

Universidad San Carlos de Guatemala
Manejo e Implementacion de Archivos

Manual Técnico de Proyecto 1

Sergio Sebastian Sandoval Ruiz
202010298
30 de marzo del 2025

Sistema de Archivos EXT2 - Proyecto 1

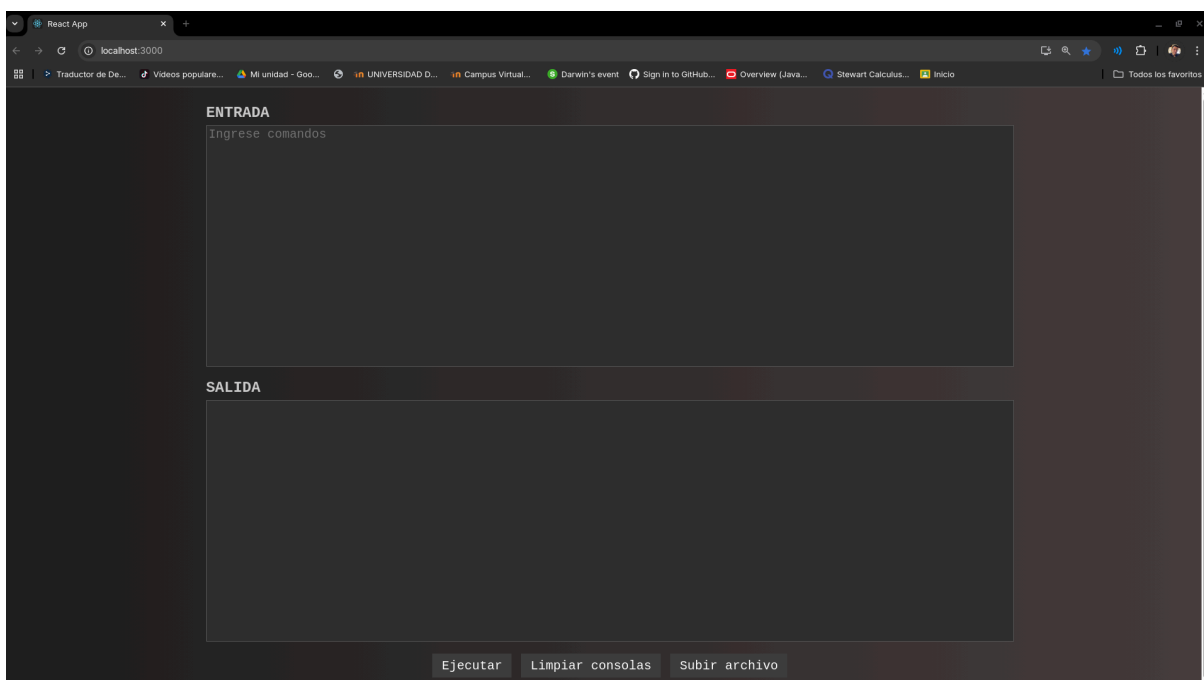
1. Descripción de la Arquitectura del Sistema

La arquitectura del sistema está compuesta por dos capas principales:

Frontend

El frontend está desarrollado con un framework moderno (React). Su función es proporcionar una interfaz de usuario amigable y dinámica que permita:

- Ingresar comandos manualmente o mediante archivos **.smia**.
- Visualizar los resultados enviados desde el backend.
- Mostrar reportes generados por comandos como **REP**.



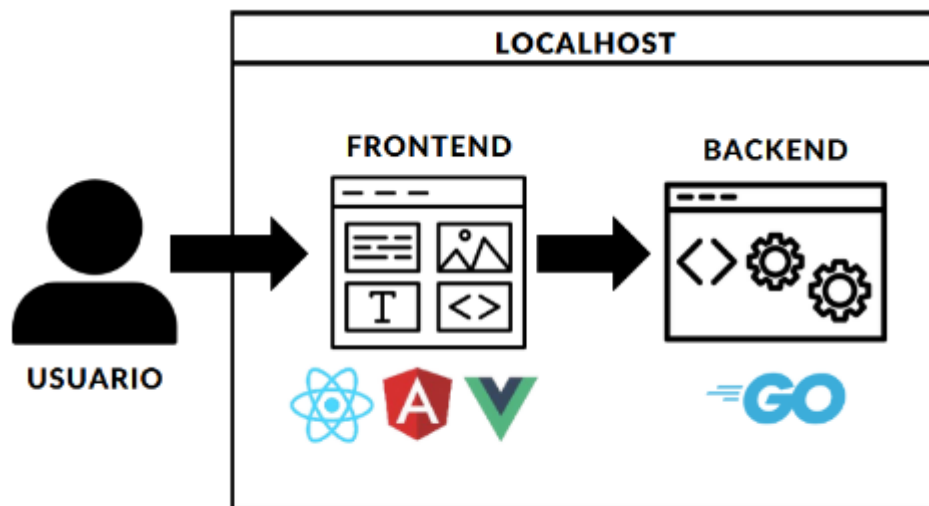
Backend

El backend, desarrollado exclusivamente en Go, es el responsable de:

- Interpretar y ejecutar los comandos recibidos desde el frontend.
- Manipular directamente archivos `.mia` que simulan discos duros.
- Administrar particiones, estructuras del sistema EXT2 y usuarios.
- Generar los reportes en formatos `.jpg`, `.pdf` o `.txt` mediante Graphviz.

Ambas capas se comunican a través de una API RESTful, lo que permite una separación clara de responsabilidades y facilita la escalabilidad del sistema.

Diagrama de Arquitectura



Explicación de las Estructuras de Datos

El sistema simula el sistema de archivos EXT2 utilizando diversas estructuras grabadas dentro del archivo binario `.mia`.

MBR (Master Boot Record)

- Se ubica al inicio del archivo `.mia`.
- Contiene información global del disco como tamaño, fecha de creación, y las particiones.

EBR (Extended Boot Record)

- Representa las unidades lógicas dentro de una partición extendida.
- Actúa como una lista enlazada para representar múltiples particiones lógicas.

Super Bloque

- Es único por partición formateada.
- Almacena información esencial del sistema como:
 - Cantidad total y libre de inodos y bloques.
 - Punteros a las estructuras bitmap, tabla de inodos y bloques.

Inodos

- Cada archivo o carpeta tiene un inodo asociado.
- Contiene permisos, fechas, tamaño, tipo (archivo/carpeta), y punteros a bloques.

Bloques

Existen tres tipos:

- Bloques de carpeta: Guardan `b_content`, que enlaza nombre de archivo y número de inodo.
- Bloques de archivo: Contienen directamente el contenido del archivo.
- Bloques de apuntadores: Referencian bloques indirectos para archivos grandes.

Bitmaps

- Indican qué inodos y bloques están ocupados (1) o libres (0).

3. Descripción de los Comandos Implementados

Aquí se enlistan los comandos implementados, divididos por categoría:

Administración de Discos

- `MKDISK`: Crea un nuevo disco `.mia`.
- `RMDISK`: Elimina un disco.
- `FDISK`: Administra particiones (primarias, extendidas, lógicas).
- `MOUNT`: Monta particiones y genera un ID único.
- `MOUNTED`: Lista particiones actualmente montadas.

Sistema de Archivos

- `MKFS`: Formatea una partición como EXT2 y crea el archivo `users.txt`.
- `CAT`: Lee el contenido de archivos si el usuario tiene permiso.

Usuarios y Grupos

- **LOGIN / LOGOUT**: Inicia o cierra sesión.
- **MKGRP / RMGRP**: Crea o elimina grupos.
- **MKUSR / RMUSR**: Crea o elimina usuarios.
- **CHGRP**: Cambia el grupo de un usuario.

Archivos y Carpetas

- **MKFILE**: Crea archivos.
- **MKDIR**: Crea directorios.

Reportes

- **REP**: Genera reportes visuales y/o textuales del sistema de archivos.
 - Ejemplos: **mbr**, **disk**, **inode**, **block**, **bm_inode**, **tree**, **file**, **ls**, etc.

Script

- Archivos **.smia** permiten ejecutar múltiples comandos de forma automatizada.