

**UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**  
**FACULTAD DE INGENIERÍA**  
**ESCUELA DE CIENCIAS Y SISTEMAS**

**LABORATORIO MANEJO E IMPLEMENTACIÓN DE ARCHIVOS**  
**Proyecto 1**



**201213421**

**JORGE ESTUARDO PUMAY SOY**

## Descripción de la arquitectura del sistema

El presente manual describe el desarrollo e implementación de un sistema de archivos basado en **EXT2**, el cual ha sido simulado utilizando **Go** para el backend y **React** para el frontend. Este proyecto tiene como objetivo permitir comprender el funcionamiento interno de un sistema de archivos, incluyendo la gestión de discos, particiones, inodos, bloques y permisos de acceso.

El backend, desarrollado en **Go**, se encarga de manejar la lógica central del sistema de archivos, permitiendo operaciones como la creación y eliminación de discos, montaje de particiones, gestión de archivos y directorios, administración de usuarios y generación de reportes. Para ello, se han implementado estructuras de datos y algoritmos que replican el comportamiento del sistema de archivos EXT2.

Por otro lado, el frontend, desarrollado con **React**, ofrece una interfaz gráfica intuitiva que facilita la interacción del usuario con el sistema. A través de esta interfaz, los usuarios pueden ejecutar comandos, visualizar el estado del sistema de archivos y obtener reportes en tiempo real.

Este documento proporciona una guía detallada sobre la arquitectura del sistema, su funcionamiento, los comandos soportados y las tecnologías utilizadas. También se incluyen instrucciones para la instalación y uso del sistema, así como ejemplos de operación para facilitar su comprensión y aplicación.

### Estructuras de Datos en el Sistema de Archivos

El sistema de archivos simulado en este proyecto replica la estructura del **EXT2**, utilizando un archivo binario con extensión **.mia** como almacenamiento principal. Para lograr esto, se han implementado diversas estructuras de datos que permiten la organización y gestión eficiente de los archivos y directorios.

A continuación, se describen las estructuras clave utilizadas en la simulación:

#### 1. Master Boot Record (MBR)

El **MBR** es la estructura principal que administra la información del disco y sus particiones. Se encuentra al inicio del archivo **.mia** y contiene:

- Tamaño total del disco.
- Número de particiones.

- Información de cada partición (nombre, tipo, tamaño y estado).

## 2. Extended Boot Record (EBR)

Cuando se utilizan particiones extendidas, el **EBR** actúa como una estructura de enlace para administrar particiones lógicas. Cada **EBR** almacena información sobre:

- La partición lógica asociada.
- Su ubicación en el disco.
- Un puntero a la siguiente partición lógica (si existe).

## 3. Superbloque

El **Superbloque** es la estructura central que administra el sistema de archivos dentro de una partición. Contiene información esencial como:

- Tamaño total de la partición.
- Cantidad de inodos y bloques disponibles.
- Mapa de bits para inodos y bloques.
- Posición del primer inodo y bloque de datos.

## 4. Tabla de Inodos

Los **inodos** almacenan metadatos sobre archivos y directorios, incluyendo:

- Tipo de archivo (directorio o archivo regular).
- Permisos de acceso.
- Usuario y grupo propietario.
- Punteros a los bloques de datos que contienen el contenido del archivo.

Cada archivo o directorio tiene un inodo asociado, lo que permite una gestión estructurada de los datos.

## 5. Bloques de Datos

Los **bloques de datos** almacenan el contenido real de los archivos. Dependiendo del tamaño del archivo, un inodo puede referenciar múltiples bloques de datos. Se manejan distintos tipos de punteros en los inodos:

- **Directos:** Apuntan directamente a los bloques de datos.
- **Indirectos:** Apuntan a bloques que contienen referencias a otros bloques de datos.

## Descripción de los Comandos Implementados

El sistema de archivos soporta una serie de comandos que permiten la administración del disco, la gestión de archivos y la interacción con el sistema. A continuación, se describe cada comando con algunas de sus parámetros principales:

### 1. Administración de Discos

#### MKDISK (Crear un nuevo disco)

##### Sintaxis:

**mkdisk -size=100 -path=/home/user/disk.mia -unit=M**

##### Parámetros:

- size: Tamaño del disco en KB, MB o GB.
- path: Ruta donde se almacenará el archivo binario.
- unit: Unidad de medida (K, M, G).

##### Descripción:

Crea un nuevo archivo **.mia** con el tamaño especificado y lo inicializa con un **MBR**.

##### Parámetros:

- path: Ruta del archivo **.mia** a eliminar.

##### Descripción:

Elimina el archivo de disco especificado.

### 2. Gestión de Particiones

#### FDISK (Administrar particiones)

##### Sintaxis:

**fdisk -size=50 -path=/home/user/disk.mia -name=Part1 -type=P -unit=M**

##### Parámetros:

- size: Tamaño de la partición.
- path: Ruta del archivo **.mia**.
- name: Nombre de la partición.
- type: Tipo de partición (P para primaria, E para extendida, L para lógica).

- unit: Unidad de tamaño (K, M, G).

**Descripción:**

Crea, elimina o ajusta particiones dentro de un disco.

**MOUNT (Montar una partición)****Sintaxis:**

**mount -path=/home/user/disk.mia -name=Part1**

**Parámetros:**

- path: Ruta del disco.
- name: Nombre de la partición a montar.

**Descripción:**

Asigna un identificador único a una partición para permitir operaciones sobre ella.

**MKFS (Formatear una partición)****Sintaxis:**

**mkfs -id=vda1 -type=full**

**Parámetros:**

- id: Identificador de la partición montada.
- type: Tipo de formato (full para limpiar toda la partición).

**Descripción:**

Inicializa el sistema de archivos **EXT2** en una partición montada.

**4. Administración de Usuarios y Grupos****LOGIN (Iniciar sesión)****Sintaxis:**

**login -user=root -pass=123 -id=vda1**

**Parámetros:**

- user: Nombre del usuario.
- pass: Contraseña del usuario.

- id: Identificador de la partición montada.

**Descripción:**

Permite el acceso al sistema de archivos con permisos de usuario.

**LOGOUT (Cerrar sesión)****Sintaxis:**

**logout**

**Descripción:**

Cierra la sesión activa en el sistema de archivos.

**5. Gestión de Archivos y Directorios****MKFILE (Crear un archivo)****Sintaxis:**

**mkfile -path=/home/user/file.txt -size=10 -cont=contenido.txt**

**Parámetros:**

- path: Ruta del archivo a crear.
- size: Tamaño inicial del archivo.
- cont: Archivo de texto con contenido inicial.

**Descripción:**

Crea un nuevo archivo en el sistema de archivos.

**CAT (Leer un archivo)****Sintaxis:**

**cat -file=/home/user/file.txt**

**Parámetros:**

- file: Ruta del archivo a leer.

**Descripción:**

Muestra el contenido de un archivo en la terminal.

## MKDIR (Crear un directorio)

### Sintaxis:

**mkdir -path=/home/user/docs**

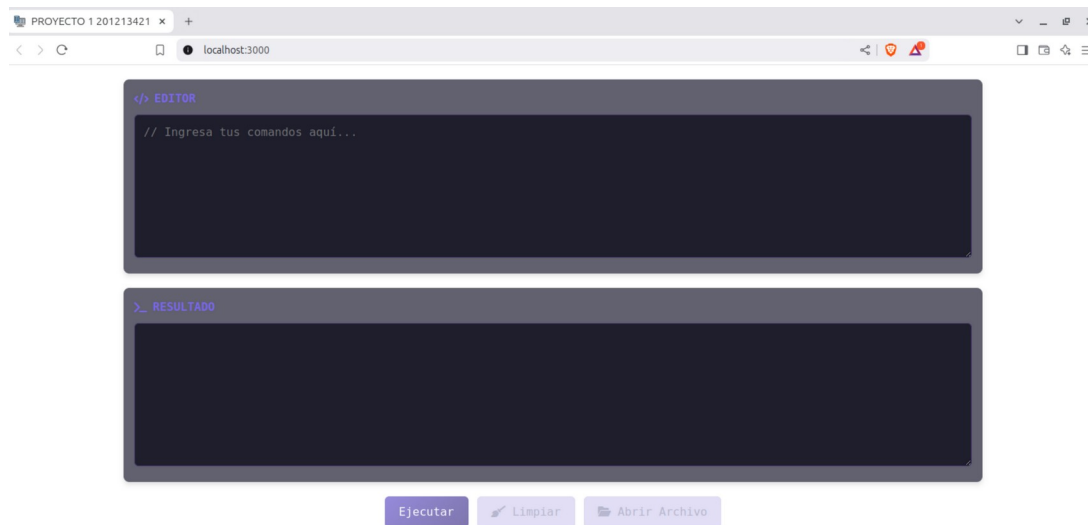
### Parámetros:

- path: Ruta del directorio a crear.

### Descripción:

Crea un nuevo directorio en el sistema.

## Guía para el uso del sistema



Esta es la interfaz de con la que el usuario va a interactuar, consta de 2 contenedores de texto, en la primera que sera editable, el usuario podrá ingresar comandos escribiendo, pegando, o subiendo un archivo completo.

En el contenedor de texto de abajo se verán los resultados de dichos comandos.

También la interfaz consta de tres botones. El botón “ejecutar”, mandara los comandos que se hayan ingresado para que el backend los procese. El botón “limpiar”, limpiara los dos contenedores de texto. Y el botón “Abrir archivo” al presionarlo se abrirá una ventana en la que se podrá escoger un archivo dentro de la computadora, con extensión .smia y se insertara el contenido en la parte del “editor” de comandos.

