Relatório Técnico

Sistema Bancário em Java

1. Introdução

O presente relatório documenta o desenvolvimento de um sistema bancário em Java, implementado com os princípios de Programação Orientada a Objetos (POO) e utilizando persistência de dados em arquivos. O sistema foi projetado para simular operações típicas de um banco, com múltiplos tipos de contas e usuários com diferentes níveis de permissões. As funcionalidades incluem saques, depósitos, transferências e autenticação por usuário e senha, bem como configurações específicas para contas adicionais e rendimento em contas poupança.

2. Metodologia

A implementação foi organizada em diversas classes, cada uma representando componentes do sistema. Abaixo está a descrição das principais classes e métodos:

2.1 Classes

- **Usuário (Abstrata)**: Classe base para os tipos Gerente, Bancário e Correntista. Contém atributos como nome, usuario, senha e métodos de autenticação.
- **Gerente:** Extende Usuário e possui permissões para cadastrar novos correntistas e configurar limites para contas adicionais.
- **Bancário:** Extende Usuário e permite realizar operações financeiras nas contas dos correntistas, exceto cadastrá-los.
- **Correntista:** Representa o cliente do banco, com acesso às operações em suas contas.
- **Conta (Abstrata):** Classe base para ContaCorrentePrincipal, ContaCorrenteAdicional e ContaPoupança. Contém métodos genéricos como depositar, sacar e atributos como saldo e numeroConta.

- **ContaCorrentePrincipal:** Permite todas as operações bancárias e pode ter cheque especial.
- **ContaCorrenteAdicional:** Vinculada a uma conta principal, com limite pré-definido de gastos.
- **ContaPoupança:** Oferece rendimento sobre o saldo com base em uma taxa de juros fixa.

2.2 Estrutura de Dados

- **Persistência:** Dados dos usuários e contas são armazenados em arquivos.txt, com leitura e escrita realizadas por meio de classes específicas para manipulação de arquivos. **- Controle de Acesso:** A autenticação utiliza um sistema de verificação por nome de usuário e senha.

2.3 Fluxo de Operações

- **Cadastro:** Apenas gerentes podem cadastrar novos correntistas e configurar limites das contas adicionais.
- **Operações Bancárias:** Saques, depósitos e transferências são validados quanto a saldo disponível e limites definidos.
- **Simulação de Rendimento:** A conta poupança calcula juros com base em um método que atualiza o saldo periodicamente.

3. Desafios Encontrados

- **Persistência em Arquivos:** A implementação de leitura e escrita eficiente exigiu a criação de métodos robustos para evitar inconsistências nos dados.
- **Solução:** Uso de formatação estruturada para os arquivos e validações na leitura.
- **Autenticação e Controle de Acesso:** Garantir que permissões fossem respeitadas de acordo com o tipo de usuário.
- **Solução:** Definição de métodos e classes específicos para lidar com níveis de permissão.

- **Cálculo de Rendimento:** Implementação de lógica que atualiza o saldo das contas poupança periodicamente.
- **Solução:** Criação de métodos que simulam rendimento com base em intervalos configuráveis.

4. Demonstração

Abaixo estão exemplos práticos das principais funcionalidades do sistema.

Exemplo 1: Cadastro de Usuário – Gerente
cadastra um novo correntista:
Nome: Caio Campos
Usuário: caio_campos
Conta Corrente Principal criada com saldo inicial de R$ 1.000,00.

Exemplo 2: Operação de Depósito - Bancário realiza depósito:
Conta: 123456
Valor: R$ 500,00
Saldo atualizado: R$ 1.500,00

Exemplo 3: Transferência
Correntista transfere valores entre contas:
Origem: Conta Corrente Principal
Destino: Conta Corrente Adicional
Valor: R$ 200,00
Saldo restante: R$ 800,00

Exemplo 4: Rendimento em Conta Poupança
Após 1 mês, o saldo de uma conta poupança é atualizado:
Saldo inicial: R$ 2.000,00
Rendimento: 0,5% ao mês

Saldo final: R$ 2.010,00

5. Conclusão

O sistema bancário implementado atende aos requisitos propostos, com uma arquitetura modular que facilita futuras manutenções e expansões. A aplicação demonstra o uso prático de conceitos de POO e manipulação de arquivos em Java, fornecendo uma solução funcional e segura para as operações bancárias simuladas.