

## **Projeto Final - Avaliação A3**

### **A3 - OCSO**

#### **PROJETO DE EXPLORER CUSTOMIZADO**

Relatório técnico apresentado na UC  
Organização de Computadores e  
Sistemas Operacionais pelo Prof. MSc  
Flávio Henrique da Silva.

**Curitiba**

**2025**

## **INTEGRANTES DO GRUPO**

- Luiz Aurio Cordeiro Junior - RA: 172414892
- Matheus Stedile Ferrari - RA: 172311034
- Diego Felipe Gonçalves - RA:172313303
- Gabriel Xavier Leite - RA:172311260
- Andrey Henrique Splett Pires - RA: 172313553

## **SUMÁRIO**

<b>1 INTRODUÇÃO</b>	<b>4</b>
<b>2 DESENVOLVIMENTO</b>	<b>5</b>
<b>2.1 Divisão das Tarefas</b>	<b>5</b>
<b>2.2 Estrutura do Projeto</b>	<b>5</b>
<b>2.3 Explicação da Aplicação/Software</b>	<b>5</b>
<b>2.4 Orientações de execução da Aplicação/Software</b>	<b>5</b>
<b>2.5 Repositório</b>	<b>5</b>
<b>3 CONCLUSÃO</b>	<b>6</b>
<b>REFERÊNCIAS</b>	<b>7</b>

## 1 INTRODUÇÃO

Este trabalho tem como objetivo desenvolver um explorador de arquivos simples no terminal, utilizando Python. A aplicação permite ao usuário navegar por diretórios, visualizar arquivos de texto, copiar e deletar itens, tudo por meio de comandos interativos no console. O projeto visa reforçar o entendimento sobre manipulação de arquivos e diretórios, uso de bibliotecas como os e shutil, além de promover a prática de programação estruturada e interação com o sistema operacional via linha de comando. De forma complementar, foi também implementado um banco de dados liteSQL que salva uma log de edição dos arquivos.

## **2 DESENVOLVIMENTO**

O projeto foi desenvolvido em Python com o objetivo de criar um explorador de arquivos funcional no terminal. Ao longo do trabalho, foram aplicados conceitos de manipulação de arquivos, estrutura de diretórios, leitura de dados e interação via linha de comando e orientação a objetos. Este documento apresenta a divisão das tarefas entre os membros, a estrutura técnica do projeto, o funcionamento da aplicação, as orientações de execução e o repositório contendo o código-fonte.

### **2.1 Divisão das Tarefas**

Aluno 1 (Andrey): Criação da função `list_directory`, responsável por listar e identificar arquivos e pastas.

Aluno 2 (Diego): Desenvolvimento da função `terminal_flow`, incluindo navegação entre diretórios e leitura de arquivos.

Aluno 3 (Xavier): Implementação das funcionalidades de cópia e colagem, com uso de `shutil`.

Aluno 4 (Mateus): Implementação da função de exclusão de arquivos e pastas, e integração com o sistema de logs (`log_operation`).

Aluno 5 (Luiz): Organização geral do projeto, testes, documentação e preparação do repositório GitHub.

### **2.2 Estrutura do Projeto**

A aplicação foi desenvolvida inteiramente em Python, utilizando as bibliotecas padrão `os` (para manipulação de diretórios e caminhos), `shutil` (para operações de cópia e remoção de arquivos/pastas) e `SQLite` (criar um simples DB com as logs) função `log_operation` contida no arquivo `file_logger.py`, que registra as operações realizadas.

Foram respeitados princípios de OOP, tendo o método `main` sido preservado apenas para invocar os métodos contidos em outras classes com métodos e atributos próprios.

As principais funções são:

`list_directory(path)`: Lista os arquivos e pastas de um diretório, identificando visualmente o que é diretório ou arquivo.

`terminal_flow()`: Função principal que controla a navegação, entrada do usuário e operações como abrir, copiar, colar, deletar ou cancelar ações.

`log_operation()`: registra as informações do arquivo que foi deletado ou copiado e salvo em um DB chamado `file_log.db`.

## 2.3 Explicação da Aplicação/Software

Ao executar o programa, o usuário visualiza os arquivos e pastas do diretório atual, numerados para facilitar a navegação. O sistema oferece comandos como:

- Digitar o número para abrir pastas ou arquivos;
- 
- `cp <número>` para copiar um item;
- 
- `paste` para colar no destino;
- 
- `del <número>` para deletar um item;
- 
- `..` para voltar ao diretório anterior;
- 
- `exit` para sair da aplicação.

A interface é simples e orientada por menus e mensagens no terminal, permitindo que o usuário realize ações de forma intuitiva.

## 2.4 Orientações de execução da Aplicação/Software

Para executar a aplicação:

1. Certifique-se de ter o Python 3.6+ instalado;
2. Clone o repositório (link abaixo);

3. Execute o arquivo principal no terminal: `python explorer.py`

A aplicação não exige bibliotecas externas além das nativas do Python.

## **2.5 Repositório**

[https://github.com/ZehD/explorer\\_terminal](https://github.com/ZehD/explorer_terminal)

### **3 CONCLUSÃO**

O desenvolvimento do explorador de arquivos no terminal permitiu aplicar conceitos fundamentais de programação com Python, especialmente no que se refere à manipulação do sistema de arquivos. A aplicação se mostrou funcional e de fácil utilização, oferecendo ao usuário uma interface simples para operações comuns como navegação, leitura, cópia e remoção de arquivos. Além disso, o trabalho promoveu a colaboração entre os integrantes do grupo e contribuiu para o aprofundamento em práticas de organização e documentação de código.



## REFERÊNCIAS

Python Brasil. (2024). Documentação Oficial do Python – Módulo os e shutil.  
Disponível em: <https://docs.python.org/pt-br/3/library/>