

Projeto Final - Avaliação A3

PASTAS E ARQUIVOS

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

Curitiba
2025

INTEGRANTES DO GRUPO

Heloisa Raiacovitch – Ra – 1725110976;

Maria Clara Kaminski Larsen – Ra – 172511409;

Samuel Henrique Wiens – Ra – 172511510.

SUMÁRIO

| | |
|------------|--|
| 1 | INTRODUÇÃO |
| 2 | DESENVOLVIMENTO |
| 2.1 | Divisão das Tarefas |
| 2.2 | Estrutura do Projeto |
| 2.3 | Explicação da Aplicação/Software |
| 2.4 | Orientações de execução da Aplicação/Software |
| 2.5 | Repositório |
| 3 | CONCLUSÃO |
| 4 | REFERÊNCIAS |

1 INTRODUÇÃO

Este projeto visa resolver o desafio comum de gerenciamento de arquivos em sistemas operacionais. O problema central que este trabalho se propõe a solucionar é a necessidade de uma ferramenta de gerenciamento de arquivos mais simplificada e customizável, que ofereça funcionalidades essenciais de forma intuitiva e acessível. Muitas vezes muitos usuários com pouca familiaridade com o sistema operacional ou aqueles que necessitam de operações repetitivas e padronizadas podem se beneficiar de uma aplicação que abstraia a complexidade subjacente do sistema de arquivos, oferecendo uma interface mais direta para ações como abrir, renomear, mover, copiar e excluir arquivos e pastas. Além disso, a capacidade de registrar todas as operações em um log é crucial para auditoria, recuperação de dados e monitoramento de atividades, algo que não é facilmente acessível ou padronizado nas ferramentas nativas dos sistemas operacionais.

2 DESENVOLVIMENTO

Nosso objetivo principal no desenvolvimento deste projeto foi criar um sistema de gerenciamento de arquivos que fosse ao mesmo tempo simples e poderoso. Cada etapa do desenvolvimento foi cuidadosamente planejada, implementada e testada no Eclipse. Para garantir a colaboração e a rastreabilidade, cada atualização era prontamente enviada para o GitHub, assegurando que o projeto permanecesse bem estruturado, fácil de entender e, acima de tudo, funcional. A ideia por trás de tudo era descomplicar a vida de quem precisa organizar seus arquivos, adicionando um recurso essencial: um registro de todas as ações.

Nesta seção, vamos detalhar como tudo isso foi construído. Você vai entender:

- Como nos organizamos para que cada membro da equipe contribuísse da melhor forma.
- A estrutura interna do projeto, as ferramentas que utilizamos e como cada parte do código funciona.

2.1 Divisão das Tarefas

Todos os integrantes da equipe (Heloisa Raiacovitch, Maria Clara Kaminski Larsen e Samuel Henrique Wiens) frequentemente colaboramos em várias seções, e quando não era possível trabalhar diretamente juntos, cada membro da equipe desenvolvia partes específicas. Essas contribuições eram então enviadas para o GitHub, permitindo que todos revisassem e modificassem qualquer porção do código.

2.2 Estrutura do projeto

Nosso projeto foi estruturado para ser organizado e modular, facilitando tanto o desenvolvimento quanto a manutenção. Optamos por uma arquitetura que nos permitisse separar as preocupações, ou seja, cada parte do código tinha uma responsabilidade clara.

A linguagem de programação principal utilizada foi Java. Escolhemos essa linguagem pela sua robustez, portabilidade (graças à JVM) e pelas suas bibliotecas ricas para operações de I/O (entrada/saída), que eram cruciais para o gerenciamento de arquivos.

BIBLIOTECAS E FRAMEWORKS QUE ÚTILIZAMOS:

Java.io. e java.nio.file.: Essas foram as bibliotecas fundamentais para interagir diretamente com o sistema de arquivos. Elas nos permitiram realizar

- operações como listar o conteúdo de diretórios, verificar a existência de arquivos/pastas, obter informações sobre eles e manipular caminhos.
 - Funcionalidade: Fornecem classes para manipulação de arquivos e diretórios, como File, Files e Paths, permitindo listar, criar, excluir, copiar e mover.
- java.time.: Utilizada para obter a data e hora atual, o que foi crucial para registrar o timestamp no nosso sistema de log.
 - Funcionalidade: Gerencia datas e horas de forma moderna e thread-safe, essencial para registrar quando as ações ocorrem no log.
- System.getProperty("user.name"): Embora não seja uma biblioteca separada, essa propriedade do sistema Java foi usada para obter o nome do usuário logado no sistema operacional.
 - Funcionalidade: Permite acessar informações do ambiente do sistema operacional, como o nome do usuário atual.

CLASSES E METODOS DESENVOLVIDOS:

Para manter o código limpo e organizado, criamos algumas classes e métodos principais, cada um com uma função específica:

- GerenciadorArquivos (Classe): Esta é a classe central que encapsula todas as operações de manipulação de arquivos e pastas.
 - GerenciadorArquivos(): O construtor da classe. Inicializa o objeto GerenciadorArquivos.
 - listarDiretorio(String caminho):
 - Funcionalidade: Recebe um caminho de diretório e retorna uma lista de todos os arquivos e pastas contidos nele. É usada para exibir o conteúdo ao usuário.
 - abrirArquivo(String caminhoArquivo):
 - Funcionalidade: Tenta "abrir" e exibir o conteúdo de um arquivo na tela. Para arquivos de texto, ele lê e mostra o conteúdo; para outros tipos, ele pode indicar que o arquivo foi processado.
 - renomear(String caminhoAntigo, String caminhoNovo):
 - Funcionalidade: Altera o nome de um arquivo ou pasta, movendo-o para o caminhoNovo que inclui o novo nome.

- mover(String caminhoOrigem, String caminhoDestino):
 - Funcionalidade: Move um arquivo ou pasta do caminhoOrigem para o caminhoDestino especificado.
- copiar(String caminhoOrigem, String caminhoDestino):
 - Funcionalidade: Cria uma cópia de um arquivo ou pasta do caminhoOrigem para o caminhoDestino.
- excluir(String caminhoAlvo):
 - Funcionalidade: Remove um arquivo ou pasta do caminhoAlvo do sistema.
- SistemaLog (Classe): Responsável por todas as operações relacionadas ao registro de eventos.
 - SistemaLog(String nomeArquivoLog): O construtor. Define o nome do arquivo onde o log será salvo (por padrão, historico_operacoes.txt).
 - registrarAcao(String usuário, String tipo Operação, String caminhoOriginal, String caminhoNovo, String status):
 - Funcionalidade: Este método é chamado sempre que uma operação de arquivo é realizada. Ele captura o timestamp, o usuario, o tipoOperação (ex: "Abrir", "Renomear"), o caminhoOriginal (se aplicável), o caminhoNovo (se aplicável) e o status (se a operação foi bem-sucedida ou falhou), e escreve essas informações no arquivo de log.
- Menu Principal (Classe): Gerencia a interface do usuário e a navegação entre as opções.
 - MenuPrincipal(GerenciadorArquivos gerenciadorArquivos, SistemaLog sistemaLog): Construtor que recebe instâncias do GerenciadorArquivos e SistemaLog para interagir com eles.
 - exibirMenu():
 - Funcionalidade: Exibe as opções do menu principal na tela para o usuário.
 - iniciar():
 - Funcionalidade: O método principal que inicia o loop da aplicação, apresentando o menu, recebendo a escolha do usuário e chamando as funções apropriadas.

Essa estrutura modular nos permitiu focar em cada funcionalidade de forma isolada, garantindo que o código fosse mais fácil de entender, testar.

2.3 Explicação da Aplicação/Software

Nossa aplicação foi pensada para ser um "manual do usuário" por si só, com passos claros e feedback direto. Veja como você pode usá-la:

1. Início da Aplicação: Ao executar o programa, você será saudado com uma mensagem de boas-vindas e o nome do usuário logado no seu sistema operacional. Em seguida, o menu principal aparecerá na tela, apresentando as opções disponíveis.

Bem-vindo ao Gerenciador de Arquivos, [Nome do Usuário]!

```
---                MENU                PRINCIPAL                ---
1.                Abrir                arquivos                ou                pastas
2.                Renomear                arquivos                ou                pastas
3.                Mover                arquivo
4.                Copiar                arquivo
5.                Excluir                arquivos                ou                pastas
6. Criar arquivo
7. Mostrar Uso de Memória
8. Listar unidade de disco
9. Sair
```

Escolha uma opção:

1. Abrir Arquivos ou Pastas (Opção 1): Se você escolher a opção 1, o sistema lhe mostrará algumas pastas padrão (Documents, Videos, Pictures, Download) e as unidades de armazenamento do seu computador (como C: e G:).

```
---                ABRIR                ARQUIVOS/PASTAS                ---
Past
1.                Padrão:
2.                Documents
3.                Videos
4.                Pictures
Download
```

```
Unidades                de                Armazenamento:
5.                C:
```


6. G:
7. Voltar ao Menu Principal

Escolha uma pasta/unidade ou digite o caminho completo:

Você pode selecionar uma dessas opções numéricas ou digitar o caminho completo de qualquer outro diretório que deseje explorar.

Após escolher, o sistema listará todos os arquivos e subpastas dentro do diretório selecionado, como na imagem:

| | | |
|----------|----|-------------------------|
| Conteúdo | de | C:\MeusDocumentos: |
| [F] | | DocumentoImportante.txt |
| [D] | | Relatorios |
| [F] | | FotoViagem.jpg |
| [F] | | Apresentacao.pptx |

Digite o nome do arquivo a ser aberto (ou deixe em branco para voltar):

Agora, basta digitar o nome do arquivo que você quer abrir (ex: DocumentoImportante.txt). Se for um arquivo de texto, o conteúdo será exibido na tela. Para outros tipos, ele pode indicar que o arquivo foi processado.

2. Renomear Arquivos ou Pastas (Opção 2): Escolha a opção 2 no menu principal. O sistema pedirá:

--- RENOMEAR ---
Digite o caminho completo do arquivo/pasta que deseja renomear:
C:\MeusDocumentos\DocumentoAntigo.txt

Digite o novo nome (ex: DocumentoNovo.txt):
DocumentoNovo.txt

Após informar os nomes, você receberá uma confirmação de sucesso ou uma mensagem de erro, caso algo dê errado.

3. Mover Arquivo (Opção 3): Ao selecionar a opção 3, você será guiado para mover um arquivo:

--- MOVER ARQUIVO ---
Digite o caminho completo do arquivo a ser movido:

C:\MeusDocumentos\Relatorio.pdf

Digite o caminho completo do novo diretório de destino:
C:\Backup\Relatorios2024

O sistema então moverá o arquivo do local de origem para o destino.

4. Copiar Arquivo (Opção 4): Similar à opção de mover, mas aqui criamos uma cópia:

--- COPIAR ARQUIVO ---
Digite o caminho completo do arquivo a ser copiado:
C:\MeusDocumentos\Apresentacao.pptx

Digite o caminho completo do novo diretório de destino:
C:\Projetos\ApresentacoesBackup

O arquivo original permanecerá intacto no local de origem, e uma nova cópia será criada no destino.

5. Excluir Arquivos ou Pastas (Opção 5): Ao escolher 5, o sistema exibirá as pastas padrão novamente para facilitar a exclusão:

--- EXCLUIR ARQUIVOS/PASTAS ---
Pastas Padrão para exclusão:
1. Documents
2. Videos
3. Pictures
4. Download
5. Voltar ao Menu Principal

Escolha uma pasta ou digite o caminho completo do arquivo/pasta a excluir:

Você pode selecionar uma pasta padrão para ver seu conteúdo e depois informar o nome do arquivo/pasta a ser excluído, ou digitar diretamente o caminho completo do item a ser removido. O sistema pedirá uma confirmação antes de apagar, garantindo que não haja exclusões acidentais.

- Criar Arquivo (Opção 6): Esta nova funcionalidade permite que você crie um novo arquivo vazio em um diretório específico.

```

---                CRIAR                ARQUIVO                ---
Digite o caminho completo para o novo arquivo (ex:
C:\MeusDocumentos\novo_doc.txt):
C:\Projetos\ListaDeTarefas.txt

```

O sistema criará o arquivo no local e com o nome especificados.

- **Mostrar Uso de Memória (Opção 7):** Ao selecionar a opção 7, o sistema fornecerá informações básicas sobre o uso de memória do sistema operacional em tempo real. Isso pode incluir a memória total, memória livre e memória em uso.

```

---                USO                DE                MEMÓRIA                ---
Memória                Total:                16                GB
Memória                Usada:                8                GB
Memória                Livre:                8                GB
Pressione Enter para voltar ao Menu Principal...

```

- **Listar Unidade de Disco (Opção 8):** Escolhendo a opção 8, o sistema listará todas as unidades de disco disponíveis no seu computador, exibindo seus respectivos nomes, letras (C:, D:, etc.) e, se disponível, informações sobre o espaço total e livre.

```

---                UNIDADES                DE                DISCO                ---
C: (Disco Local) - Total: 500GB, Livre: 200GB
D: (Dados) - Total: 1TB, Livre: 500GB
Pressione Enter para voltar ao Menu Principal...

```

9.Sair (Opção 6): A qualquer momento, você pode digitar 6 no menu principal para encerrar a aplicação.

Todas as ações que você realizar serão registradas no nosso arquivo de log (historico_operacoes.txt), que fica na mesma pasta da aplicação, permitindo que você consulte o histórico de todas as operações.

2.4 Orientações de execução da Aplicação/Software

Para rodar nossa aplicação no seu computador, siga os passos abaixo. É bem simples!

Dependências Necessárias:

Nosso projeto foi desenvolvido em Java. Portanto, a única dependência que você precisa ter instalada no seu sistema é o Java Development Kit (JDK).

- Java Development Kit (JDK) 8 ou superior:
 - Certifique-se de ter o JDK instalado em sua máquina. Você pode baixá-lo no site oficial da Oracle ou em distribuições OpenJDK (como o Adoptium):
 - [Oracle JDK Downloads](#)
 - [Adoptium \(OpenJDK\)](#)
 - As bibliotecas que utilizamos (java.io.*, java.nio.file.*, java.time.*) são bibliotecas padrão do Java, o que significa que elas já vêm com a instalação e não precisam ser instaladas separadamente.

Passo a Passo para Execução:

1. Baixe o Projeto: A primeira coisa é baixar o código-fonte do nosso repositório. Você pode fazer isso de duas maneiras:
 - a. Opção 1: Clonar via Git: Se você tem o Git instalado, abra seu terminal ou prompt de comando e execute:

```
Bash
git clone [LINK_DO_REPOSITORIO]
```

- b. Opção 2: Download Manual: Se preferir, vá até a página do nosso repositório ([Link para o Repositório], que será fornecido na próxima seção), clique no botão "Code" (ou equivalente) e selecione "Download ZIP". Depois, descompacte o arquivo em uma pasta de sua escolha.
2. Navegue até a Pasta do Projeto: Após baixar (e descompactar, se for o caso), use o terminal ou prompt de comando para navegar até a pasta raiz do projeto, onde você encontrará os arquivos .java. Por exemplo:

```
Bash
cd /caminho/para/sua/pasta/do/projeto
```

(Substitua /caminho/para/sua/pasta/do/projeto pelo diretório real onde você salvou o projeto.)

3. Compile os Arquivos Java: Para compilar o código-fonte, use o compilador Java (javac). Se os arquivos .java estiverem organizados em pacotes, você precisará compilar a partir da pasta raiz que contém a estrutura de pacotes. Assumindo que os arquivos estão na pasta raiz ou em uma subpasta como src:

```
Bash
javac *.java
# Ou se estiver em src:
javac src/*.java
```

Se houver múltiplos arquivos e pacotes, o ideal é usar uma IDE como o Eclipse (onde o projeto foi desenvolvido) ou um sistema de build como o Maven/Gradle para compilar. Para um projeto simples, os comandos acima devem funcionar.

4. Execute a Aplicação: Após a compilação bem-sucedida, os arquivos .class serão gerados. Agora, você pode executar a aplicação principal usando o comando java. Assumindo que a classe principal com o método main se chama MenuPrincipal:

```
Bash
java MenuPrincipal
```

5. Interaja com o Menu: A aplicação será iniciada e o menu principal aparecerá no seu terminal. Agora é só seguir as instruções na tela, digitando os números correspondentes às opções e as informações solicitadas.
6. Verifique o Log: Enquanto você usa a aplicação, o arquivo de log (historico_operacoes.txt) será criado (se não existir) e atualizado na mesma pasta onde o executável do programa está. Você pode abri-lo com qualquer editor de texto para ver o histórico das suas ações.

Pronto! Seguindo esses passos, você conseguirá executar e testar nosso sistema de gerenciamento de arquivos.

2.54 REPOSITÓRIO

Você pode encontrar todo o código-fonte do nosso projeto e colaborar (se desejar!) acessando nosso repositório no GitHub:

<https://github.com/Heloisaraiacovitch/projeto-flavio-a3>

Lá, você terá acesso a todos os arquivos, poderá ver o histórico de *commits*, abrir *issues* para sugestões ou *bugs*, e até mesmo propor melhorias

3.0 CONCLUSÃO

Este projeto nos permitiu construir um sistema de gerenciamento de arquivos funcional e acessível, que atende à necessidade de uma ferramenta mais simplificada e com recursos de auditoria. Conseguimos implementar todas as funcionalidades propostas – desde a navegação intuitiva até a renomeação, movimentação, cópia, exclusão e agora, a criação de arquivos, a exibição do uso de memória e a listagem de unidades de disco. O sistema de log se provou um diferencial importante, oferecendo uma trilha detalhada de todas as operações, o que é valioso para controle e segurança.

A experiência de desenvolvimento nos proporcionou um aprofundamento na interação com o sistema de arquivos de um sistema operacional e na importância de um design robusto e de fácil usabilidade. A colaboração contínua da equipe, com a prática de programação conjunta e o uso do GitHub para versionamento e revisão, foi fundamental para o sucesso e a qualidade do código.

4.0 REFERÊNCIAS

<https://www.mygreatlearning.com/blog/what-is-file-handling-in-java/>

<https://www.robsonmartins.com/content/info/java/fundamentos.php>

Auxílio demais na construção do código em geral.