

UNIVERSIDADE FEDERAL DO PARANÁ
HENRIQUE ULBRICH DE SOUZA

Sistema bancário

Trabalho semestral apresentado como requisito para
obtenção de nota na disciplina “*Linguagem e*
Programação”, ministrada pelo Prof. Drº Roberson
César Alves de Araujo

CURITIBA
2024

ÍNDICE DETALHADO

1. INTRODUÇÃO AO DOCUMENTO.....	2
1.1. TEMA.....	2
1.2. OBJETIVO DO PROJETO.....	2
1.3. DELIMITAÇÃO DO PROBLEMA.....	2
1.4. JUSTIFICATIVA DA ESCOLHA DO TEMA.....	2
1.5. MÉTODO DE TRABALHO.....	2
1.6. GLOSSÁRIO.....	3
2. DESCRIÇÃO GERAL DO SISTEMA.....	4
2.1. DESCRIÇÃO DO PROBLEMA.....	4
2.2. PRINCIPAIS ENVOLVIDOS E SUAS CARACTERÍSTICAS.....	4
2.3. REGRAS DE NEGÓCIO.....	4
3. REQUISITOS DO SISTEMA.....	6
3.1. REQUISITOS FUNCIONAIS.....	6
3.2. REQUISITOS NÃO-FUNCIONAIS.....	7
4. ANÁLISE E <i>DESIGN</i>.....	8
4.1. ARQUITETURA DO SISTEMA.....	8
4.2. MODELO DO DOMÍNIO.....	8
4.3. AMBIENTE DE DESENVOLVIMENTO.....	9
5. TESTES.....	10
5.1. PLANO DE TESTES.....	10
5.2. EXECUÇÃO DO PLANO DE TESTES.....	12
6. IMPLANTAÇÃO.....	13
6.1. MANUAL DE IMPLANTAÇÃO.....	13
7. MANUAL DO USUÁRIO.....	14
8. CONCLUSÕES E CONSIDERAÇÕES FINAIS.....	15
REFERÊNCIAS.....	16

1. Introdução ao projeto

Este documento descreve um sistema de gerenciamento de contas bancárias desenvolvido em C. O sistema permite que os usuários criem contas, façam login, consultem saldo, realizem depósitos, saques, transferências e gerencie empréstimos. Este sistema é projetado para funcionar em modo de linha de comando e armazena os dados em uma estrutura de lista encadeada na memória.

1.1. Tema

O projeto é um sistema de gerenciamento de contas bancárias utilizando a linguagem C.

1.2. Objetivo do Projeto

O projeto tem como objetivo central o desenvolvimento de um sistema que possibilite a criação, o gerenciamento e a operação de contas bancárias. Este sistema busca oferecer uma interface intuitiva e segura, permitindo que os usuários realizem operações bancárias essenciais, tais como depósitos, saques, transferências e a gestão de empréstimos.

1.3. Delimitação do Problema

O projeto aborda um projeto simples e eficiente de gerenciamento de contas bancárias, operando em um ambiente de linha de comandos, muitas das soluções atuais são complexas e dependem de interfaces gráficas e banco de dados, o que acaba sendo excessivo para um projeto pequeno, este projeto visa fornecer uma solução leve

1.4. Justificativa da Escolha do Tema

Sempre tive interesse em compreender a lógica de conversação de dados de um sistema bancário, neste projeto tive a oportunidade de entender a funcionalidade e lógica de um app de banco que utilizamos todos os dias.

1.5. Método de Trabalho

Análise e Definição dos Requisitos: Identificação das funcionalidades essenciais do sistema de contas bancárias.

Projeto do Sistema: Estruturação das funções e da lógica de operação do sistema.

Implementação: Desenvolvimento do código utilizando a linguagem C, com foco na clareza e eficiência.

Testes e Validação: Realização de testes unitários e de integração para garantir o correto funcionamento das funcionalidades implementadas.

1.6. Glossário

Conta Bancária: Estrutura que contém informações sobre um cliente, como nome, CPF, data de nascimento, número da conta, senha, saldo, empréstimo e um ponteiro para a próxima conta.

CPF: Cadastro de pessoas físicas, no formato extenso sem separação(ex:00000000000)

Data de Nascimento: Data em que o titular da conta nasceu, no formato **dd/mm/yyyy**.

Número da Conta: Identificador único atribuído a cada conta bancária no sistema.

Senha: Cadeia de caracteres utilizada para autenticar o usuário no sistema.

Saldo: Quantidade de dinheiro disponível na conta do usuário.

Empréstimo: Quantia emprestada ao titular da conta, que deve ser paga com juros.

Menu Inicial: Interface inicial do sistema onde o usuário pode escolher entre cadastrar uma nova conta, fazer login em uma conta existente ou sair do sistema.

Menu Pós-Login: Interface do sistema apresentada após o login, onde o usuário pode realizar operações bancárias como consultar saldo, depositar, sacar, transferir, gerenciar empréstimos e fechar a conta.

Operações Bancárias: Ações que um usuário pode realizar em sua conta, incluindo consulta de saldo, depósito, saque, transferência e gerenciamento de empréstimos.

Cadastro de Conta: Processo de criação de uma nova conta bancária no sistema.

Login: Processo de autenticação onde o usuário fornece o número da conta e a senha para acessar suas informações e realizar operações.

Depósito: Ato de adicionar uma quantia de dinheiro à conta bancária do usuário.

Saque: Ato de retirar uma quantia de dinheiro da conta bancária do usuário.

Transferência: Ato de enviar uma quantia de dinheiro de uma conta bancária para outra dentro do sistema.

Juros: Taxa adicional cobrada sobre o valor do empréstimo, neste caso, 10% do valor emprestado.

Quitação de Empréstimo: Pagamento do valor total devido de um empréstimo, incluindo os juros.

2. Descrição Geral do Sistema

2.1. Principais Envolvidos e suas Características

2.1.1. Usuários do Sistema

O programa se destina a usuários acadêmicos, que desejam ver a funcionalidade de um sistema bancário de forma simplificada.

2.1.2. Desenvolvedores do Sistema

Eu, Henrique Ulbrich de Souza.

2.2. Regras de Negócio

Cadastro de Conta

1. **Entrada de Dados:** Nome, CPF, Data de Nascimento, Número da Conta, Senha.
2. **Validação:** O número da conta deve ser único. CPF deve ter 11 caracteres.
3. **Inicialização:** Saldo e valor de empréstimo são inicializados como 0.
4. **Armazenamento:** A nova conta é adicionada ao início da lista de contas.

Login

1. **Autenticação:** O usuário deve fornecer o número da conta e a senha.
2. **Validação:** A combinação de número da conta e senha deve corresponder a uma conta existente.

Depósito

1. **Entrada de Dados:** Valor do depósito.
2. **Validação:** O valor deve ser positivo.
3. **Processamento:** O valor é adicionado ao saldo da conta.

Saque

1. **Entrada de Dados:** Valor do saque.
2. **Validação:** O valor não pode exceder o saldo disponível.
3. **Processamento:** O valor é subtraído do saldo da conta.

Transferência

1. **Entrada de Dados:** Número da conta de destino e valor da transferência.
2. **Validação:**
 - A conta de destino deve existir.
 - O valor não pode exceder o saldo disponível.
3. **Processamento:** O valor é subtraído do saldo da conta de origem e adicionado ao saldo da conta de destino.

Empréstimo

1. **Entrada de Dados:** Valor do empréstimo.
2. **Processamento:**
 - O valor solicitado é adicionado ao saldo da conta.
 - O valor com juros (10%) é adicionado ao valor de empréstimo da conta.

Pagamento de empréstimo

1. **Validação:**
 - A conta deve ter um empréstimo pendente.
 - O saldo deve ser suficiente para cobrir o valor do empréstimo.
2. **Processamento:** O saldo é subtraído pelo valor do empréstimo e o valor do empréstimo é zerado.

Consulta de saldo

1. **Exibição:** O saldo atual e o valor de empréstimos pendentes (se houver) são exibidos ao usuário.

Fechamento de conta

1. **Validação:** A conta não pode ter empréstimos pendentes.
2. **Processamento:** A conta é removida da lista de contas e a memória alocada é liberada.

3. Requisitos do Sistema

3.1. Requisitos Funcionais

Cadastro de Conta

- O sistema deve permitir o cadastro de uma nova conta bancária, solicitando o nome, CPF, data de nascimento, número da conta e senha.

Login

- O sistema deve permitir que o usuário faça login em uma conta existente, verificando o número da conta e a senha.

Operações Bancárias

- Após o login, o usuário deve ser capaz de:
 - Consultar o saldo da conta e o valor devido em empréstimos.
 - Depositar um valor na conta.
 - Sacar um valor da conta, desde que haja saldo suficiente.
 - Transferir dinheiro para outra conta existente no sistema, desde que haja saldo suficiente.
 - Realizar empréstimos com juros de 10%.
 - Pagar empréstimos, desde que haja saldo suficiente.
 - Fechar a conta, desde que não haja empréstimos pendentes.

3.2. Requisitos Não-Funcionais

Usabilidade

- O sistema deve ser intuitivo e fácil de usar, com menus claros e instruções para cada operação.

Performance

- O sistema deve responder rapidamente a cada operação, uma vez que os dados são mantidos na memória.

Segurança

- O sistema deve verificar a senha do usuário antes de permitir o acesso às operações de conta.
- O número da conta deve ser único para cada usuário.

4. Análise e Design

4.1. Arquitetura do Sistema

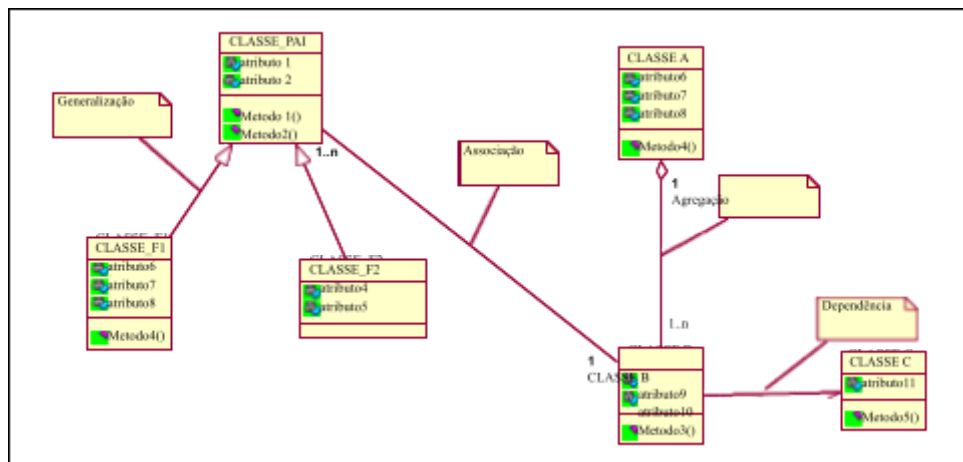
O programa funciona em qualquer computador, desde que o mesmo possua um compilador de código com suporte para linguagem de programação C.

4.2. Modelo do Domínio

Neste item deve ser apresentado o modelo do domínio, que representa um primeiro modelo conceitual do diagrama de classes. Posteriormente, esse diagrama deve ser validado e complementado para compor o diagrama de classes final.

O diagrama de classes deve possuir todas as classes identificadas do sistema, deve conter os atributos e métodos de cada classe, e os relacionamento entre elas.

A seguir é apresentada a notação básica de um diagrama de classes



Notação básica do diagrama de classes

4.3. Ambiente de Desenvolvimento

O programa foi desenvolvido em um computador que utiliza o sistema operacional windows. Foi utilizado o compilador CodeBlocks.

5. Testes

Este capítulo tem como objetivo identificar defeitos no sistema, validar as funções do sistema, verificar se os requisitos foram implementados de forma adequada e avaliar a qualidade do software.

5.1. Plano de Testes

1. Cadastro de Conta

Testes:

- Cadastro de uma conta com todos os campos válidos.
- Cadastro de uma conta com um CPF já existente.
- Cadastro de uma conta com um número de conta já existente.
- Cadastro de uma conta com campos obrigatórios vazios.

Expectativas:

- A conta deve ser cadastrada com sucesso quando todos os campos são válidos.
- Deve haver uma mensagem de erro ao tentar cadastrar uma conta com um CPF ou número de conta duplicados.
- Deve haver uma mensagem de erro quando campos obrigatórios estão vazios.

2. Login

Testes:

- Login com número de conta e senha corretos.
- Login com número de conta incorreto.
- Login com senha incorreta.
- Login com ambos incorretos.

Expectativas:

- O login deve ser bem-sucedido com credenciais corretas.
- Deve haver uma mensagem de erro para credenciais incorretas.

3. Depósito

Testes:

- Depósito de um valor positivo.
- Depósito de um valor zero ou negativo.

Expectativas:

- O saldo deve ser atualizado corretamente para valores positivos.
- Deve haver uma mensagem de erro para valores zero ou negativos.

4. Saque

Testes:

- Saque de um valor dentro do saldo disponível.
- Saque de um valor superior ao saldo disponível.
- Saque de um valor zero ou negativo.

Expectativas:

- O saldo deve ser atualizado corretamente para valores válidos dentro do saldo disponível.
- Deve haver uma mensagem de erro para tentativas de saque que excedam o saldo ou para valores zero ou negativos.

5. Transferência

Testes:

- Transferência para uma conta existente com saldo suficiente.
- Transferência para uma conta inexistente.
- Transferência de um valor superior ao saldo disponível.
- Transferência de um valor zero ou negativo.

Expectativas:

- O saldo das contas de origem e destino deve ser atualizado corretamente para transferências válidas.
- Deve haver uma mensagem de erro para contas inexistentes ou saldo insuficiente, bem como para valores zero ou negativos.

6. Empréstimo

Testes:

- Solicitação de um empréstimo de valor positivo.
- Solicitação de um empréstimo de valor zero ou negativo.

Expectativas:

- O saldo deve ser atualizado e o valor do empréstimo deve ser registrado corretamente para valores positivos.
- Deve haver uma mensagem de erro para valores zero ou negativos.

7. Pagamento de Empréstimo

Testes:

- Pagamento de um empréstimo com saldo suficiente.
- Pagamento de um empréstimo com saldo insuficiente.
- Tentativa de pagamento sem empréstimos pendentes.

Expectativas:

- O saldo deve ser atualizado corretamente e o valor do empréstimo deve ser zerado para saldos suficientes.
- Deve haver uma mensagem de erro para saldo insuficiente ou ausência de empréstimos pendentes.

8. Consulta de Saldo

Testes:

- Consulta de saldo com e sem empréstimos pendentes.

Expectativas:

- O saldo e os valores de empréstimos pendentes devem ser exibidos corretamente.

9. Fechamento de Conta

Testes:

- Fechamento de uma conta sem empréstimos pendentes.
- Tentativa de fechamento de uma conta com empréstimos pendentes.

Expectativas:

- A conta deve ser fechada corretamente se não houver empréstimos pendentes.
- Deve haver uma mensagem de erro para tentativas de fechamento com empréstimos pendentes.

5.2. Execução do Plano de Testes

Para cada caso de teste acima, execute as seguintes etapas:

1. **Configuração:** Prepare o ambiente de teste inicializando o programa.
2. **Execução:** Realize as ações especificadas em cada caso de teste.
3. **Verificação:** Compare os resultados obtidos com os resultados esperados.
4. **Registro:** Documente os resultados de cada teste, incluindo qualquer comportamento inesperado.

6. Implantação

Este capítulo tem como objetivo apresentar informações relevantes para a implantação e funcionamento do sistema.

6.1. Manual de Implantação

Requisitos - GCC (GNU Compiler Collection)

Instalação do GCC

Windows

1. Baixe e instale o Codeblocks.
2. dentro do Codeblocks abra um novo arquivo e cole o código do programa que está na mesma pasta desta documentação.
3. Execute o código e estará pronto para uso.

7. Manual do Usuário

1. Selecione a opção de registrar uma nova conta.
2. Faça o login após o registro.
3. Escolha quais funções você deseja utilizar, dentre elas temos saque, depósito, transferência, fechar conta ou voltar para o menu inicial.
4. Nas opção de depósito, saque e transferencia coloque o valor desejado(ex: 100). Não utilize pontuação nem caracteres especiais.
5. Só selecione a opção de deletar a conta caso queira fazer isso, não tem confirmação.

8. Conclusões e Considerações Finais

O projeto de sistema bancário desenvolvido em C abrange funcionalidades essenciais para a gestão de contas bancárias, incluindo cadastro, login, depósito, saque, transferência, empréstimos e consulta de saldo. O desenvolvimento e a execução de um plano de testes abrangente garantiram que as funcionalidades principais do sistema foram verificadas e validadas conforme as regras de negócio especificadas.

O programa não possui criptografia de senhas. É recomendado a implementação da criptografia para uma segurança maior caso o programa seja implementado para funcionar online. Atualmente o programa não possui sistema de banco de dados, ou seja, assim que for fechado todas as informações inseridas são perdidas. O programa funciona via prompt, não possui interface.

Referências

https://www.youtube.com/watch?v=DeftMK6sdyQ&ab_channel=IsabelleCruz
<https://awari.com.br/aprenda-a-desenvolver-um-projeto-em-c-com-banco-de-dados/>