

**GRADO EN INGENIERÍA
SOFTWARE SISTEMAS
OPERATIVOS, 2025-2026**

Autores: Alejandro Mamán y Miguel Cabrera.

Tabla de evaluación:

Elemento	Ptos	Realizad o o No Realizad o	Problemas detectados	Si se solucionan los problemas explicar cómo
Repositorio-condición inicial	N/A	Hecho	Sin problemas	
Inicio programa	1	Hecho	Tuvimos problemas a la hora de empezar puesto que no teníamos muy claro de por dónde comenzar	Gracias al uso de Copilot + ChatGPT, ordenamos el orden en el que debíamos realizar las funciones para llegar al programa final
Vuelco contenido caché	1	Hecho	Sin problemas	
Calculo línea caché	1	Hecho	Desplazamiento de bits	Esto fue un poco encrucijada, puesto que primero probamos a hacerlo sin desplazamiento de bits haciendo conversiones de base para poder operar, luego caímos en que se puede usar el

				desplazamiento de bits en C; tras un poco de testing llegamos a una solucion
Escritura del byte leído	0,5	Hecho	Sin problemas	
Sleep	0,5	Hecho	Sin problemas	
Terminación del proceso	1	Hecho	Sin problemas	
Commits	1	Hecho	Sin problemas	
Escritura fichero binario	1	Hecho	Problemas leves	Por culpa de fallos tontos y falta de transcripción al principio nos chocamos un poco con esta parte luego ya caímos en como hacer el parseo
Desarrollo funciones	1	Hecho	Problemas de organizacion	Como ya mencionamos anteriormente tuvimos problemas en el orden de desarrollo de las funciones, además creamos funciones innecesarias, con la ayuda de Copilot entendimos el propósito de ciertas funciones que nos proporcionaba el enunciado y a partir de ahí ya supimos la funcionalidad que debíamos desarrollar para cada una
Limpieza y documentación código	1	Hecho	Sin problemas	
Video	1	Hecho -> Link	Sin problemas	

Prompts

En este práctica se han usado algunos prompts de IA de GPT-4o mini:

1. no existe nullptr en C? Respuesta: En C, no existe el concepto de **nullptr** como en C++. Sin embargo, se utiliza el literal **NULL** para representar un puntero nulo.
Conclusión: Usar NULL en vez de nullptr.
2. Cuando uso free() a donde apunta el puntero liberado? Respuesta: Después de llamar a **free**, el puntero queda en un estado indefinido (no apuntará a ninguna dirección válida). Es una buena práctica asignar el puntero a **NULL** después de liberarlo para evitar accesos accidentales. **Conclusión:** El puntero liberado tiene que apuntar a NULL al liberarlo.
3. cual es la diferencia entre unsigned char y char? Respuesta: **char:** Se considera un tipo de dato **con signo** por defecto. Esto significa que puede almacenar valores desde **-128** hasta **127** (dependiendo de la implementación). **unsigned char:** Es un tipo de dato **sin signo**, lo que permite almacenar valores de **0** a **255**. Esto es útil para representaciones binarias o cuando se necesita más rango positivo.
Conclusión: Para la función de lectura de un binario hay que usar unsigned char.
4. como detecto el final de la fila? Respuesta: Cuando utilizas la función **fgets**, puedes detectar el final de la línea al buscar el carácter de salto de línea (**\n**) o el final del archivo (EOF). **Conclusión:** En los whiles de lectura de archivo usar char = '**\n**' para que llegue al final de la línea.

Enlaces

Para ver el **video explicativo** de la práctica valen cualquiera de estos dos enlaces:

1. [youtube](#)
2. [drive](#)

Y el enlace para github: <https://github.com/miguelandin/megatronix.git>