

# Relatório do Projeto

Ricardo Pinto Nº202200637

Miguel Borges Nº202200252

Miguel Pinto Nº202200964

Rodrigo Maduro Nº202200166

- **Introdução**

O objetivo do projeto é desenvolver uma aplicação gráfica em JavaFX que simula a gestão de um sistema de arquivos chamado PFS. Utilizando o ADT Tree como Tipo Abstrato de Dados, a aplicação permitirá operações básicas sobre arquivos, como criação, edição, cópia e exclusão. O sistema de arquivos será modelado como uma árvore, com diretorias e ficheiros como nós. A persistência da informação será implementada seguindo princípios de orientação a objetos e padrões de software.

- **Funcionalidades**

Funcionalidade	Função
Criar	Cria ficheiros e pastas pelo nome e insere na PFS como filho numa posição especificada e retorna essa posição
Copiar	Copia um documento na posição especificada para a área de transferência. Retorna a posição do documento copiado.
Mover	Move um documento da posição inicial para a posição final na estrutura do sistema de arquivos. Lança exceção se a operação de movimentação for inválida.
Renomear	Renomeia um documento na posição especificada. Lança exceções se a posição for inválida ou se o novo nome for nulo.
Remover	Remove um documento na posição especificada do sistema de arquivos. Lança exceção se a posição for inválida.
Visualizar	Exibe o conteúdo de um arquivo na posição especificada. Lança exceção se a posição for nula ou se o documento não for um arquivo.

Editar	Edita o conteúdo de um documento na posição especificada. Lança exceção se a posição for nula ou se o documento não for acessível.
--------	---

- Modo de utilização

A interface “user” contém a TreeView que representa visualmente a estrutura do PFS, exibindo diretórios e arquivos em forma de árvore.

É composta também por um painel de detalhes que exibe informações detalhadas sobre o item selecionado na TreeView como a data de criação, data da última alteração, número de alterações registadas, dimensão, status de bloqueio entre outros.

Adicionalmente possui um painel de conteúdo (mostra ou permite a edição do conteúdo do arquivo selecionado) e uma ToolBar, ou seja, uma barra de ferramentas que contém botões para realizar operações sobre diretórios e arquivos, como criar, copiar, mover, renomear, remover, visualizar e editar.

A interface é projetada para ser intuitiva e fornecer uma experiência amigável ao usuário, permitindo interações diretas com os elementos do PFS. As informações detalhadas e opções de operação são apresentadas de forma clara, facilitando a utilização das funcionalidades oferecidas pela aplicação.

- Arquitetura lógica da aplicação

Package	Classes
Backup	FileBackupHandler, Memento(Interface), Originator(Interface), PFSCaretaker
Commands	BackupCommand, Command(Interface), CommandManager, CopyCommand, CreateFileCommand, CreateFolderCommand, EditCommand, ExtractCommand, MoveCommand, PasteCommand, RemoveCommand, RenameCommand, ZipCommand
Exceptions	DocumentNotFoundException, InvalidDocumentException, InvalidFileException, InvalidMoveException, InvalidNameException, NoMementoException
Factories	DocumentFactory(Interface), FolderFactory, MyFileFactory
Log	ProcedureLogger
View	AlertUtils, DocumentGridItem, DocumentGridManager, EditController, EditGUI, GUI, GUIController,

	InfoController, InfoGUI, PropertiesController, PropertiesGUI, RenameController, RenameGUI
--	---

- Tipos Abstratos de Dados

A classe PFS mantém uma árvore (TreeLinked<Document> pfs) para representar a estrutura hierárquica do sistema de arquivos.

Cada nó da árvore é um Document, que pode ser um arquivo (MyFile) ou uma pasta (Folder).

O ADT Position na classe PFS é usado para representar as posições dos documentos na estrutura de árvore do sistema de arquivos. A interface Position fornece métodos para acessar informações sobre o documento em uma posição específica e para navegar pela árvore.

- Padrões de Software

Padrão	Função
Memento	O Padrão Memento é um padrão comportamental que permite capturar e externalizar um estado interno de um objeto, de modo que o objeto possa ser restaurado para esse estado mais tarde.
Command	O Padrão Command é um padrão comportamental que encapsula uma solicitação como um objeto, permitindo parametrizar clientes com diferentes solicitações, enfileirar solicitações e suportar operações reversíveis.
Factory	O Padrão Factory é utilizado para delegar a criação de objetos para subclasses, o que permite que uma classe adie a instância exata dos objetos para suas subclasses.

MVC	O padrão MVC foi utilizado para fazer a separação do Model, View e Controller
-----	---

- Refactoring

Code Smell	Classe	Refactoring
Long Method	GUI	Extract method
Data Class	Clipboard	Adição de metodos uteis no futuro
comments	Na maioria das classes	Remoção de comentarios desnecessarios, visto que o javadoc está presente
Duplicate code	PropertiesController	Remoção de codigo sem sentido

- Trabalho Realizado

Rodrigo Maduro	25%
Miguel Borges	25%
Miguel Pinto	25%
Ricardo Pinto	25%

O trabalho foi repartido igual por todos e todos contribuíram um pouco para a cada função, ouve um elevado espirito de solidariedade para nos ajudarmos uns aos outros de modo a concluir este projeto.

- Avaliação Crítica

Ao finalizar este projeto de Programação Avançada, avaliamos nossa performance como bastante positiva. Reconhecemos que há aspectos que poderíamos ter aprofundado e aprimorado, como a melhoria da interface gráfica. Embora ela tenha alcançado um bom padrão, acreditamos que poderia ser mais funcional e otimizada para o usuário. Além disso, a representação das estatísticas também é um ponto que poderíamos ter aprimorado.

Gostaríamos de expressar nosso agradecimento aos professores da disciplina de Programação Avançada, com especial destaque ao docente de Laboratório, pela fundamental ajuda proporcionada na execução deste projeto que nos permitiu elevar as nossas capacidades de programação a um patamar superior.