

# Taller: Implementación de ext2

Sistemas Operativos

16 de octubre de 2025

Segundo cuatrimestre - 2025

## 1. Descripción general

En este taller vamos a completar la implementación de un sistema de archivos **ext2**.

El archivo principal del proyecto es `src/ext2fs`, donde se encuentra la lógica central del sistema de archivos. Es en ese archivo donde deberemos realizar nuestras modificaciones. Además, se incluyen otros archivos que simulan aspectos del funcionamiento del disco, las particiones, etc., los cuales ya están provistos y **no requieren cambios**.

## 2. Ejercicios

### Ejercicio 1: Traducción de bloques a LBA

Implementar la función:

```
get_block_address(struct Ext2FSInode *inode, unsigned int block_number)
```

Esta función debe devolver la dirección **LBA** (Logical Block Address) correspondiente a un bloque de datos, dado un inodo y un número de bloque dentro del inodo. En esta implementación, las direcciones LBA ocupan **4 bytes**.

Se debe contemplar el esquema de direccionamiento de bloques de ext2, incluyendo:

- Bloques **directos**
- Bloque **indirecto simple**
- Bloque **indirecto doble**

No es necesario implementar la triple indirección.

**Recomendaciones:**

- Consultar la estructura `Ext2FSInode` en la carpeta `include`.
- Utilizar la función `read_block`.
- Tener en cuenta el uso de *aritmética de punteros* para acceder correctamente a los bloques de punteros.

## Ejercicio 2: Carga de inodos

Implementar la función:

```
Ext2FS::load_inode(unsigned int inode_number)
```

Esta función debe recibir un número de inodo y devolver un puntero al inodo correspondiente dentro del sistema de archivos.

**Recomendaciones:**

- Utilizar las funciones auxiliares `blockgroup_for_inode`, `blockgroup_inode_index` y `block_group`.
- `block_group` en particular, dado su numero de `block_group`, nos devuelve el descriptor del grupo, para conseguir la posicion de la tabla de inodos.

## Ejercicio 3: Búsqueda de archivos en un directorio

Implementar la función:

```
get_file_inode_from_dir_inode(struct Ext2FSInode *from, const char *filename)
```

Dado un inodo que representa un directorio y un nombre de archivo, esta función debe buscar la entrada correspondiente a ese archivo dentro del directorio. Si el archivo se encuentra, se debe devolver un puntero a su inodo; en caso contrario, se debe retornar NULL.

## 3. Testing

Consultar el archivo README incluido en la carpeta del proyecto para conocer cómo ejecutar los tests que validan la implementación. El tamaño de bloque usado para los tests es de 1024 bytes