UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
CENTRO MULTIDISCIPLINAR
CURSO DE TECNOLOGIA DA INFORMAÇÃO

ALAN ALMEIDA DA SILVA
BRUNO VICTOR PAIVA DA SILVA
TIAGO AMARO NUNES

PROJETO DE TESTE: SISTEMA BANCÁRIO

Pau dos Ferros - RN
2023

PROJETO DE TESTE: SISTEMA BANCÁRIO

Projeto de um sistema bancário, para pôr em prática o conteúdo da disciplina teste de software.

Professor (a): Alysson Filgueira Milanez.

Discentes: Alan Almeida Da Silva;
Bruno Victor Paiva Da Silva;
Tiago Amaro Nunes.

Pau dos Ferros - RN
2023

INTRODUÇÃO

Bem-vindo ao projeto do Sistema Bancário em Java! Neste projeto, desenvolvemos um sistema que simula as operações básicas de um banco, permitindo a criação de contas, depósitos, saques e transferências entre contas.
Nosso objetivo é aplicar conceitos de Teste de Software para criar uma aplicação eficiente, segura e fácil de usar.

Ao longo do projeto, aprenderemos a modelar classes que representam clientes e contas bancárias, utilizar herança para diferentes tipos de contas (corrente e poupança), e programar métodos para realizar as operações bancárias.
Também iremos testar a aplicação para garantir que todas as funcionalidades estejam funcionando corretamente e criar um banco de dados para armazenar as informações relevantes.

DESENVOLVIMENTO

O Sistema Bancário proporciona uma experiência financeira abrangente e conveniente para nossos clientes. Com suas diversas funcionalidades, os usuários terão à disposição um conjunto completo de ferramentas para gerenciar suas contas e realizar suas operações bancárias de forma segura e eficiente.

Requisitos Funcionais:

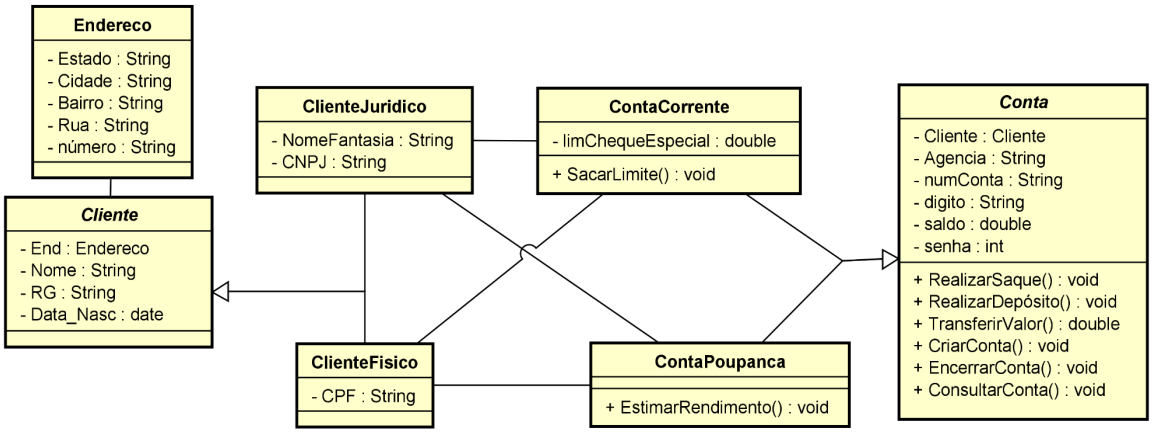
Requisito Funcional (RF)	Descrição
RF001 - Criar conta bancária	Clientes podem criar conta (corrente ou poupança), para pessoa física ou jurídica, com dados pessoais.
RF002 - Acessar conta	Login com credenciais para acessar conta bancária.
RF003 - Visualizar informações	Ver saldo, número da conta e histórico.
RF004 - Realizar depósito	Adicionar fundos à conta.
RF005 - Realizar saque	Retirar dinheiro ou transferir para outras contas.
RF006 - Transferir entre contas	Entre contas próprias ou do mesmo banco.
RF007 - Exibir extrato	Ver transações recentes.
RF008 - Ver histórico	Visualizar histórico completo de transações.

RF009 - Cheque especial	Estabelecer um limite e ver informações do cheque especial.
RF010 - Encerrar conta	Solicitar encerramento e transferência de saldo.

Requisitos não funcionais:

Requisito Não Funcional (RNF)	Descrição
RNF001 - Validar Idade	O usuário só pode criar uma conta bancária se possuir idade igual ou superior a 18 anos.
RNF002 - Controle de acesso	garantir que apenas pessoas autorizadas possam visualizar ou modificar informações financeiras.
RNF003 - Validar transação	Usuário só realizará uma transação após confirmar sua senha.
RNF004 - Limite de saque	Cada usuário tem direito a 2 saques por acesso.
RNF005 - Restrição cliente jurídico	Uma pessoa jurídica não deve conseguir abrir conta poupança.

ARTEFATO



O diagrama ilustra a relação entre as entidades do sistema bancário, como Cliente e Conta, e as subclasses especializadas, ClienteJurídico, ClienteFísico, ContaCorrente e ContaPoupanca. A entidade Endereço está relacionada ao cadastro de um cliente no sistema, ou seja, todo cliente deve ter um endereço

Essa modelagem fornece uma visão geral do sistema bancário, onde clientes podem possuir múltiplas contas, incluindo contas correntes e poupanças, e realizar transações financeiras entre essas contas. A relação de herança entre Conta, ContaCorrente e ContaPoupanca permite que a aplicação diferencie e gerencie os diferentes tipos de contas de forma adequada.

TIPOS DE TESTES A SEREM UTILIZADOS:

- Fases de Teste (testa o sistema)

Teste de unidade: implementação do código fonte, foco individual na unidade do sistema. *Objetivo:* resolver problemas limitados a unidade, não detecta todos os erros;

Teste de integração: execução entre as partes, verifica se o resultado foi aquele esperado. *Objetivo:* encontrar falhas na integração interna;

Teste de validação: procura atender aos requisitos da modelagem;

Teste de aceitação: simula operações de rotina do sistema, verifica se o comportamento ocorreu como o esperado;

Teste de sistema: o software e os outros elementos são testados em conjunto. *Objetivo:* executar o sistema sob ponto de vista do usuário final, verificando as funcionalidades e requisitos.

- Criação do oráculo

Determina se o teste passa ou não;

É acompanhado de uma especificação com base no resultado;

É preciso estabelecer precondições necessárias para a criação do oráculo, pois não existe modelo pronto (comparativo).

- Consistência

Deve haver um bom relacionamento com o que deve ser entregue;

Teste de defeito e teste de validação.

- Testes Caixa Branca

Usa o código para derivar casos de teste;

Sistemas grandes requerem muita dificuldade para realizar teste de caixa branca, mas acabam sendo realizados;

A implementação do teste de caixa branca sem as especificações de requisitos do oráculo acaba se tornando impossível de serem detectadas.

LInks úteis:

Repositório GitHub -> https://github.com/brunopaiva1/Sistema_bancario