# Mini File System Project

## Chapter I - Foreword

Este projeto contínuo tem como objetivo desenvolver um sistema de arquivos minimalista em C, exercitando conceitos fundamentais como inodes, blocos, alocação, exclusão, nomes de arquivos e persistência em disco. Cada etapa é incremental, prática e relacionada a fundamentos de sistemas operacionais.

## Chapter II - Subject

Program name: mini\_fs

Turn in files: Makefile e todos os arquivos fonte necessários

Makefile: Sim

Arguments: Nenhum (o sistema roda com testes em main)

Funções externas autorizadas: printf, memcpy, fopen, fread, fwrite, etc.

Libft autorizada: Não se aplica

* Tarefas atuais:
* • Implementar `read\_file(int inode\_index)`:  
   - Dado um índice de inode válido, imprimir o conteúdo do arquivo bloco a bloco.
* • Implementar `delete\_file(int inode\_index)`:  
   - Marcar o inode e seus blocos como livres nos bitmaps correspondentes.  
   - Resetar os dados do inode.
* • Associar nomes aos arquivos:  
   - Criar uma estrutura simples de diretório: vetor de `{ char name[16]; int inode\_index }`  
   - Permitir gravar arquivos com nome e buscar por nome.
* • Implementar persistência em disco:  
   - Criar `save\_to\_disk(const char \*filename)` para salvar inodes, blocos e bitmaps num arquivo binário.  
   - Criar `load\_from\_disk(const char \*filename)` para restaurar tudo na memória.

## Chapter III - Annex

• Todos os acessos à memória devem respeitar os limites dos vetores.  
• Deve-se validar índices de inode antes de usá-los.  
• Bitmaps devem permanecer consistentes após cada operação.  
• O código deve compilar com as flags: `-Wall -Wextra -Werror`.