

## Manejo e Implementación de Archivos - Proyecto FAT

**Fecha de entrega: 15 de octubre.**

### Objetivo general

Este proyecto tiene como objetivo principal aplicar los conocimientos teóricos y prácticos sobre la administración de archivos dentro del contexto de un sistema operativo, mediante la creación de un programa que simule los mecanismos de control y gestión de archivos basados en **FAT**.

### Objetivos de aprendizaje

- Analiza, descompone y resuelve problemas de forma efectiva, aplicando el concepto de Manejo e Implementación de Archivos.
- Implementa correctamente el uso de librerías de **C# o Python** que permiten la manipulación de archivos y su serialización.
- Desarrolla una solución práctica que replique los principios de organización y control de archivos de **FAT**, incluyendo seguridad y permisos de acceso.

### Descripción

En base al contenido visto en clase sobre Manejo de Archivos y el Sistema de Archivos **FAT**, crear un programa que permita **simular los mecanismos que controlan el sistema de Archivos FAT** para llevar el control de datos que se pueden crear, listar, abrir, modificar, eliminar y recuperar archivos.

Cada vez que el usuario cree un nuevo archivo, se deberá solicitar el **nombre del archivo** y a continuación los **datos** que este contendrá.

A nivel lógico, cada archivo estará representado por una **tabla FAT**, almacenada en un archivo serializado en JSON, con los siguientes campos:

- Nombre del archivo.
- Ruta o nombre del archivo de datos inicial.
- Estado de “papelera de reciclaje virtual” (boolean, false por defecto).
- Cantidad total de caracteres.
- Fechas de creación, modificación y eliminación.
- Owner (propietario) y permisos asignados (lectura, escritura).

### Datos físicos

Los datos físicos del archivo creado por el usuario deberán ser **segmentados en bloques de máximo 20 caracteres**, simulando el funcionamiento de FAT. Cada bloque se almacenará en un archivo JSON con:

- **datos**: contenido del bloque (máx. 20 caracteres).
- **siguiente** archivo: ruta del siguiente bloque.
- **eof**: booleano que indica fin de archivo.

Cada vez que se cree un archivo:

- Se generarán los bloques de datos de 20 caracteres.
- Se creará la entrada correspondiente en la tabla FAT.

Cuando el usuario **modifique un archivo**, el sistema deberá:

1. Leer y mostrar todo el contenido actual.
2. Solicitar el nuevo contenido.
3. Generar nuevos bloques de 20 caracteres y actualizar la tabla FAT.
4. **Eliminar físicamente** los bloques de datos anteriores.

Cuando el usuario **elimine un archivo manualmente**, la bandera de **Papelera de Reciclaje** en la tabla FAT se actualizará a **true**.

### Gestión de permisos

- Existirá un usuario **administrador (owner)** que podrá asignar permisos a otros usuarios:
  - **Lectura**: permite abrir y leer el archivo.
  - **Escritura**: permite modificar el archivo.
- Solo el **owner** podrá asignar o revocar permisos.
- El sistema debe validar los permisos antes de permitir abrir o modificar un archivo.

### Operaciones mínimas del programa

- Crear archivo.
- Listar archivos (excluyendo los eliminados).
- Mostrar archivos en la “papelera” del sistema.
- Abrir archivo (mostrar metadatos y concatenar contenido de todos los bloques).
- Modificar archivo (respetando permisos).
- Eliminar archivo (mover a papelera).
- Recuperar archivo desde la papelera.

## Entregables

- Documento PDF con:
  - Explicación de cómo se replicó la lógica FAT (tabla FAT y bloques de datos).
  - Descripción de la gestión de permisos.
  - Evidencias de pruebas realizadas.
  - Link a un video (máx. 10 min) mostrando y explicando el sistema.
  - Link al repositorio público de GitHub con el código fuente.
- Código fuente completo de la aplicación en una archivo comprimido en ZIP.

## Consideraciones importantes

- Desarrollo individual.
- Entrega únicamente en el portal habilitado.
- Copia parcial o total del proyecto tendrá **0 puntos**.

**Fecha de entrega** Miércoles 15 de Octubre de 2025 durante el período de clase.

## Criterios de Evaluación (10 pts)

### 4 puntos:

Creación de archivos con tabla FAT en JSON (nombre, ruta, papelera, tamaño y fechas).

Segmentación del contenido en bloques de máximo 20 caracteres, con siguiente y eof.

### 2 puntos:

Modificación de archivos: actualizar contenido, crear nuevos bloques y eliminar los antiguos.

### 2 puntos:

Eliminación y recuperación de archivos mediante bandera de papelera.

### 2 puntos:

Gestión de permisos: owner/admin asigna lectura/escritura y solo owner puede cambiar permisos.

**Total: 10 puntos**