

Documentação Trabalho Final Orientação a Objetos – Sistema de Banco

JEAN FELIPE MARTINHO – 201835022

Link repositório - https://github.com/JeanFDev/TrabalhoFinal_Banco.git

Este documento descreve a implementação do sistema bancário desenvolvido em Java 21 em projeto Maven. O sistema simula as principais funcionalidades de um banco, incluindo cadastro de clientes, contas, investimentos e operações financeiras.

O sistema foi desenvolvido utilizando a arquitetura em camadas, com as seguintes camadas principais:

- **Camada de Interface** responsável pela interação com o usuário, utilizando Swing para a criação de interfaces gráficas.
- **Camada de Negócio** contém as classes que implementam a lógica de negócios do sistema, esta camada foi dividida nos pacotes usuário e entidades com o intuito de organizar e formalizar o código, nelas estão presentes classes como Cliente, Conta, Gerente, Caixa e RendaFixa.
- **Camada de Persistência** responsável pela persistência dos dados em arquivos CSV, utilizando a classe Persistencia.
- **Camada de Exceções** contém as classes de exceções personalizadas, como SaldoInsuficienteException e SenhaIncorretaException.
- **Camada de Utilidades** contém classes de utilidades, como GerenciadorContas que foi criada para garantir que os números de conta sejam únicos e sequenciais.
- **Camada de Testes** contém classes de testes unitários JUnit.

Conceitos de Orientação a Objetos

- **Encapsulamento:** As classes encapsulam seus atributos e métodos, controlando o acesso aos dados.
- **Herança:** A classe ContaCorrente e ContaPoupanca herdam da classe abstrata Conta, implementando o método consultarSaldo(). A classe Usuario é herdada pelas classes Cliente, Gerente e Caixa.
- **Polimorfismo:** O método investir() é polimórfico, permitindo que diferentes tipos de investimentos sejam realizados, por exemplo, na renda fixa o cliente investe um valor personalizado, enquanto na renda variável é possível comprar ações de um valor pré-estipulado.
- **Abstração:** A classe Conta é abstrata, definindo um contrato para as classes de conta concretas, conta poupança e conta corrente.
- **Exceções:** Exceções personalizadas foram utilizadas para tratar erros de forma específica, como por exemplo saldo insuficiente e senha incorreta.
- **Testes:** Testes Unitário JUnit foram criados para verificar a funcionalidade do sistema.

Decisões para o Projeto:

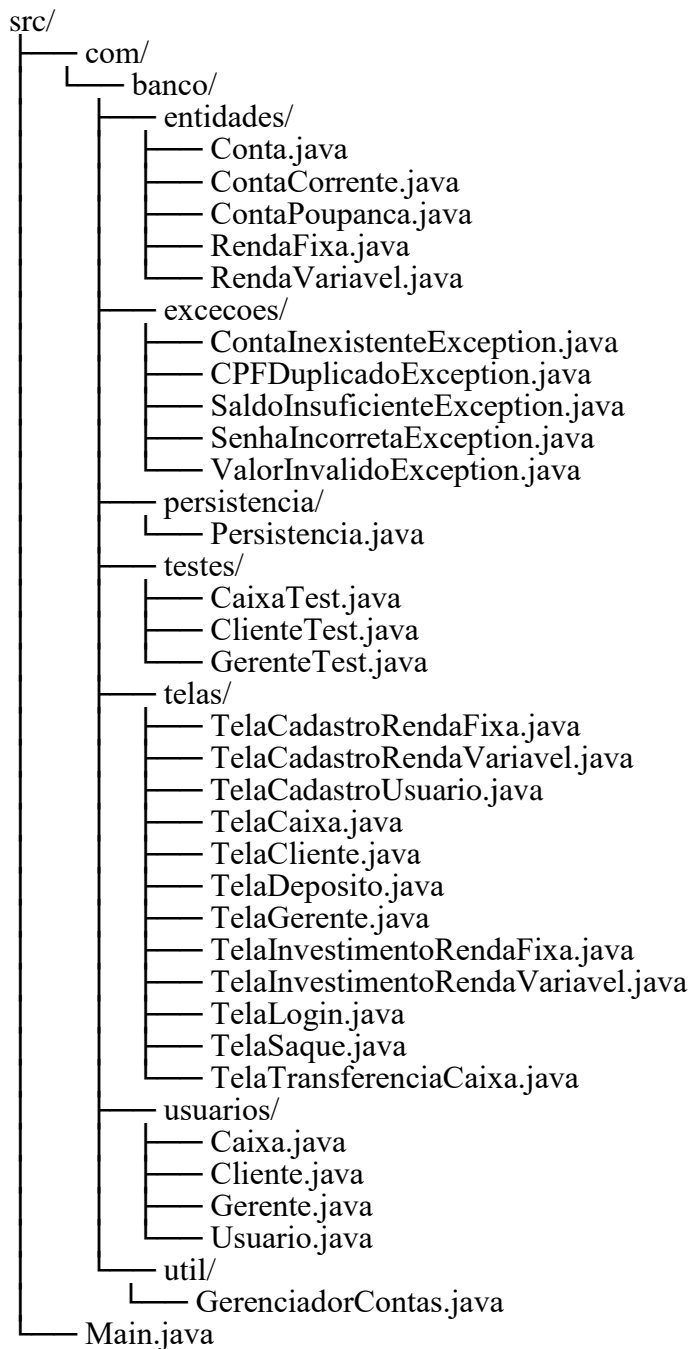
- **Persistência em Arquivos CSV:** A persistência em arquivos CSV foi escolhida pela facilidade de implementação utilizando classes padrão do Java, como FileWriter, FileReader, BufferedReader e PrintWriter. Quando o programa é

fechado salva todas as alterações feitas nos objetos nos arquivos CSV e quando é reaberto é buscado todas as informações pré-estabelecidas.

No caso de transações os arquivos são atualizados no momento da ação, caso concluída com êxito. Os arquivos usuarios.csv e contas.csv possuem as informações pertinentes a cada usuário e conta criado, responsáveis por manter a informação.

- **Interface Gráfica:** Foram desenvolvido menus simples e bem organizados para utilização grafica do usuário, separando cada tela com sua função.

Organização de Pacotes e Classes no diretório:



Execução:

Para execução do sistema é necessário ter instalado previamente o JDK 21 assim como Apache Maven 3.6.3 ou superior;

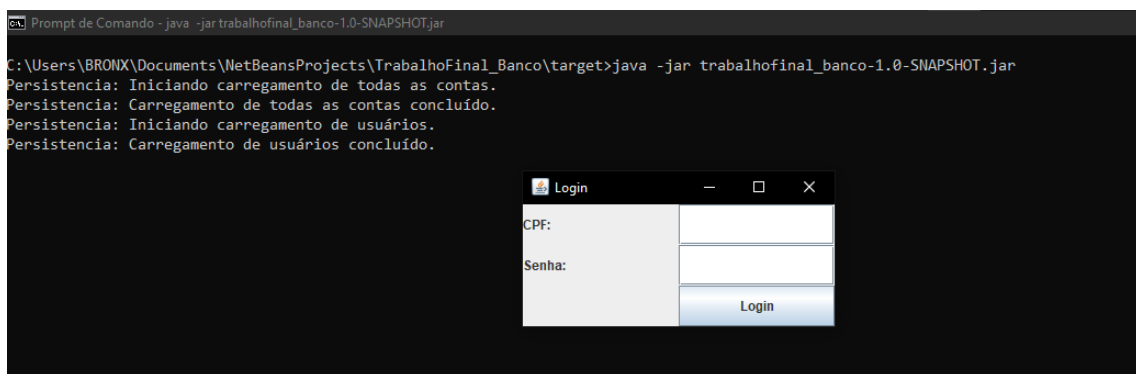
Será necessário acessar a pasta target do projeto como por exemplo:

```
cd C:\Users\usuario\...\NetBeansProjects\TrabalhoFinal_Banco\target\
```

Localizando o diretório atual do projeto, e rodando o comando

```
java -jar trabalhofinal_banco-1.0-SNAPSHOT.jar
```

Desta forma iniciando o programa.



Caso queira executar o projeto via IDE os arquivos de persistencia devem ser colocado junto a src do projeto.

Para testes pode ser verificado no arquivo usuarios.csv informações de login e senha.

Agradeço a paciência e atenção!

Muito Obrigado!

Jean Felipe Martinho