
# No liberamos D
FREE E
PRINT
```

## Requisitos del Programa

1. **Lectura del archivo de entrada:** El programa debe aceptar un archivo como argumento, leerlo y ejecutar las operaciones en el orden especificado.
2. **Gestión de memoria:**
  - Implementar asignación dinámica de memoria con `malloc` o `calloc`.
  - Reasignar memoria con `realloc`.
  - Liberar memoria con `free` cuando corresponda.
3. **Simulación de escenarios:**
  - Demostrar fragmentación de memoria.
  - Simular una fuga de memoria.

## Criterios de Evaluación

- **Implementación de operaciones (30%):** Correcta implementación de las operaciones de memoria dinámica (`ALLOC`, `REALLOC`, `FREE`).
- **Algoritmos de asignación (20%):** Implementación y correcta ejecución de los algoritmos Best-fit, First-fit y Worst-fit.
- **Simulación de casos (20%):** Funcionalidad de la simulación de fragmentación de memoria y fugas de memoria.
- **Estructura del código y manejo de errores (15%):** Código bien estructurado, con manejo de errores adecuado.
- **Video de ejecución (10%):** Se debe entregar un video corto (máximo 5 minutos) demostrando la ejecución del programa, mostrando cómo realiza las operaciones de gestión de memoria con el archivo de entrada.

- **Documentación (5%):** Documentación clara y concisa del programa, explicando cómo compilar y ejecutar el código.

## Entregable

Debe presentar un archivo comprimido en el TecDigital que incluya:

- El código fuente del programa.
- El archivo de entrada utilizado para las pruebas.
- Un video corto (máximo 5 minutos) que muestre la ejecución del programa.

La entrega debe realizarse antes de las 10:00pm del día de entrega. Si la entrega se realiza después de la hora establecida, se penalizará con una reducción de 5 puntos porcentuales por cada 24 horas de retraso. Por ejemplo, si entrega a las 10:05pm, su evaluación tendrá una nota base de 95%; si entrega después de las 10:05pm del día siguiente, su nota base será 90%, y así sucesivamente.