**Nomes: Guilherme Reis Carvalho, Nicolas Lobo, Vinicius Trevisan, Luiz Assi, Guilherme Marafon**

**Seção 1 – Descrição do projeto:**

**Nome: Banking System**

**1.1. Contexto e Propósito**

O software objeto deste estudo é um sistema de simulação de operações bancárias, desenvolvido em Java com a biblioteca Swing para a interface gráfica. O seu propósito principal é fornecer uma aplicação funcional para o gerenciamento de contas bancárias, permitindo ao usuário realizar transações básicas de forma segura e intuitiva.

A aplicação garante a persistência dos dados entre as sessões, salvando todas as informações das contas em um arquivo binário local (data.bin). A interação do usuário se inicia em uma tela de login, que leva a um menu principal onde todas as funcionalidades podem ser acessadas.

O código-fonte está estruturado em pacotes que dividem as responsabilidades do sistema: GUI para a interface, Bank para a lógica de negócio, Data para a persistência de arquivos e Exceptions para o tratamento de erros específicos. O ponto de entrada da aplicação é o método main na classe Application.java.

**1.2. Principais Funcionalidades**

O sistema oferece um conjunto de funcionalidades essenciais para um ambiente bancário simulado:

* **Gerenciamento de Contas:** Permite a criação e gestão de diferentes tipos de contas bancárias, cada uma com suas regras e atributos. Os tipos suportados são:
  + Conta Corrente
  + Conta Poupança
  + Conta Estudantil
* **Transações Financeiras:** O usuário pode executar operações de depósito e saque em contas existentes através de telas específicas para cada função.
* **Consulta de Informações:** Disponibiliza uma funcionalidade para exibir uma lista completa de todas as contas cadastradas no sistema.
* **Regras de Negócio e Validações:** A aplicação implementa um sistema robusto de validações para garantir a integridade dos dados e das operações, incluindo:
  + Verificação de valores inválidos para depósitos e saques.
  + Aplicação de limites de saldo máximo e de retirada.
  + Tratamento de erros para contas não encontradas.

**1.3. Métricas de código**

|  |  |  |
| --- | --- | --- |
| Métrica | Valor | Ferramenta |
| LOC | 138 files, 3872 codes, 172 comments, 317 blanks, all 4361 lines | VSCodeCounter |
| Complexidade ciclomática total | 99 | Analyze (IntelliJ) |
| Número de classes | 21 | Analyze (IntelliJ) |
| Número de métodos | 39 | Analyze (IntelliJ) |

**1.4. Vídeo**

https://youtu.be/w4AiQV118yY