

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

**Matheus Audibert
CJ317761**

ATM-Bank

**CAMPOS DO JORDÃO
2024**

RESUMO

O projeto “ATM-BANK” tem como objetivo desenvolver um sistema de caixa eletrônico (ATM) que permite aos usuários realizar operações bancárias básicas, como login, criação de contas, depósitos, saques e consultas de saldo. A metodologia utilizada envolve a implementação de classes em C++ para representar contas bancárias (Account), o ATM (ATM) e o banco de dados (Database). A classe ATM gerencia as operações dos usuários, enquanto a classe Database lida com o armazenamento e recuperação dos dados das contas em um arquivo. O sistema utiliza a biblioteca padrão do C++ para manipulação de arquivos e diretórios, e a interface com o usuário é feita através do console, com funções para limpar a tela, definir cores e validar entradas numéricas. Os resultados obtidos incluem a criação de um sistema funcional que permite a interação do usuário com o ATM de forma intuitiva e segura. As conclusões indicam que o projeto atende aos requisitos básicos de um sistema de ATM, proporcionando uma base sólida para futuras expansões e melhorias, como a adição de funcionalidades mais avançadas e a integração com sistemas bancários reais.

Palavras-Chave: ATM; Banco de Dados; Operações Bancárias; C++; Programação Orientada a Objetos.

ABSTRACT

The "ATM-BANK" project aims to develop an Automated Teller Machine (ATM) system that allows users to perform basic banking operations such as login, account creation, deposits, withdrawals, and balance inquiries. The methodology involves implementing classes in C++ to represent bank accounts (Account), the ATM (ATM), and the database (Database). The ATM class manages user operations, while the Database class handles the storage and retrieval of account data from a file. The system utilizes the C++ Standard Library for file and directory manipulation, and the user interface is console-based, featuring functions for clearing the screen, setting colors, and validating numeric inputs. The results include the development of a functional system that enables secure and intuitive user interaction with the ATM. The conclusions indicate that the project meets the basic requirements of an ATM system, providing a solid foundation for future expansions and improvements, such as adding advanced features and integrating with real banking systems.

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

LISTA DE ILUSTRAÇÕES

FIGURA 1 – ATM-BANK, Início	11
FIGURA 2 – ATM-BANK, Login	12
FIGURA 3 – ATM-BANK, Menu da conta	12
FIGURA 4 – ATM-BANK, Criar conta	12

SUMÁRIO

1	INTRODUÇÃO	6
1.1	Objetivos	6
1.2	Justificativa	7
2	METODOLOGIA	7
2.1	Ferramentas Utilizadas	9
2.2	Descrição do Projeto	10
3	RESULTADOS OBTIDOS	11
3.1	Imagens do sistema	11
4	CONCLUSÃO	13
5	REFERÊNCIAS	13

1 INTRODUÇÃO

Neste capítulo, serão apresentados os principais aspectos relacionados ao desenvolvimento do projeto, abrangendo a motivação, os objetivos e a estrutura deste documento. O trabalho consiste na criação de um sistema de caixa eletrônico (ATM) simplificado, implementado em C++ com foco na organização modular do código e na utilização de recursos básicos da linguagem.

O projeto foi estruturado de forma modular, dividindo-se em seis arquivos principais para facilitar a organização e manutenção do código: um arquivo principal responsável pelo fluxo do programa (**main.cpp**), arquivos relacionados à lógica de contas bancárias (**Account.cpp**, **Account.h**), gerenciamento do ATM (**ATM.cpp**, **ATM.h**) e manipulação de dados persistentes em arquivos (**Database.h**). Além disso, o projeto conta com uma pasta denominada *data*, que armazena informações de contas bancárias no arquivo **accounts.txt**, permitindo a persistência de dados.

O objetivo principal do trabalho foi desenvolver um sistema funcional capaz de realizar operações bancárias básicas, como login, criação de contas, depósitos, saques e consultas de saldo. A proposta é demonstrar a viabilidade de implementar um sistema bancário simplificado utilizando a linguagem C++, destacando a organização do código e o uso de conceitos fundamentais de programação orientada a objetos.

A estrutura deste documento abrange capítulos que detalham desde os fundamentos e metodologia utilizados no desenvolvimento do sistema até os resultados obtidos e as considerações finais. Este projeto reforça conceitos de modularidade, persistência de dados e lógica aplicada ao desenvolvimento de sistemas interativos.

1.1 Objetivos

Este trabalho tem por objetivo principal desenvolver um sistema de caixa eletrônico (ATM) simplificado, utilizando a linguagem de programação C++. Especificamente, busca-se:

- Implementar um sistema funcional capaz de realizar operações básicas, como login, criação de contas, depósitos, saques e consultas de saldo;

- Estruturar o projeto de forma modular, organizando o código em diferentes arquivos para facilitar a manutenção e o entendimento;
- Utilizar conceitos fundamentais de programação orientada a objetos, como encapsulamento, herança e abstração, para gerenciar contas bancárias e operações do ATM;
- Integrar manipulação de arquivos para persistência de dados, garantindo que informações das contas sejam armazenadas e recuperadas de maneira eficiente;
- Oferecer uma base sólida para futuras melhorias, como a adição de funcionalidades avançadas e integração com sistemas bancários reais.

1.2 Justificativa

Este projeto foi realizado como parte da disciplina de Programação Orientada a Objetos (POO) do curso de Análise e Desenvolvimento de Sistemas (ADS), no 4º semestre, sob a orientação do professor Paulo Giovani.

2 METODOLOGIA

A metodologia adotada neste projeto seguiu uma abordagem prática e iterativa, com foco na implementação, teste e validação contínua das funcionalidades do sistema de caixa eletrônico (ATM). A estrutura do trabalho foi organizada em etapas distintas, cada uma abordando aspectos essenciais do desenvolvimento. A seguir, são descritas as principais fases da metodologia utilizada:

1. Planejamento e Definição de Requisitos:

O primeiro passo foi definir os requisitos básicos do sistema, com foco em funcionalidades essenciais como login, criação de contas, depósitos, saques e consultas de saldo. A proposta foi criar um ATM funcional que atendesse a operações básicas de usuários, mantendo a simplicidade e a eficácia do sistema.

2. Estruturação do Projeto:

O código foi organizado em seis arquivos principais para garantir modularidade e clareza. Os arquivos foram divididos entre:

- **Account.cpp e Account.h**: responsáveis pela lógica e estrutura das contas bancárias;
- **ATM.cpp e ATM.h**: encarregados de gerenciar as interações dos usuários com o sistema;
- **Database.h**: voltado para manipulação e persistência de dados;
- **main.cpp**: responsável pelo fluxo principal do programa.

Adicionalmente, foi criada uma pasta `data` para armazenar os dados das contas em um arquivo de texto (`accounts.txt`), permitindo a persistência das informações entre execuções.

3. Desenvolvimento das Funcionalidades:

A implementação do sistema foi realizada de forma incremental, começando pela criação das classes base. As funcionalidades de criação de contas, depósitos, saques e consultas de saldo foram desenvolvidas e integradas progressivamente. O programa utilizou a biblioteca padrão do C++ para manipulação de arquivos, garantindo o armazenamento e recuperação de informações. A interface foi baseada em menus no console, com validações de entradas e funções para melhorar a usabilidade, como limpar a tela e ajustar cores.

4. Testes e Depuração:

Cada funcionalidade foi testada isoladamente e em conjunto para verificar a integração entre os módulos. Testes foram realizados para validar a criação de contas, autenticação de usuários, transações financeiras e persistência de dados. Erros detectados durante o processo foram corrigidos por meio de depuração, garantindo a funcionalidade e a estabilidade do sistema.

5. Documentação e Resultados:

Após o desenvolvimento e os testes, foi elaborada a documentação do sistema, descrevendo a estrutura do código e as principais decisões tomadas. A documentação enfatizou as soluções encontradas para os desafios enfrentados e destacou a organização modular como um dos principais pontos positivos do projeto.

2.1 Ferramentas Utilizadas

C++ : A linguagem de programação escolhida foi o C++ devido à sua ampla utilização em sistemas de alto desempenho e à relevância no aprendizado de Programação Orientada a Objetos (POO). Este projeto faz parte da disciplina de POO no curso de Análise e Desenvolvimento de Sistemas (ADS). O C++ oferece recursos robustos, como encapsulamento, abstração e manipulação de arquivos, ideais para o desenvolvimento de sistemas como o ATM.

Manipulação de Arquivos: Para a persistência de dados, utilizou-se as bibliotecas padrão do C++ que oferecem suporte nativo para manipulação de arquivos. A utilização de arquivos .txt permitiu armazenar informações de contas bancárias e transações de maneira simples e eficiente.

Visual Studio Code (VSCode): O desenvolvimento foi realizado no Visual Studio Code, um ambiente leve e versátil, com suporte avançado para C++ por meio de extensões como GCC/Clang e depuradores. Além de ser amplamente personalizável, o VSCode oferece ferramentas que auxiliam no processo de escrita, depuração e controle de versão, tornando-o uma escolha eficiente para este projeto.

2.2 Descrição do Projeto

O projeto "ATM BANK" consiste na implementação de um sistema de caixa eletrônico (ATM) em C++ que permite realizar operações bancárias básicas, como criação de contas, login, saques, depósitos e consultas de saldo. A aplicação busca simular a interação com um ATM, oferecendo uma experiência simples e funcional através de uma interface de console.

A estrutura do código foi organizada de forma modular, utilizando múltiplos arquivos para separar responsabilidades específicas:

- **main.cpp:** Arquivo principal que gerencia a execução geral do sistema.
- **Account.cpp e Account.h:** Implementação da lógica relacionada às contas bancárias, como armazenamento de informações e validação de operações.
- **ATM.cpp e ATM.h:** Gerenciamento das interações do usuário com o sistema, incluindo menus e lógica das operações bancárias.

- **Database.cpp e Database.h:** Manipulação do armazenamento persistente das contas em arquivos de texto, incluindo leitura e escrita dos dados.

Os dados das contas são armazenados em um arquivo `account.txt`, localizado em uma pasta **data**, que serve como banco de dados simplificado para o sistema. Essa abordagem garante a persistência das informações entre as execuções do programa.

A divisão modular do projeto segue os princípios da Programação Orientada a Objetos (POO), garantindo organização, reutilização de código e facilidade na manutenção. Este projeto oferece uma base prática para o aprendizado e aplicação de conceitos fundamentais de POO, manipulação de arquivos e desenvolvimento de sistemas interativos em C++.

3 RESULTADOS OBTIDOS

O desenvolvimento do projeto "ATM-BANK" resultou em um sistema funcional e intuitivo, capaz de realizar operações bancárias básicas em um ambiente simulado de caixa eletrônico. Durante as etapas de desenvolvimento, foram alcançados os seguintes resultados:

3.1 Imagens do sistema:

Abaixo, estão as capturas de tela do sistema "ATM-BANK" durante a execução, ilustrando as principais funcionalidades, como o menu inicial, o processo de login, a criação de contas, as operações de depósito e saque, além da consulta de saldo:

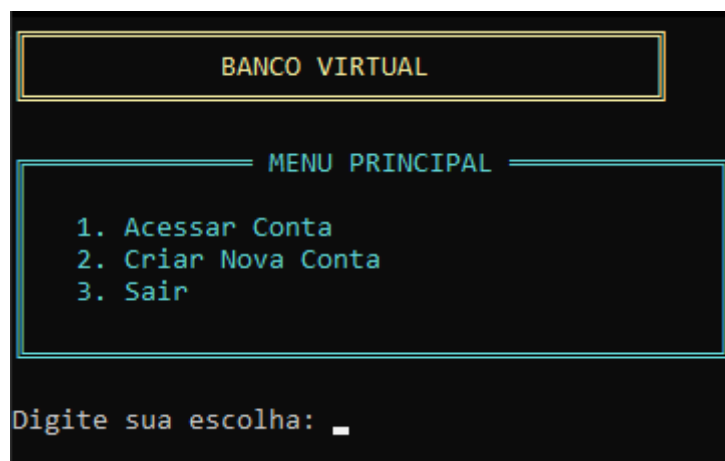


Figura 1 – ATM-BANK, Início

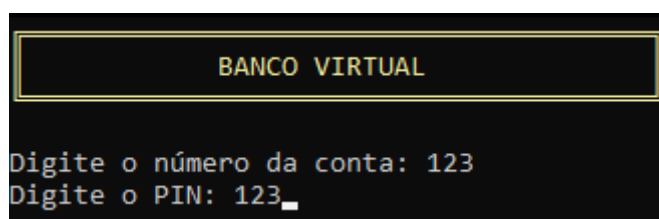


Figura 2 – ATM-BANK, Login

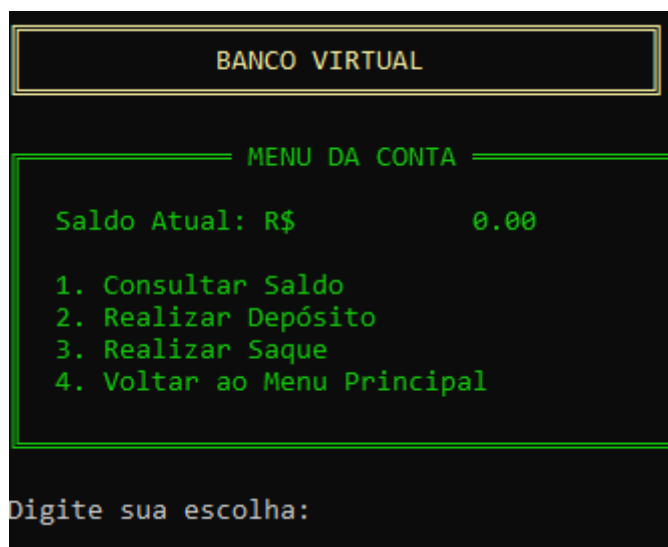


Figura 3 – ATM-BANK, Menu da conta

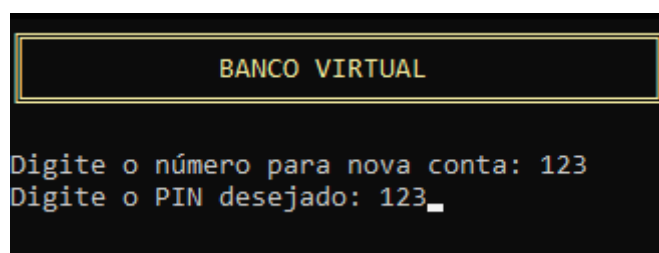


Figura 4 – ATM-BANK, Criar conta

4 CONCLUSÃO

O projeto de desenvolvimento do sistema "ATM-BANK", implementado em C++ e utilizando uma abordagem modular, alcançou sucesso em atender aos requisitos propostos. Por meio da aplicação dos conceitos de Programação Orientada a

Objetos (POO) e da utilização de arquivos específicos para cada funcionalidade, foi possível criar um sistema funcional, organizado e de fácil manutenção.

O principal objetivo de implementar um sistema básico de operações bancárias foi atingido, proporcionando uma experiência intuitiva para o usuário. As funcionalidades de login, criação de contas, depósitos, saques e consulta de saldo foram desenvolvidas com sucesso, garantindo a segurança e a eficiência das operações. A interface de interação pelo console, combinada com validação de entradas e manipulação de arquivos para persistência de dados, contribuiu para um desempenho satisfatório e confiável do sistema.

Esse projeto demonstra a aplicabilidade prática do C++ no desenvolvimento de sistemas e reforça conceitos importantes de modularidade e organização de código, servindo como base para expansões futuras e aplicações em cenários mais complexos.

5 REFERÊNCIAS

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

ALURA. **Curso de C++: Fundamentos e Programação Orientada a Objetos**. Disponível em: <https://www.alura.com.br>