FACULDADE UNIRUY WYDEN

Paulo Brezolin
Felipe Sales
Gabriel Barros
Igor Cardoso
Jorsan Santos
Jonathan Magalhães

AVALIAÇÃO ENGENHARIA DE SOFTWARE

SISTEMA BANCÁRIO EM JAVA

Salvador, 07/06/2024

1 - Introdução

Este trabalho apresenta o desenvolvimento de um sistema bancário em Java localizado no repositório: https://github.com/PauloBrezolin/trabalho_engenharia_software, que permite a gestão de usuários e contas bancárias. O sistema oferece funcionalidades como cadastro de usuários, criação e login em contas bancárias, e realização de transações como depósitos, saques e transferências.

2 - Objetivos da Aplicação

O principal objetivo da aplicação é fornecer uma plataforma simples e segura para que usuários possam gerenciar suas contas bancárias, realizar transações financeiras e consultar saldos, simulando o funcionamento de um banco real.

3 - Funções/Lista de Eventos (Funcionalidades) – RF / RNF

Requisitos Funcionais (RF)

1. **Cadastro de Usuário (RF1):** Permitir o cadastro de novos usuários.
2. **Login de Usuário (RF2):** Validar o CPF do usuário para permitir acesso ao sistema.
3. **Cadastro de Conta (RF3):** Permitir a criação de contas bancárias (corrente ou poupança).
4. **Login de Conta (RF4):** Validar a existência de contas associadas ao usuário.
5. **Consulta de Saldo (RF5):** Permitir que o usuário consulte o saldo de sua conta.
6. **Depósito (RF6):** Permitir que o usuário deposite dinheiro em sua conta.
7. **Saque (RF7):** Permitir que o usuário saque dinheiro de sua conta.
8. **Transferência (RF8):** Permitir transferências de dinheiro entre contas.

Requisitos Não Funcionais (RNF)

1. **Segurança (RNF1):** Garantir que apenas usuários autenticados possam acessar suas contas.
2. **Usabilidade (RNF2):** Fornecer uma interface de usuário intuitiva e fácil de usar.
3. **Performance (RNF3):** Assegurar que o sistema responda rapidamente às ações do usuário.
4. **Confiabilidade (RNF4):** Garantir que os dados financeiros sejam armazenados e manipulados de forma precisa.

3.1 - Levantamento de Requisitos (CANVAS – Mapear Processos)

- **Clientes:** Usuários que desejam gerenciar suas contas bancárias.
- **Proposta de Valor:** Facilitar a gestão de contas bancárias e realizar transações financeiras com segurança.
- **Canais:** Aplicação Java para desktop.
- **Relacionamento com Clientes:** Interface intuitiva e suporte a funcionalidades essenciais de um banco.
- **Fontes de Receita:** Não aplicável (protótipo educativo).
- **Recursos Chave:** Interface de usuário, sistema de autenticação, transações financeiras.
- **Atividades Chave:** Cadastro de usuário, login, gestão de contas, transações.
- **Parcerias:** Não aplicável.
- **Estrutura de Custos:** Desenvolvimento e manutenção da aplicação.

3.1.1 - Entrevista / Questionário / Layout de Tela (Prototipação)

Entrevista/Questionário:

1. Quais são as funcionalidades mais importantes para você em um sistema bancário?
2. Você prefere uma interface mais simples ou com mais opções de customização?
3. Quais tipos de transações você realiza com mais frequência?
4. Qual a sua maior preocupação ao usar um sistema bancário online?

Layout de Tela (Protótipos):

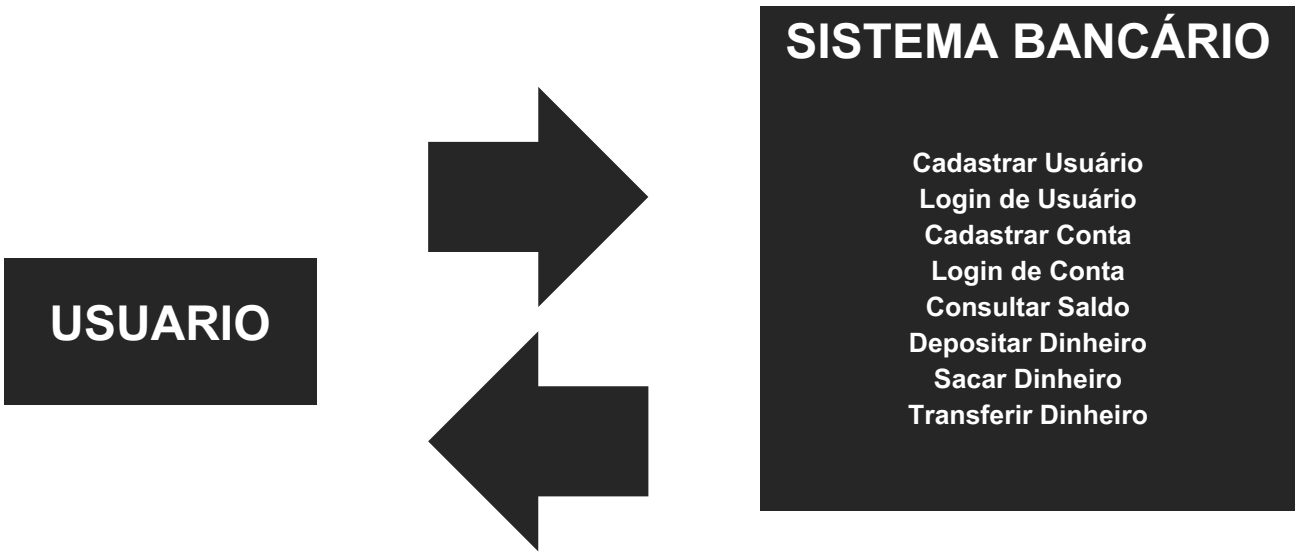
1. Tela de Login
2. Tela de Cadastro de Usuário
3. Tela de Cadastro de Conta
4. Tela Principal de Transações

3.1.2 - Casos de Uso; Diagrama de Casos de Uso - UML

Casos de Uso

1. **Cadastrar Usuário**
2. **Login de Usuário**
3. **Cadastrar Conta**
4. **Login de Conta**
5. **Consultar Saldo**
6. **Depositar Dinheiro**
7. **Sacar Dinheiro**
8. **Transferir Dinheiro**

Diagrama de Casos de Uso



3.2 - Especificação de Programas

Main.java
- Contém o fluxo principal da aplicação.
- Gerencia a interação com o usuário.
- Realiza chamadas aos métodos de cadastro, login e transações.

EntityFactory.java
- Interface para operações com o banco de dados.
- Métodos para cadastrar usuários, validar CPF, cadastrar contas, validar contas, atualizar saldo, etc.

Conta.java
- Classe conta.

Pessoa.java
- Classe pessoa.

Transferencia.java
- Classe transferencia.

3.2.1 Layout da Tela

1. **Tela de Login:** Campo para inserir CPF e botão para login.
2. **Tela de Cadastro de Usuário:** Campos para nome, telefone e CPF.
3. **Tela de Cadastro de Conta:** Opções para selecionar o tipo de conta e gerar número da conta.
4. **Tela Principal de Transações:** Opções para consultar saldo, depositar, sacar e transferir dinheiro.

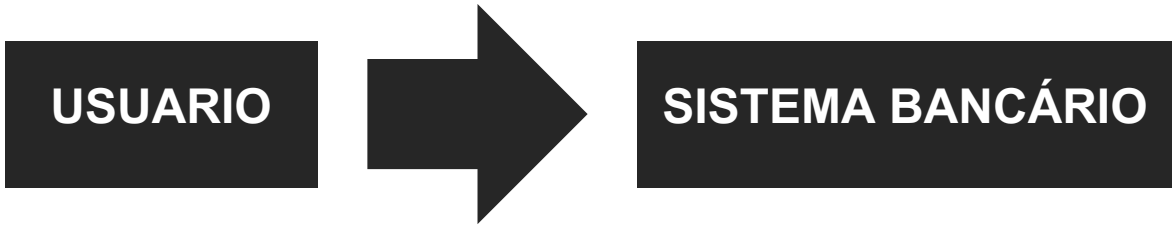
3.2.2 - Regras de Negócio

1. **RN1:** CPF deve conter exatamente 11 dígitos numéricos.
2. **RN2:** Nome deve conter apenas letras.
3. **RN3:** Telefone deve conter exatamente 11 dígitos numéricos.
4. **RN4:** Usuário só pode ter uma conta corrente e uma conta poupança.
5. **RN5:** Saque e transferência só podem ser realizados se houver saldo suficiente.

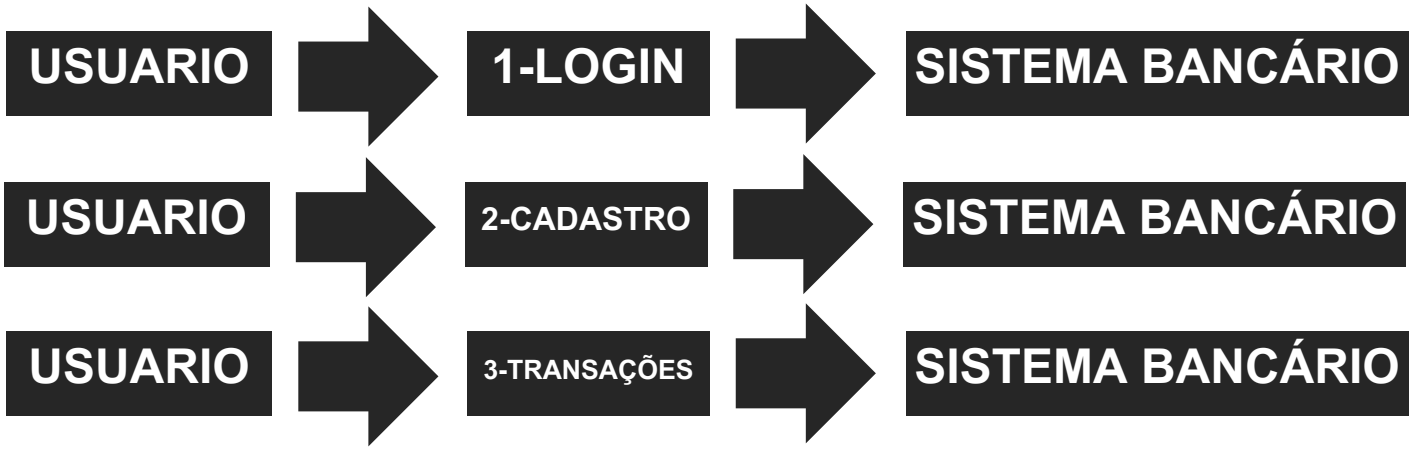
3.2.3 - Entidades Envolvidas (Classes) / Tabelas

1. **Pessoa:** Nome, CPF, Telefone.
2. **Conta:** Número, Dígito, Saldo, Tipo de Conta, Pessoa (associada).
3. **Transação:** Tipo, Valor, Data, Conta de Origem, Conta de Destino.

4 - Diagrama de Contexto

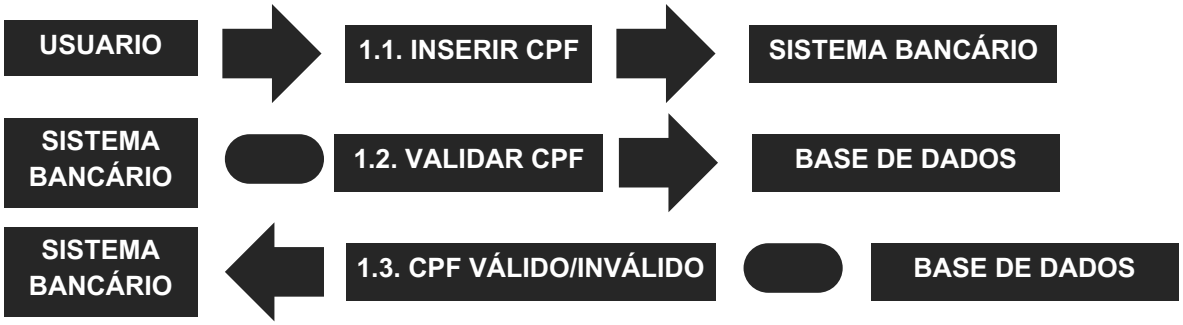


5 - DFD Nível Zero



6 - DFD por Evento

Evento: Login de Usuário



Evento: Cadastro de Usuário

7 - DER / Diagrama de Classe

DER

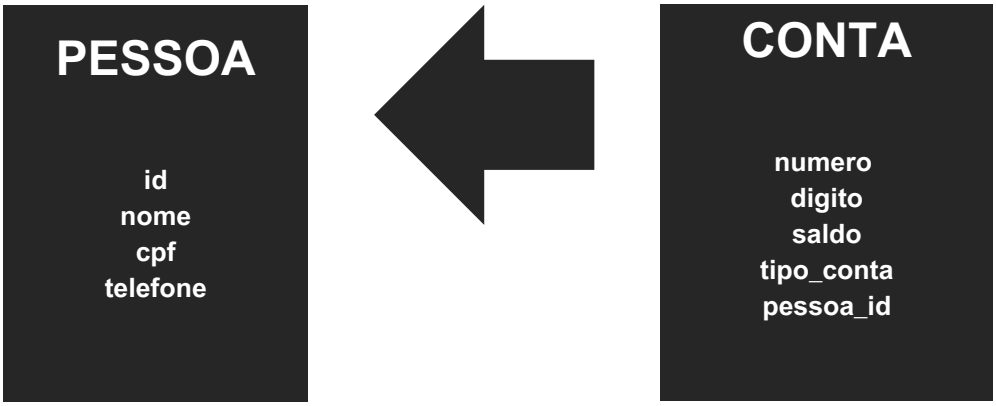
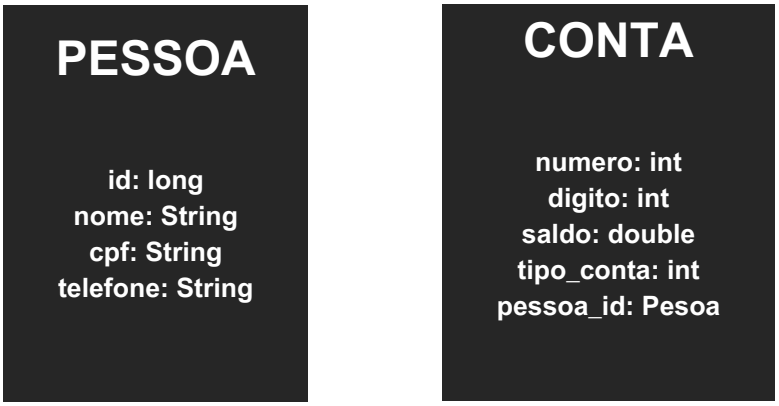


Diagrama de Classe



7.1 - Artefato gráfico – design do domínio

Representação gráfica das classes e suas relações como mostrado acima no diagrama de classe.

7.1 - Dicionário de Dados (DDL)

CREATE TABLE Pessoa (id BIGINT PRIMARY KEY, nome VARCHAR(100), cpf CHAR(11), telefone CHAR(11));

CREATE TABLE Conta (numero INT PRIMARY KEY, digito INT, saldo DOUBLE, tipo_conta INT, pessoa_id BIGINT, FOREIGN KEY (pessoa_id) REFERENCES Pessoa(id));

7.3 - Modelo Comportamental (Relação Entidade Pai x Filha)

Uma Pessoa pode ter várias Contas, mas cada Conta pertence a uma única Pessoa.

8 - Política de Testes (Qualidade do Produto) – Testes Internos

1. **Testes de Unidade:** Testar individualmente métodos de cadastro, login e transações.
2. **Testes de Integração:** Testar interação entre diferentes partes do sistema.
3. **Testes de Aceitação:** Garantir que o sistema atenda aos requisitos dos usuários finais.

9 - Descrever a Implantação

Homologação (Validar Aplicação)

1. **Teste em Ambiente Controlado:** Validar todas as funcionalidades antes de liberar para produção.
2. **Feedback dos Usuários:** Coletar feedback de um grupo seleto de usuários.

Instalação

1. **Requisitos de Sistema:** Java 8+, dependências de persistência (como JPA/Hibernate).
2. **Deployment:** Disponibilizar o executável ou arquivo jar para download.

Treinamento

1. **Documentação:** Fornecer documentação detalhada para uso do sistema.
2. **Tutoriais:** Criar tutoriais em vídeo para explicar as principais funcionalidades.

10 - Aplicação Protótipo (Linguagem livre escolha)

10.1 - Menu/Submenu

- **Menu Principal:** Opções de login, cadastro de usuário e cadastro de conta.
- **Submenu de Transações:** Opções para consultar saldo, depositar, sacar e transferir dinheiro.

10.2 - Telas Funcionais

- **Tela de Login:** Interface para entrada do CPF e validação.
- **Tela de Cadastro de Usuário:** Formulário para entrada de nome, telefone e CPF.
- **Tela de Cadastro de Conta:** Interface para seleção do tipo de conta e geração do número da conta.
- **Tela de Transações:** Menu de opções de transações financeiras.

10.3 - Telas de Diálogo

- **Diálogo de Confirmação:** Exibir após cada transação confirmando sucesso ou falha.
- **Diálogo de Erro:** Exibir mensagens de erro como CPF inválido ou saldo insuficiente.

10.4 - Layout Relatórios

- **Relatório de Transações:** Listar todas as transações realizadas por um usuário com detalhes como data, tipo, valor e contas envolvidas.