

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CAMPUS CORNÉLIO PROCÓPIO
Bacharelado em Engenharia Da Computação

Paulo Cesar De Oliveira Mitsi
Guilherme Munhos Matos

MAXIMUS BANK

CORNÉLIO PROCÓPIO – PR
2023

**Paulo Cesar de Oliveira Mitsi
Guilherme Munhos Matos**

MAXIMUS BANK

Projeto elaborado na disciplina de Programação Orientada a Objetos do curso de Engenharia de Software, do Campus Cornélio Procópio da Universidade Tecnológica Federal do Paraná.

Professora: Profa. Gisele Alves Santana

**CORNÉLIO PROCÓPIO
2023**

RESUMO

O projeto desenvolvido é um sistema bancário em Java usando o NetBeans e PostgreSQL como banco de dados. Seus objetivos incluí criar uma interface intuitiva para transações, saques, depósitos e empréstimos. Adotou-se uma abordagem modular para facilitar a manutenção e expansão. Além disso, os resultados alcançados incluem a capacidade de cadastrar, editar e excluir clientes, associando contas correntes, poupanças e seguro de vida. O sistema simula operações bancárias com consistência e segurança, protegendo dados sensíveis. Logo, o projeto atingiu com sucesso seus objetivos, oferecendo um sistema bancário completo, seguro e adaptável às demandas do setor.

Sistema bancário, Java, NetBeans, PostgreSQL;

SUMÁRIO

1	INTRODUÇÃO.....	5
2	FERRAMENTAS E TECNOLOGIAS	5
3	DESENVOLVIMENTO	6
3.1	Levantamento de Requisitos	7
3.2	Diagramas de Casos de Uso.....	8
3.3	Especificações de Casos de Uso	10
3.4	Diagrama de Classe	11
3.5	Diagrama de Objetos.....	12
3.6	Banco de dados	13
3.7	Diagrama de Sequência	14
3.8	Telas do aplicativo.....	15
4	CONCLUSÃO	16

1 INTRODUÇÃO

O projeto em questão visa desenvolver um sistema bancário eficiente e abrangente, utilizando a linguagem de programação Java e a plataforma de desenvolvimento NetBeans. A metodologia adotada consistiu na integração do PostgreSQL como banco de dados, proporcionando uma estrutura robusta para armazenar e gerenciar informações cruciais. O escopo do sistema inclui funcionalidades como transações, saques, depósitos e empréstimos.

Os principais resultados obtidos incluem a capacidade de cadastrar, editar e excluir clientes, associando diferentes tipos de contas a eles, como conta corrente, conta poupança e seguro de vida. O sistema simula com sucesso as operações bancárias, garantindo consistência e confiabilidade nos dados.

No âmbito dos objetivos, destaca-se a criação de uma interface intuitiva para usuários realizar operações bancárias de maneira simplificada. A metodologia de implementação focou na modularidade do código, facilitando a manutenção e expansão do sistema. Durante o desenvolvimento, foram adotadas práticas de segurança para proteger dados sensíveis dos clientes.

2 FERRAMENTAS E TECNOLOGIAS

Foram utilizados a linguagem Java, IDE do Netbeans e PostgreSQL.

3 DESENVOLVIMENTO

O sistema desenvolvido incorpora requisitos adicionais para aprimorar a segurança, usabilidade e desempenho. A segurança ao acesso de classes e funções é uma prioridade, garantindo que apenas o sistema tenha funcionalidades específicas, fortalecendo a confidencialidade e a integridade dos dados.

A priori, a interface do usuário foi a primeira a ser projetada com o objetivo de ser relativamente fácil de compreender, utilizar e para auxiliar na implementação das funcionalidades do sistema. Logo, essa prática visa proporcionar uma experiência intuitiva aos usuários, facilitando a navegação pelo sistema e a execução das operações bancárias.

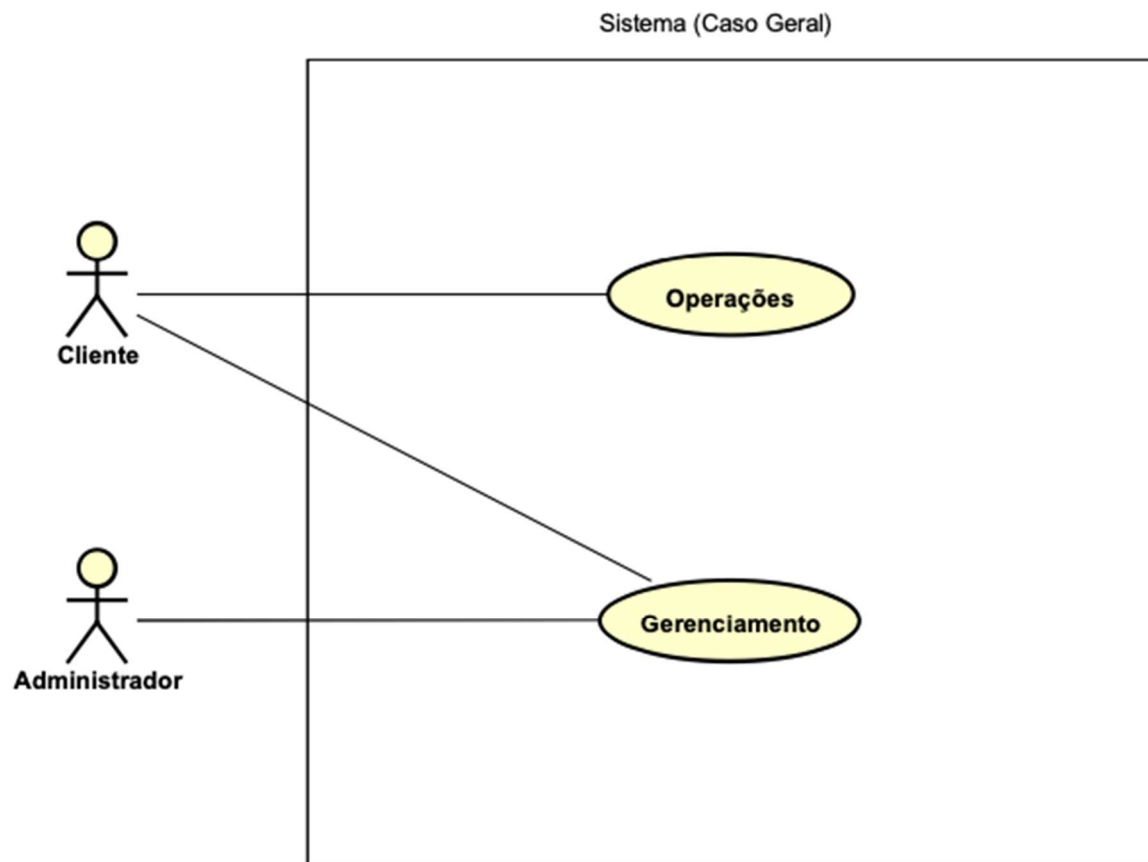
Quanto ao desempenho, o sistema foi desenvolvido visando um alto tempo de resposta, assegurando eficiência e agilidade nas operações realizadas pelos usuários. A alta disponibilidade é uma característica incorporada ao sistema, garantindo que este esteja sempre operacional quando estiver sendo executado.

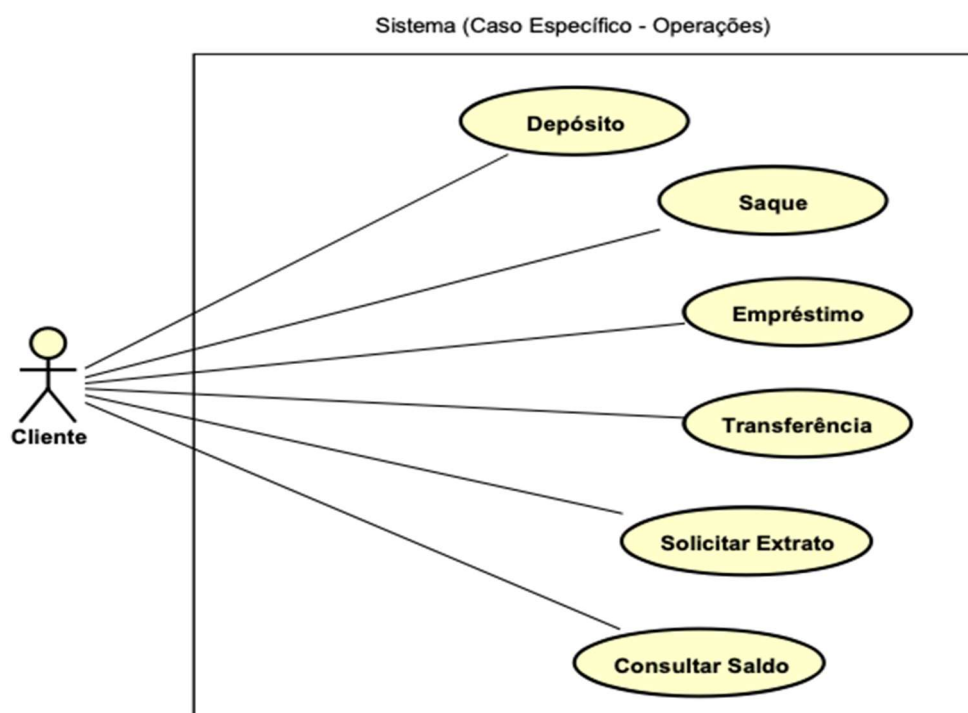
O sistema é totalmente operacional em Windows 7, 10 e 11. A comunicação eficiente com o banco de dados é um requisito essencial do sistema, garantindo que as informações sejam armazenadas e recuperadas de maneira eficaz e segura.

3.1 Levantamento de Requisitos

<i>Requisito</i>	Descrição	Prioridade
RF001	O sistema deve realizar depósito de dinheiro em contas	Alta
RF002	O sistema deve realizar transferência de dinheiro entre contas	Alta
RF003	O sistema deve realizar saques de dinheiro das contas cadastradas	Alta
RF004	O sistema pode realizar empréstimo para contas	Alta
RF005	O sistema deve realizar cadastro, exclusão, edição e consulta de contas em um database	Alta
RNF01	Segurança ao acesso de classes e funções	Alta
RNF02	Interface fácil de compreender e utilizar	Média
RNF03	O sistema deve contar com baixo tempo de resposta	Alta
RNF04	Operacional em Windows 7/10/11	Baixa
RNF05	Disponibilidade sempre que estiver sendo executado	Alta
RNF06	O sistema deve se comunicar com um banco de dados	Alta
RNF07	O sistema deverá ser codificado de forma a ficar simples a manutenção do código	Média

3.2 Diagramas de Casos de Uso



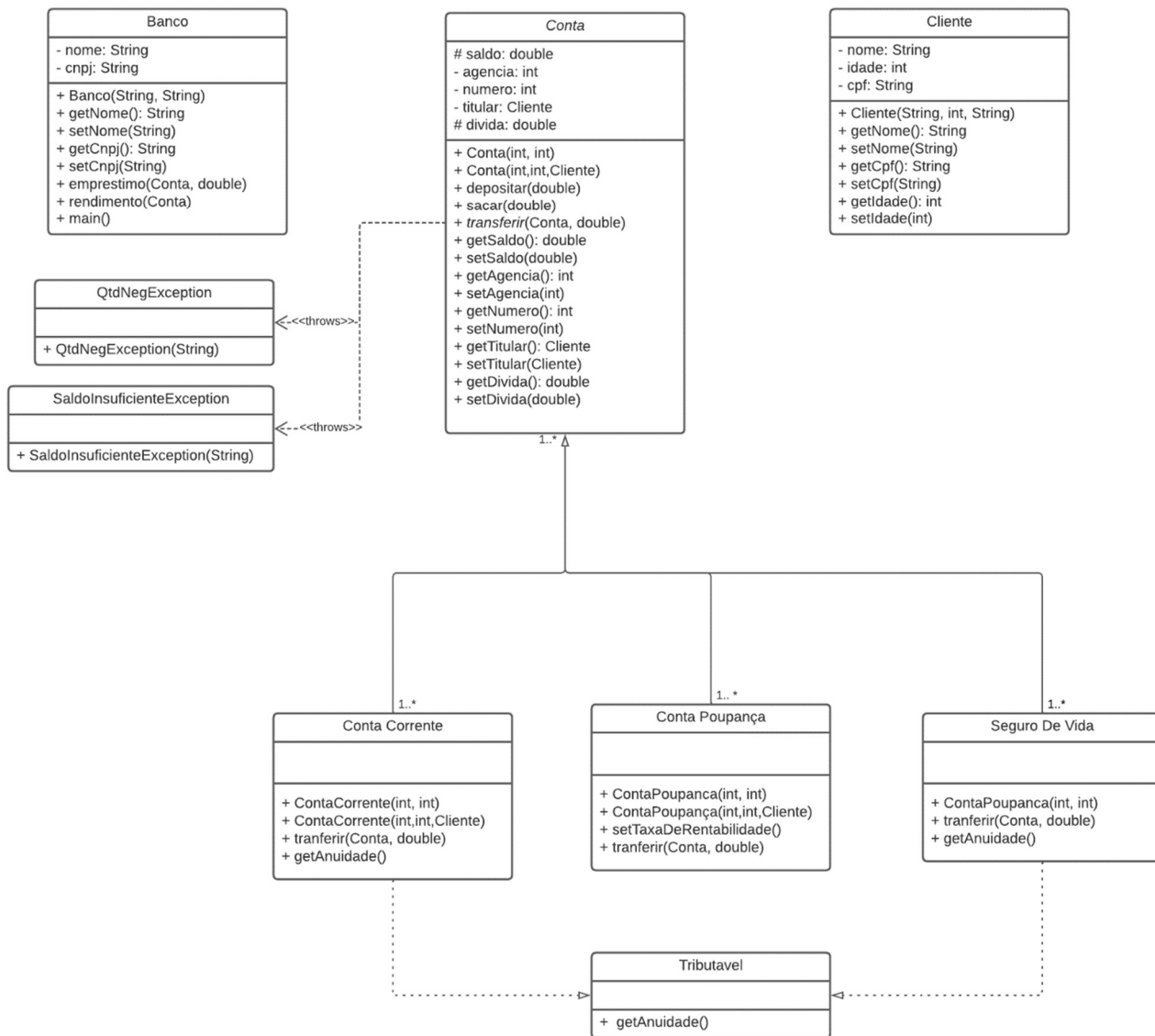


3.3 Especificações de Casos de Uso

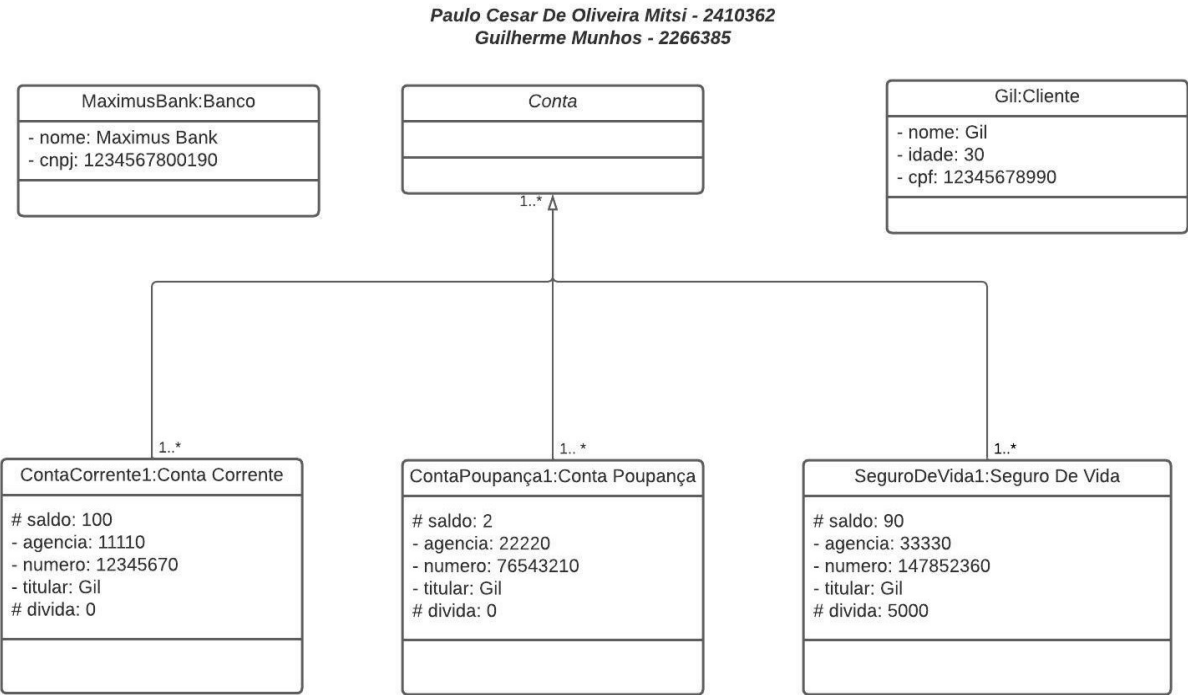
Nome do Caso de Uso	Consultar dados do cliente
Ator Principal	Administrador
Atores secundários	Próprio aplicativo
Resumo	Este caso de uso descreve as etapas percorridas por um administrador para consultar os dados de um cliente.
Pré-condições	Nenhuma.
Pós-condições	O aplicativo buscará no banco de dados e apresentará em uma tabela (se existirem) os dados do determinado cliente.
Ações do Ator	Ações do Sistema
1. Realizar a consulta por Nome ou CPF.	
2. Confirmar os dados inseridos.	
	3. Verificar se existem contas com os dados fornecidos.
	4. Caso existam, apresentá-las em uma tabela. Caso não existam, informar que nenhum cliente foi encontrado com os dados fornecidos.
	5. Perguntar se o ator deseja realizar mais operações.
6. Responder ao questionamento	
	7. Caso deseje realizar mais operações, repita o processo com o campo de busca limpo, caso contrário, encerre o procedimento.

3.4 Diagrama de Classe

Paulo Cesar De Oliveira Mitsi - 2410362
Guilherme Munhos - 2266385



3.5 Diagrama de Objetos



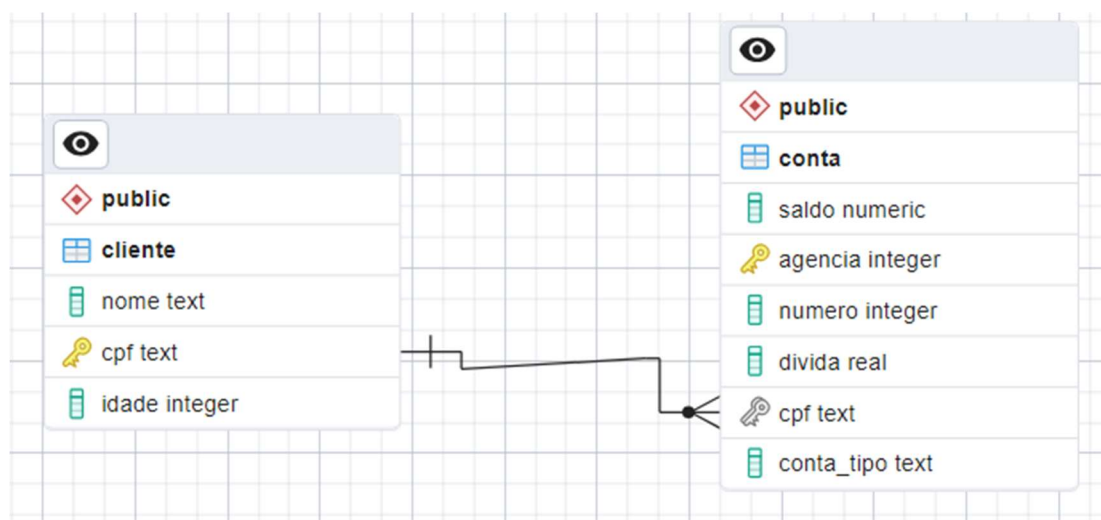
3.6 Banco de dados

Tabela cliente

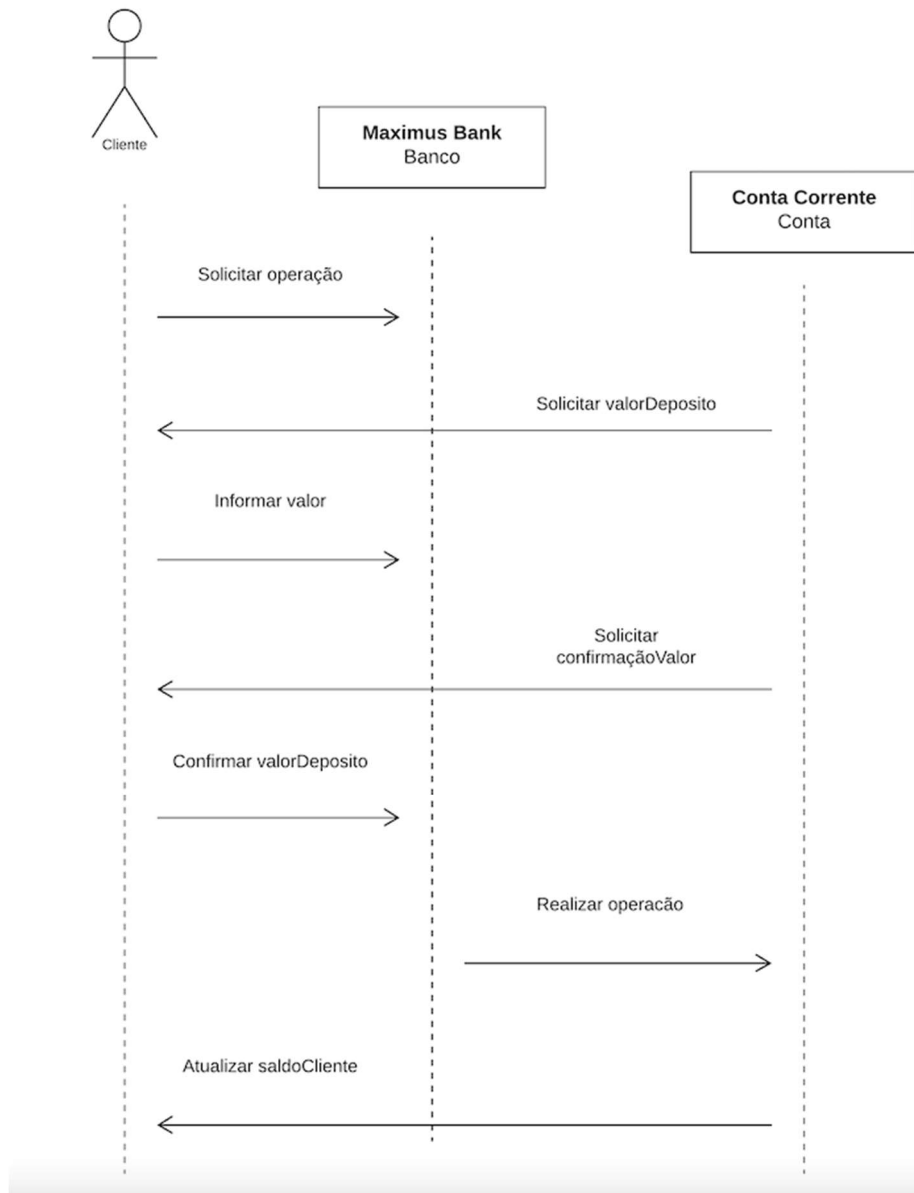
	nome text	cpf [PK] text	idade integer
1	Scalco	13107381901	48
2	Alex	29297947498	31
3	Guilherme	54125841269	20

Tabela Conta

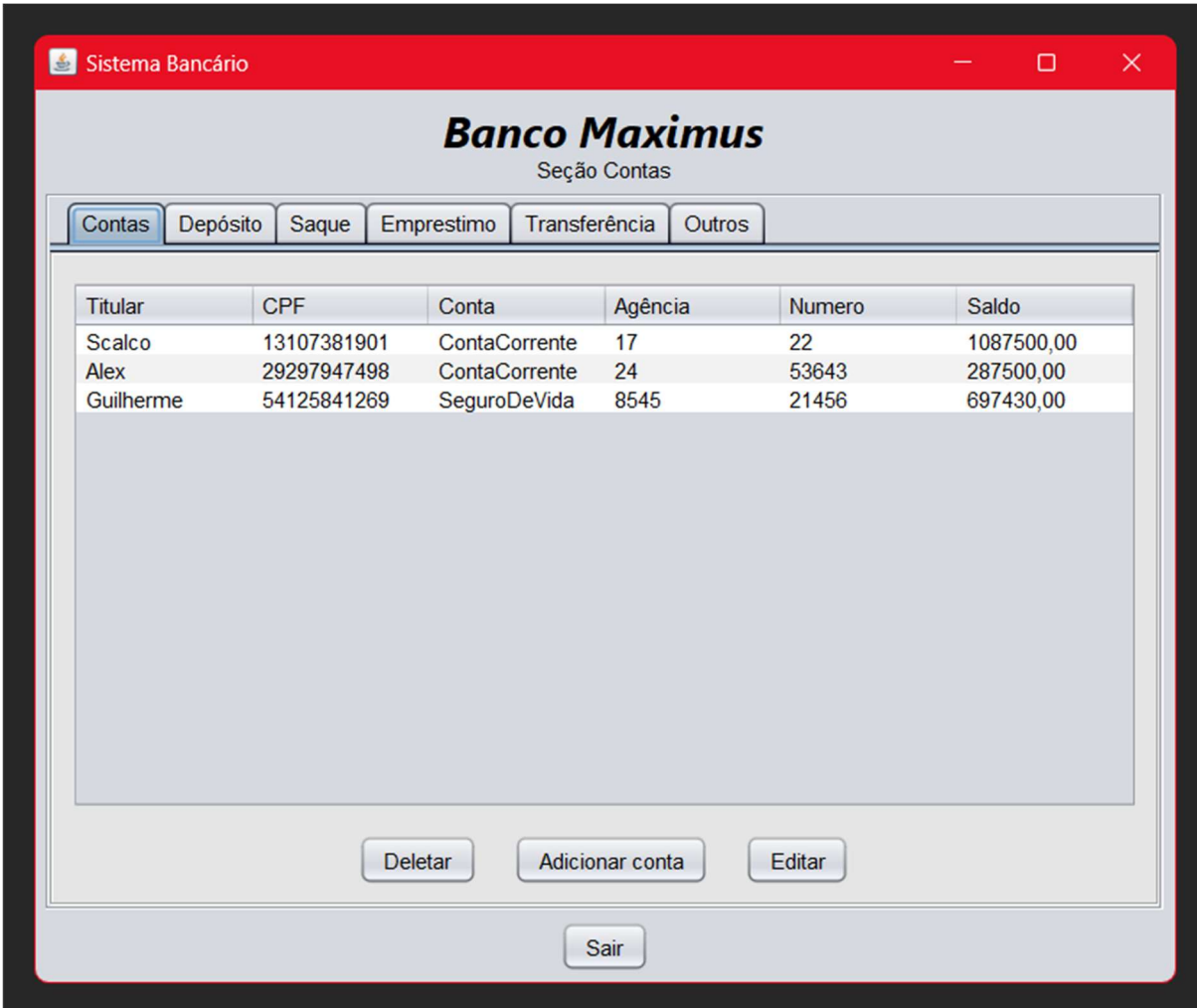
	saldo numeric	agencia [PK] integer	numero integer	divida real	cpf text	conta_tipo text
1	1087500	17	22	200000	13107381901	ContaCorrente
2	287500	24	53643	0	29297947498	ContaCorrente
3	697430.0	8545	21456	0	54125841269	SeguroDeVida



3.7 Diagrama de Sequência



3.8 Telas do aplicativo



Sistema Bancário

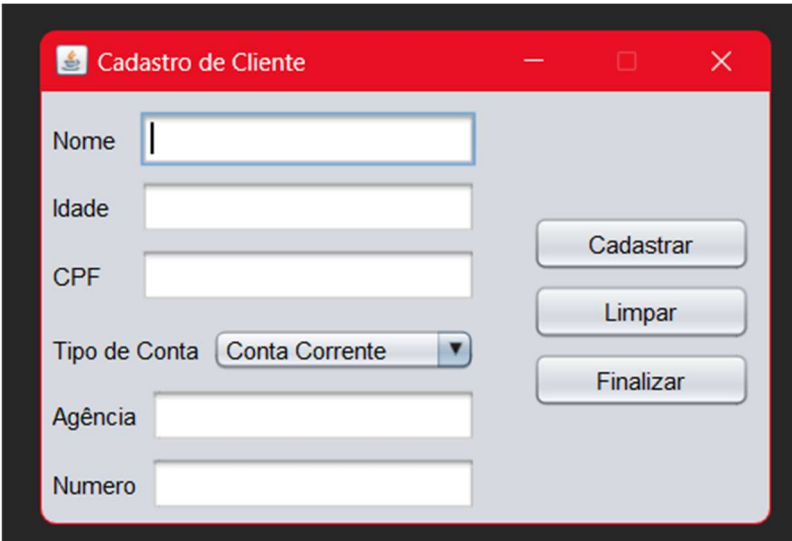
Banco Maximus
Seção Contas

Contas Depósito Saque Emprestimo Transferência Outros

Titular	CPF	Conta	Agência	Numero	Saldo
Scalco	13107381901	ContaCorrente	17	22	1087500,00
Alex	29297947498	ContaCorrente	24	53643	287500,00
Guilherme	54125841269	SeguroDeVida	8545	21456	697430,00

Deletar Adicionar conta Editar

Sair



Cadastro de Cliente

Nome

Idade

CPF

Tipo de Conta

Agência

Numero

Cadastrar

Limpar

Finalizar

4 CONCLUSÃO

Os resultados alcançados no projeto revelaram sucesso na implementação de um sistema bancário funcional. As principais conquistas incluem a capacidade de realizar transações, saques, depósitos e empréstimos, além da gestão eficiente de clientes e diferentes tipos de contas. A interface simples proporciona uma experiência amigável, enquanto a segurança dos dados sensíveis é mantida. O projeto atingiu seus objetivos, oferecendo um sistema seguro e adaptável.

Durante o desenvolvimento, foi identificado um desafio significativo relacionado à criação de contas diferentes em agências com números iguais. Este problema estava vinculado à utilização de uma chave primária (PK) no banco de dados, gerando conflitos ao tentar criar registros distintos com a mesma agência. A resolução desse problema demandou uma revisão cuidadosa do modelo de dados e ajustes na estrutura do banco para evitar violações de integridade, na qual foi apenas permitido a criação de contas com apenas um tipo de agência.

Algumas melhorias foram pensadas para deixar o projeto mais completo, porém não foram implementadas, pois iriam deixar muito complexo e dificultoso na implementação, além de demandar muito tempo que excederia a data de entrega. Logo as sugestões de Melhorias são:

- Validação de Dados: Implementar verificações adicionais para evitar conflitos de chave primária e assegurar a integridade dos dados, prevenindo problemas como o mencionado com agências de números iguais.
- Aprimoramento da Interface: Considerar melhorias na interface do usuário para tornar as operações ainda mais intuitivas, contribuindo para uma experiência de usuário mais eficiente.
- Aumento da Complexidade: Expandir as funcionalidades do sistema para incluir operações mais avançadas, como investimentos e a implementação de crédito, aumentando a complexidade do projeto e proporcionando desafios adicionais para o aprendizado.

O projeto trouxe uma valiosa contribuição para o aprendizado ao confrontar desafios práticos, como problemas de integridade no banco de dados e a sua própria implementação. A resolução desses problemas demandou uma compreensão profunda dos conceitos teóricos aplicados na prática, promovendo uma experiência de aprendizado mais robusta. Além disso, o desenvolvimento modular e a adaptação a problemas inesperados proporcionaram insights valiosos sobre a complexidade do desenvolvimento de software em ambientes do mundo real. Essa experiência prática é fundamental para o aprimoramento das habilidades técnicas e analíticas dos envolvidos no projeto.