

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
SÃO PAULO**

MARIEL SOARES NICIOLI

PROJETO FINAL: SISTEMA DE CAIXA ELETRÔNICO

CAMPOS DO JORDÃO

2024

RESUMO

Este trabalho descreve o desenvolvimento de um sistema simulado de caixa eletrônico como parte do projeto final da disciplina de Programação Orientada a Objetos. Utilizando a linguagem C++, o sistema implementa funcionalidades básicas de um terminal bancário, como consulta de saldo, saques e depósitos, com uma arquitetura modular baseada nos princípios da programação orientada a objetos. O desenvolvimento foi realizado no ambiente Visual Studio Code, com o compilador MinGW GCC, e resultou em um sistema funcional e validado, permitindo interação eficiente no terminal. O sistema alcançou os objetivos propostos, funcionando de forma eficiente e apresentando validações adequadas para as transações simuladas. Além disso, o desenvolvimento proporcionou uma aplicação prática dos conceitos teóricos aprendidos e oferece perspectivas para melhorias futuras, como persistência de dados, interface gráfica e inclusão de novas funcionalidades.

Palavras-Chave: Programação Orientada a Objetos; Sistema Bancário; Caixa Eletrônico; C++.

ABSTRACT

his work describes the development of a simulated ATM system as part of the final project for the Object-Oriented Programming course. Using the C++ language, the system implements basic functionalities of a banking terminal, such as balance inquiry, withdrawals, and deposits, with a modular architecture based on object-oriented programming principles. The development was carried out in the Visual Studio Code environment, using the MinGW GCC compiler, resulting in a functional and validated system that allows efficient interaction in the terminal. The system achieved the proposed objectives, operating efficiently and providing proper validation for the simulated transactions. Additionally, the development offered a practical application of the theoretical concepts learned and presents opportunities for future improvements, such as data persistence, a graphical user interface, and the inclusion of new features.

Keywords: Object-Oriented Programming; Banking System; ATM; C++.

LISTA DE ILUSTRAÇÕES

FIGURA 1 – Menu Principal	16
FIGURA 2 – Opção de “Sacar”	16
FIGURA 3 – Opção de “Ver meu saldo”	

SUMÁRIO

1	INTRODUÇÃO	7
1.1	Objetivos	7
1.2	Justificativa	8
1.3	Aspectos Metodológicos	8
1.4	Aporte Teórico	9
2	METODOLOGIA	10
2.1	Projeto Proposto	10
2.2	Ferramentas Utilizadas	10
2.3	Descrição do Projeto	11
3	RESULTADOS OBTIDOS	14
4	CONCLUSÃO	15
	REFERÊNCIAS	24

1 INTRODUÇÃO

O avanço das tecnologias computacionais trouxe diversas inovações para o cotidiano das pessoas, incluindo sistemas que automatizam tarefas complexas e aumentam a eficiência em serviços bancários. Dentre esses sistemas, destacam-se os caixas eletrônicos, que permitem a realização de operações financeiras de forma autônoma e segura. Este trabalho apresenta o desenvolvimento de um sistema de caixa eletrônico utilizando a linguagem C++ e orientado a objetos, com o objetivo de aplicar os conceitos teóricos estudados em um projeto prático e funcional.

A partir de uma abordagem pedagógica, este projeto busca modelar as principais funcionalidades de um caixa eletrônico, enfatizando os princípios de programação orientada a objetos, como abstração, encapsulamento, herança e polimorfismo. Este documento descreve detalhadamente todas as etapas de desenvolvimento do sistema, desde a análise inicial até a implementação e resultados obtidos, respeitando as normas técnicas da ABNT.

1.1 Objetivos

O objetivo principal deste trabalho é desenvolver um sistema simulado de caixa eletrônico utilizando a linguagem C++ e técnicas de programação orientada a objetos. Os objetivos específicos incluem:

- Compreender e aplicar conceitos fundamentais de orientação a objetos.
- Implementar funcionalidades básicas de um caixa eletrônico, incluindo consulta de saldo, saques, depósitos e gerenciamento de contas.
- Documentar todas as etapas do desenvolvimento, promovendo o aprendizado e a reflexão sobre o processo.

1.2 Justificativa

A escolha do desenvolvimento de um sistema de caixa eletrônico como tema do projeto justifica-se tanto por sua relevância prática e educacional quanto por ter sido o sistema escolhido como projeto final da disciplina. A escolha reflete a complexidade moderada do problema, permitindo a aplicação direta dos conceitos teóricos abordados ao longo do curso.

Sistemas bancários são amplamente utilizados no dia a dia, e sua modelagem em um ambiente de aprendizado fornece um exemplo tangível de como conceitos teóricos podem ser aplicados para resolver problemas reais.

A utilização de C++ como linguagem de implementação complementa esse aprendizado, permitindo explorar aspectos fundamentais, como controle de memória e gestão de objetos. Dessa forma, este projeto cumpre o duplo propósito de atender aos requisitos acadêmicos e fortalecer competências técnicas relevantes para o mercado de trabalho.

1.3 Aspectos Metodológicos

O desenvolvimento do sistema seguiu uma abordagem incremental, dividida em etapas que permitiram a construção de um projeto estruturado e funcional. As principais etapas foram:

1. **Análise do problema:** Identificação dos requisitos do sistema e definição das funcionalidades principais.
2. **Implementação:** Codificação do sistema utilizando a linguagem C++ e aplicação de técnicas de orientação a objetos.
3. **Testes e validação:** Verificação das funcionalidades implementadas para garantir o funcionamento adequado.
4. **Documentação:** Produção de relatórios e do código fonte comentado, seguindo boas práticas de escrita técnica.

1.4 Aporte Teórico

Este trabalho fundamenta-se nos princípios da programação orientada a objetos (POO), uma abordagem amplamente utilizada no desenvolvimento de sistemas computacionais modernos. A POO permite modelar problemas do mundo real de forma intuitiva e estruturada, promovendo a reutilização de código, a modularização e a manutenção facilitada.

A base teórica para o desenvolvimento do projeto foi extraída de obras como Como Programar em C++ de Deitel, que apresenta os fundamentos de C++ aplicados à POO, e de artigos acadêmicos e manuais de engenharia de software que destacam as boas práticas de desenvolvimento.

Além disso, o projeto alinha-se aos princípios do ciclo de vida do desenvolvimento de software, abordando as etapas de análise, implementação, teste e documentação.

2 METODOLOGIA

Nesta seção, são apresentadas as considerações iniciais sobre o projeto, as ferramentas utilizadas para o desenvolvimento e uma descrição detalhada do sistema de caixa eletrônico implementado. A metodologia seguiu uma abordagem incremental, priorizando a aplicação prática dos conceitos teóricos da programação orientada a objetos (POO).

Nesta seção serão apresentadas detalhadamente a metodologia utilizada neste trabalho, porque esta foi a escolhida (porque esta é mais adequada que outras) e suas etapas, os documentos referentes ao sistema proposto, como eles foram elaborados e demais artefatos referentes a este projeto.

2.1 Projeto Proposto

O projeto proposto consiste no desenvolvimento de um sistema simulado de caixa eletrônico que implementa as funcionalidades básicas de um terminal bancário, como consulta de saldo, saques e depósitos. Este sistema foi desenvolvido em C++ com base nos princípios da programação orientada a objetos, explorando conceitos do princípio.

O objetivo do projeto é permitir que o aluno aplique o conhecimento adquirido durante a disciplina de Programação Orientada a Objetos em um problema do mundo real, simulando o funcionamento de um caixa eletrônico por meio de interações textuais simples entre o usuário e o sistema.

2.2 Ferramentas Utilizadas

Para o desenvolvimento do sistema de caixa eletrônico, foram utilizadas ferramentas que oferecem suporte eficiente à programação em C++ e à aplicação de conceitos de programação orientada a objetos. As principais ferramentas e recursos empregados foram:

- **Visual Studio Code (VS Code):** Escolhido como ambiente de desenvolvimento integrado (IDE) por sua interface leve, extensibilidade e suporte a diversas linguagens de programação, incluindo C++. Ele também oferece integração com depuradores e outras ferramentas essenciais para a produtividade no desenvolvimento.
- **Compilador MinGW GCC:** Utilizado para compilar e executar o código C++ diretamente no ambiente do VS Code, garantindo a compatibilidade e a performance necessárias para a execução do projeto.
- **Extensões do Visual Studio Code:** Foram instaladas extensões como C/C++ e Code Runner, que proporcionam suporte à sintaxe, dicas de código e execução simplificada.

2.3 Descrição do Projeto

O projeto desenvolvido tem como objetivo a criação de um sistema simulado de caixa eletrônico, em que são implementadas as funcionalidades principais de um terminal bancário. Este sistema foi projetado para operar de forma simples e eficiente, simulando o comportamento de um caixa eletrônico real, permitindo ao usuário realizar operações bancárias básicas, como consulta de saldo, saques e depósitos.

Estrutura do Sistema

O sistema foi estruturado utilizando a abordagem de programação orientada a objetos (POO), com o intuito de dividir a aplicação em módulos independentes e reutilizáveis. As classes que compõem o sistema foram projetadas para representar as entidades do mundo real, como contas bancárias e transações. Abaixo, é descrita a função de cada uma das principais classes envolvidas no sistema:

1. Classe ATM

A classe ATM é responsável por gerenciar a interação entre o usuário e o sistema. Ela atua como a interface principal, sendo a classe que lida com as entradas do usuário e apresenta o menu de opções, como a consulta de saldo, o saque e o depósito. A classe também realiza a validação das credenciais fornecidas

pelo usuário, como o número da conta e a senha

2. Classe BankDatabase

A classe BankDatabase simula a parte do banco de dados, armazenando as informações sobre as contas, como número de conta, saldo e senha. Ela é responsável por fornecer os dados necessários para as operações, como verificar o saldo de uma conta ou atualizar o saldo após uma transação.

3. Classe Account

A classe Account representa uma conta bancária individual. Ela armazena os dados fundamentais de cada conta, como o número da conta, a senha e o saldo. Cada conta é tratada como um objeto, o que permite que as operações sejam realizadas de forma encapsulada e segura.

4. Classe Transaction (Classe Base)

A classe Transaction é uma classe base abstrata para representar qualquer transação realizada no caixa eletrônico. Ela possui métodos comuns a todas as transações, como a execução da transação e a impressão de um recibo.

4.1 Subclasses de Transaction

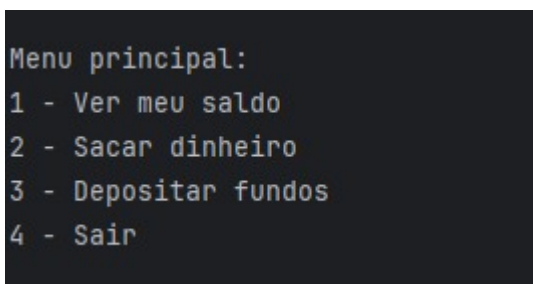
A partir da classe Transaction, foram desenvolvidas subclasses específicas para cada tipo de transação:

- **BalanceInquiry:** Realiza a consulta de saldo. Quando o usuário escolhe essa opção, o sistema exibe o saldo atual da sua conta.
- **Withdrawal:** Realiza o saque de um valor informado pelo usuário. A transação verifica se o saldo é suficiente antes de permitir o saque e ajusta o saldo após a operação.
- **Deposit:** Permite o depósito de um valor na conta do usuário. A operação

também ajusta o saldo da conta após a realização do depósito.

Fluxo de Execução do Sistema

1. O usuário insere seu número de conta e senha no caixa eletrônico, sendo validado pela classe ATM.
2. Após a validação, o sistema exibe um menu de opções, permitindo que o usuário escolha entre as transações disponíveis (consultar saldo, sacar ou depositar).
3. O sistema processa a operação escolhida, interagindo com as classes BankDatabase, Account e as transações específicas (BalanceInquiry, Withdrawal ou Deposit).
4. O resultado da transação é apresentado ao usuário, e o sistema volta ao menu inicial ou encerra a sessão, caso o usuário deseje.

A screenshot of a terminal window showing a main menu. The text is as follows:

```
Menu principal:  
1 - Ver meu saldo  
2 - Sacar dinheiro  
3 - Depositar fundos  
4 - Sair
```

Figura 1 – Menu Principal

```
Menu principal:
1 - Ver meu saldo
2 - Sacar dinheiro
3 - Depositar fundos
4 - Sair

Insira uma opcao:2

Opcoes de retirada:
1 - $20
2 - $40
3 - $60
4 - $100
5 - $200
6 - Cancelar transacao

Escolha uma opcao de retirada (1-6):2

Por favor, retire seu dinheiro do caixa eletrônico.
```

Figura 2 – Opção de “Sacar”

```
Menu principal:
1 - Ver meu saldo
2 - Sacar dinheiro
3 - Depositar fundos
4 - Sair

Insira uma opcao:1

Informacoes de saldo:
- Saldo disponivel: $1000.00
- Saldo total: $1200.00
```

Figura 3 – Opção de “Ver meu saldo”

3 RESULTADOS OBTIDOS

Nesta seção, são apresentados os resultados do sistema de caixa eletrônico desenvolvido. São exibidas capturas de tela que ilustram o funcionamento das principais funcionalidades implementadas.

4 CONCLUSÃO

O sistema de caixa eletrônico foi implementado com sucesso, atendendo aos requisitos iniciais do projeto. A utilização de conceitos de POO permitiu a construção de um código modular, reutilizável e fácil de entender. Cada operação do sistema foi desenvolvida de maneira a garantir a integridade das transações, com verificações adequadas de saldo antes de permitir saques e atualizações de saldo ao realizar depósitos.

Os testes realizados demonstraram que o sistema é capaz de executar corretamente as transações básicas, proporcionando uma experiência de usuário satisfatória. O funcionamento das principais funcionalidades foi validado através de interações simples no terminal, garantindo que o fluxo de operações está fluindo de acordo com o esperado.

Desta forma, o projeto foi uma excelente oportunidade para consolidar os conhecimentos adquiridos durante a disciplina de Programação Orientada a Objetos. Através dele, foi possível aplicar de forma prática conceitos fundamentais, como classes, objetos, herança, encapsulamento e polimorfismo. Além disso, o uso do C++ para desenvolver um sistema realista, mesmo que simulado, fortaleceu a compreensão de aspectos técnicos como manipulação de memória e controle de fluxo de execução.

Sugestões de Melhorias

Embora o sistema tenha atendido aos objetivos propostos, há várias melhorias que podem ser implementadas para torná-lo mais robusto e funcional. Algumas sugestões incluem:

- **Interface Gráfico):** O sistema atual foi desenvolvido em modo texto. A implementação de uma interface gráfica (GUI) permitiria uma experiência mais intuitiva e interativa para o usuário, usando bibliotecas como Qt ou SFML.
- **Persistência de Dados:** O sistema simula um banco de dados simples, mas para um sistema real, seria necessário utilizar um banco de dados persistente, como SQLite ou MySQL, para armazenar as informações das contas de forma mais robusta e segura.

- **Implementação de Transações de Transferência:** Uma possível melhoria seria a adição de uma funcionalidade de transferência entre contas, permitindo que os usuários transfiram valores para outras contas bancárias.

REFERÊNCIAS

A. LIVROS:

DEITEL, Paul; DEITEL, Harvey. **C++**: Como Programar. 10. ed. São Paulo: Pearson, 2017.