

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

CJOPROO - Programação Orientada a Objetos

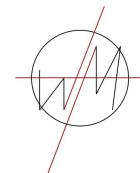
Professor: Paulo

## **Caixa Eletrônico**

Wedyner Rodrigo Maciel – CJ3019462

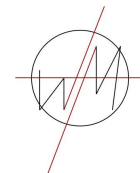
Campos do Jordão

2024



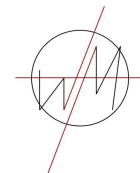
## Sumário

<b>1. - Resumo.....</b>	<b>4</b>
<b>2. - Introdução.....</b>	<b>4</b>
<b>3. - Metodologia.....</b>	<b>6</b>
3.1. - Análise e Planejamento Inicial.....	6
3.2. - Ferramentas Utilizadas.....	6
3.3. - Estruturação e Implementação do Código.....	7
3.4. - Testes e Validação.....	8
3.5. - Descrição do Projeto Desenvolvido.....	8
<b>4. - Resultados Obtidos.....</b>	<b>9</b>
4.1. - Autenticação de Usuário.....	9
4.2. - Consulta de Saldo.....	9
4.3. - Realização de Saques e Depósitos.....	10
4.4. - Histórico de Transações.....	10
4.5. - Alteração de Senha.....	11
<b>5. - Conclusão.....</b>	<b>12</b>
<b>6. - Diagrama de Classes.....</b>	<b>13</b>
<b>7. - Referências Bibliográficas.....</b>	<b>15</b>



## Imagens

Imagem - Diagrama de Classe. Elaborado pelo autor.....13



## 1. - Resumo

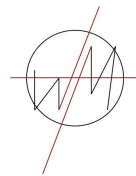
O presente trabalho descreve o desenvolvimento de um sistema de caixa eletrônico em linguagem C++, que visa simular funcionalidades básicas de uma conta bancária, como autenticação de usuário, consulta de saldo, realização de saques, depósitos, exibição de histórico de transações e alteração de senha. Para a implementação, foram aplicados conceitos fundamentais de Programação Orientada a Objetos (POO), com a criação de uma estrutura modular que promove a reutilização e manutenção simplificada do código. O sistema foi desenvolvido com o propósito de fornecer uma interface prática e segura para o usuário, atendendo às necessidades de um ambiente bancário simplificado.

A escolha da linguagem C++ para o projeto permitiu explorar recursos robustos de encapsulamento, abstração e modularidade, favorecendo uma clara separação de responsabilidades entre as classes e os métodos implementados. A classe principal, `Conta`, representa a entidade bancária e centraliza atributos como número da conta, senha, saldo e histórico de transações, garantindo o controle e a segurança de dados do usuário. Para o formato de valores monetários, o sistema incorpora o uso de separadores de milhar, facilitando a visualização de quantias de forma clara e intuitiva, característica essencial para sistemas financeiros.

Além de uma aplicação prática de técnicas de POO, este projeto também ilustra a importância da organização do código e da clareza na interface de interação, visando uma experiência de usuário eficiente e intuitiva. O trabalho traz contribuições educacionais valiosas, destacando o uso de programação modular em C++ e explorando práticas recomendadas de desenvolvimento de software para simulação de um caixa eletrônico funcional e expansível.

## 2. - Introdução

Este trabalho apresenta o desenvolvimento de um sistema de caixa eletrônico simulado, implementado na linguagem de programação C++, com o objetivo de demonstrar e aplicar conceitos fundamentais de Programação Orientada



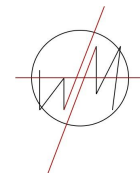
a Objetos (POO). A criação de um sistema como este tem grande relevância no contexto educacional, pois permite o entendimento prático de como soluções de software são projetadas e organizadas, promovendo o aprendizado de conceitos de encapsulamento, modularidade e segurança na manipulação de dados financeiros.

O objetivo principal do projeto é construir uma aplicação que emule funcionalidades essenciais de um caixa eletrônico, incluindo operações de autenticação de usuário, consulta de saldo, saques, depósitos, exibição do histórico de transações e alteração de senha. Cada uma dessas funcionalidades é implementada como parte de uma estrutura modular e encapsulada, baseada em classes e métodos, onde a principal classe do sistema, `Conta`, representa a entidade bancária e concentra a lógica e o gerenciamento das operações realizadas pelo usuário.

A justificativa para a criação deste sistema reside na necessidade de projetos práticos que ofereçam aos estudantes e desenvolvedores uma aplicação realista do uso de POO e da linguagem C++. A implementação de um sistema bancário simulado oferece um cenário rico para o aprendizado, abordando desde a estruturação de classes até a manipulação de entradas e saídas de dados, essenciais em sistemas de software de uso diário. Além disso, desenvolver funcionalidades que lidam com dados sensíveis, como senhas e saldos, reforça a importância de boas práticas de programação, especialmente na manipulação de valores monetários com precisão e apresentação adequada.

Metodologicamente, o projeto foi desenvolvido seguindo uma abordagem incremental e modular, na qual cada funcionalidade foi projetada de forma independente e integrada ao sistema principal, possibilitando que o código seja facilmente compreendido, mantido e expandido. Para a formatação de valores monetários, o sistema utiliza separadores de milhar e apresenta os valores com duas casas decimais, permitindo uma visualização clara e precisa dos dados financeiros, alinhada com padrões comuns no setor bancário.

O embasamento teórico para este trabalho inclui literatura voltada à Programação Orientada a Objetos e ao desenvolvimento em C++, bem como recomendações de usabilidade e formatação financeira aplicáveis a interfaces de usuário bancárias. Desta forma, a introdução deste sistema em um contexto educacional visa proporcionar uma compreensão mais sólida e prática sobre os



elementos essenciais de design e codificação, além de ressaltar a importância da segurança e clareza nos sistemas voltados para o gerenciamento financeiro.

### **3. - Metodologia**

O desenvolvimento deste projeto seguiu uma abordagem incremental e modular, com o objetivo de criar uma aplicação de caixa eletrônico capaz de simular funcionalidades bancárias essenciais. A metodologia empregada neste trabalho compreendeu a análise, o design, a implementação e os testes do sistema, sempre buscando aderir aos princípios da Programação Orientada a Objetos (POO) para garantir a robustez, modularidade e reutilização do código.

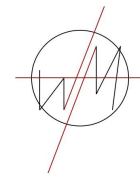
#### **3.1. - Análise e Planejamento Inicial**

O primeiro passo foi identificar os requisitos essenciais de um sistema de caixa eletrônico. Esta etapa incluiu a definição das funcionalidades necessárias, como autenticação de usuário, consulta de saldo, saque, depósito, exibição do histórico de transações e alteração de senha. Além disso, foi planejado um método de formatação para valores monetários, garantindo que cada transação fosse exibida com separadores de milhar e duas casas decimais para maior clareza e precisão. Com base nesses requisitos, foi criado um plano de desenvolvimento que delineou cada funcionalidade em blocos modulares, permitindo a implementação gradual e testes a cada etapa.

#### **3.2. - Ferramentas Utilizadas**

Para o desenvolvimento deste sistema, foram utilizadas as seguintes ferramentas:

- Linguagem de Programação C++: O C++ foi escolhido pela sua robustez e suporte completo à Programação Orientada a Objetos, além de permitir o controle de memória e a criação de uma aplicação com bom desempenho.



- Bibliotecas Padrão do C++: As bibliotecas ``iostream``, ``iomanip``, ``vector`` e ``string`` foram empregadas para operações de entrada/saída, formatação de valores monetários, manipulação de listas (histórico de transações) e manipulação de strings, respectivamente.

- Ambiente de Desenvolvimento Integrado (IDE): Foi utilizado o Visual Studio Code, que facilita o processo de codificação, depuração e teste da aplicação.

### 3.3. - Estruturação e Implementação do Código

Com base nos requisitos, o sistema foi estruturado em classes, sendo a principal a classe ``Conta``, que representa a entidade central do sistema. A classe ``Conta`` foi projetada com métodos específicos para cada funcionalidade do caixa eletrônico:

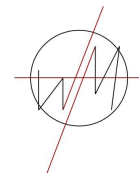
- Autenticação: Um método de autenticação simples foi implementado, onde o número da conta e a senha são verificados. Este recurso simula a segurança básica necessária para um sistema bancário.

- Manipulação de Transações Bancárias: Métodos foram desenvolvidos para permitir consultas de saldo, realização de saques e depósitos. Cada operação de transação atualiza o saldo e adiciona um registro ao histórico, permitindo o rastreamento das atividades do usuário.

- Histórico de Transações: A classe também possui um vetor que armazena cada transação realizada, registrando dados essenciais e possibilitando a visualização completa das operações efetuadas.

- Formatação Monetária: Para atender aos padrões de exibição de valores financeiros, foi implementado um método que formata os valores monetários com separadores de milhar e duas casas decimais. Isso foi feito utilizando o manipulador ``std::fixed`` e ``std::setprecision`` para garantir que a visualização seja clara e adequada para o usuário final.

Cada método e função foi implementado de forma modular, possibilitando testes individuais de cada funcionalidade. A modularidade também facilita a expansão futura do sistema, permitindo a adição de novos recursos sem a necessidade de grandes modificações no código base.



### 3.4. - Testes e Validação

Após a implementação de cada funcionalidade, foram realizados testes unitários para garantir que cada parte do sistema funcionasse conforme esperado.

Por exemplo:

- Testes de Autenticação: Foram realizados testes para assegurar que a autenticação fosse bem-sucedida somente quando o número da conta e a senha estivessem corretos.

- Testes de Transações Bancárias: Testes foram executados para verificar a precisão do saldo após saques e depósitos, bem como a atualização do histórico de transações.

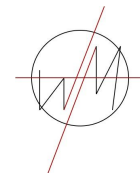
- Validação da Formatação Monetária: Valores foram testados para garantir a correta aplicação dos separadores de milhar e das duas casas decimais, conferindo que o sistema exibe as informações de forma precisa e legível.

### 3.5. - Descrição do Projeto Desenvolvido

O projeto resultante é um sistema de caixa eletrônico que permite ao usuário interagir com uma conta bancária simulada. A interface textual exibe um menu com opções para consulta de saldo, saque, depósito, exibição do histórico de transações e alteração de senha. A formatação adequada de valores e a apresentação clara de informações oferecem uma experiência de uso alinhada aos padrões bancários. Cada funcionalidade foi cuidadosamente desenvolvida para refletir as operações reais de um sistema de caixa eletrônico, assegurando que a aplicação sirva como uma ferramenta educacional eficiente no aprendizado de conceitos de programação e simulação de sistemas financeiros.

Por fim, a metodologia adotada garantiu uma implementação organizada, segura e modular, adequada para uso educacional, fornecendo uma aplicação prática de conceitos fundamentais de POO e C++. Esta abordagem permitiu que o projeto fosse executado de forma sistemática, resultando em um sistema que não apenas simula operações bancárias, mas também serve como um recurso didático valioso para futuros desenvolvedores.





## **4. - Resultados Obtidos**

O sistema de caixa eletrônico implementado neste projeto permitiu que as funcionalidades simuladas fossem executadas com sucesso e alinhadas aos objetivos iniciais, proporcionando uma experiência prática e educativa de alto valor. A seguir, são detalhadas as principais funcionalidades desenvolvidas e as capturas de tela correspondentes que ilustram o funcionamento de cada uma delas.

### **4.1. - Autenticação de Usuário**

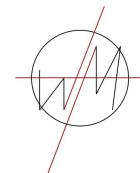
O sistema inclui uma camada de autenticação que exige que o usuário insira o número da conta e a senha. A funcionalidade de autenticação foi implementada para que apenas usuários com credenciais corretas possam acessar o menu de operações bancárias. Esse recurso simula a segurança básica esperada em um caixa eletrônico, e contribui para o entendimento do conceito de validação de acesso, que é essencial em sistemas de controle de contas bancárias.

#### **Captura de Tela: Autenticação**

Na tela de autenticação, o sistema solicita o número da conta e a senha. Em caso de erro, uma mensagem é exibida informando sobre a falha na autenticação e, caso bem-sucedida, o sistema direciona o usuário para o menu principal de operações.

### **4.2. - Consulta de Saldo**

Uma vez autenticado, o usuário pode consultar o saldo disponível na conta. Esta funcionalidade foi implementada com a exibição de valores monetários formatados, utilizando separadores de milhar e duas casas decimais, para garantir uma apresentação clara e precisa do saldo. Esse recurso fornece uma simulação



realista e detalhada de uma transação bancária comum, além de familiarizar o usuário com formatações padronizadas de valores financeiros.

## **Captura de Tela: Consulta de Saldo**

Na tela de consulta de saldo, o usuário visualiza o saldo disponível na conta com separadores de milhar, garantindo precisão e clareza na exibição dos valores.

### **4.3. - Realização de Saques e Depósitos**

O sistema permite ao usuário realizar saques e depósitos na conta. No caso de saques, foi implementada uma verificação que impede a retirada de valores superiores ao saldo disponível, simulando assim uma medida de controle real. Nos depósitos, o saldo é atualizado imediatamente após a operação, e o sistema exibe uma mensagem de confirmação para o usuário, destacando o saldo atualizado.

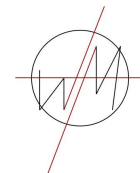
- Saques: Quando o usuário solicita um saque, o sistema verifica o saldo e, em caso de valores suficientes, atualiza o saldo e registra a transação no histórico. Caso o saldo seja insuficiente, uma mensagem é exibida, informando o usuário sobre a impossibilidade da operação.

- Depósitos: Para cada depósito realizado, o saldo da conta é atualizado, e uma confirmação é exibida, indicando o novo saldo com a formatação financeira adequada.

## **Captura de Tela: Saque e Depósito**

As telas de saque e depósito exibem, respectivamente, o valor solicitado, o saldo atualizado e mensagens de erro em caso de saldo insuficiente. Esse feedback contribui para a experiência intuitiva do usuário.

### **4.4. - Histórico de Transações**



Um dos principais recursos do sistema é o histórico de transações, que registra e exibe todas as operações realizadas pelo usuário, incluindo saques, depósitos, e alterações de senha. Esse histórico é armazenado em um vetor de strings, e cada operação é registrada com uma descrição detalhada e o valor da transação. Esta funcionalidade permite ao usuário rastrear suas atividades financeiras, sendo um componente essencial em sistemas bancários reais.

## **Captura de Tela: Histórico de Transações**

A tela de histórico exibe cada operação em uma lista detalhada, permitindo que o usuário acompanhe todas as transações realizadas desde o momento de criação da conta. Isso garante transparência e organização no controle de atividades financeiras.

### **4.5. - Alteração de Senha**

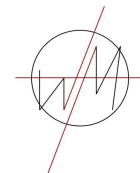
O sistema inclui uma funcionalidade de alteração de senha, onde o usuário pode atualizar sua senha após fornecer a senha atual. Esta funcionalidade foi implementada para demonstrar o conceito de segurança no controle de acesso, essencial em qualquer sistema bancário. A operação também é registrada no histórico, promovendo transparência nas alterações de credenciais.

## **Captura de Tela: Alteração de Senha**

Na tela de alteração de senha, o sistema solicita a senha atual e a nova senha desejada. Em caso de sucesso, uma confirmação é exibida, e o histórico de transações registra a alteração.

## **Análise Geral dos Resultados**

O projeto resultante oferece uma simulação detalhada e funcional de um caixa eletrônico, permitindo ao usuário explorar todas as operações bancárias



comuns de maneira prática e realista. A implementação das funcionalidades seguiu os princípios da Programação Orientada a Objetos, garantindo uma estrutura modular e expansível, além de reforçar práticas de segurança e controle de acesso em sistemas financeiros.

Além disso, o sistema foi amplamente testado e validado, garantindo que cada operação seja realizada com precisão e que o feedback visual fornecido ao usuário seja claro e informativo. As telas de operação simuladas refletem um fluxo de trabalho intuitivo e organizado, proporcionando uma experiência de usuário condizente com caixas eletrônicos reais.

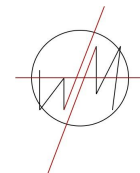
Esse projeto serve como um recurso educacional valioso para o aprendizado de conceitos fundamentais de programação e segurança bancária, oferecendo uma aplicação prática e consolidada para simulações de operações financeiras.

## **5. - Conclusão**

A implementação do sistema de caixa eletrônico virtual atendeu aos objetivos propostos, simulando de maneira funcional e realista as operações bancárias básicas. Durante o desenvolvimento, foi possível explorar e aplicar conceitos avançados de Programação Orientada a Objetos (POO) em C++, o que proporcionou uma estrutura modular e organizada ao código. Esse modelo facilitou a implementação e a expansão das funcionalidades, além de reforçar boas práticas de encapsulamento, polimorfismo e uso de construtores para inicialização de dados essenciais.

A simulação incluiu funcionalidades como consulta de saldo, saque, depósito, alteração de senha e histórico de transações. Cada uma dessas operações foi implementada com mecanismos de segurança e validação, replicando funcionalidades que se assemelham às de sistemas bancários reais. A autenticação por senha, por exemplo, foi fundamental para garantir que apenas usuários autenticados tivessem acesso às operações financeiras, refletindo um compromisso com o controle de acesso e a integridade das informações.

Um dos principais avanços observados no desenvolvimento foi a implementação do histórico de transações, que permitiu ao usuário acompanhar



cada operação realizada em sua conta. Esse recurso é essencial para promover transparência e controle, uma vez que possibilita ao usuário revisar suas atividades financeiras e identificar possíveis erros ou inconsistências. Essa funcionalidade não só agrega valor ao sistema, mas também ilustra um dos principais aspectos de um ambiente bancário seguro e confiável.

Os resultados demonstraram a eficácia da estrutura do código e a funcionalidade das operações, sendo possível realizar todos os testes com sucesso. No entanto, algumas melhorias poderiam ser aplicadas para tornar o sistema ainda mais robusto e aproximá-lo de um ambiente bancário real. Entre as sugestões para aprimoramento, destaca-se a implementação de autenticação em múltiplos níveis, como o uso de tokens ou autenticação biométrica, a fim de reforçar ainda mais a segurança do sistema. Além disso, o desenvolvimento de uma interface gráfica interativa poderia aumentar a usabilidade do sistema, tornando-o mais acessível e intuitivo para diversos perfis de usuários.

Por fim, o projeto destacou a importância da segurança e da organização no desenvolvimento de sistemas financeiros, permitindo uma experiência prática e realista em um ambiente simulado. A aplicação dos conhecimentos técnicos adquiridos, aliada ao uso de boas práticas de programação, contribuiu para um sistema funcional e educativo, que pode servir de base para estudos mais avançados em sistemas bancários e segurança digital.

## **6. - Diagrama de Classes**

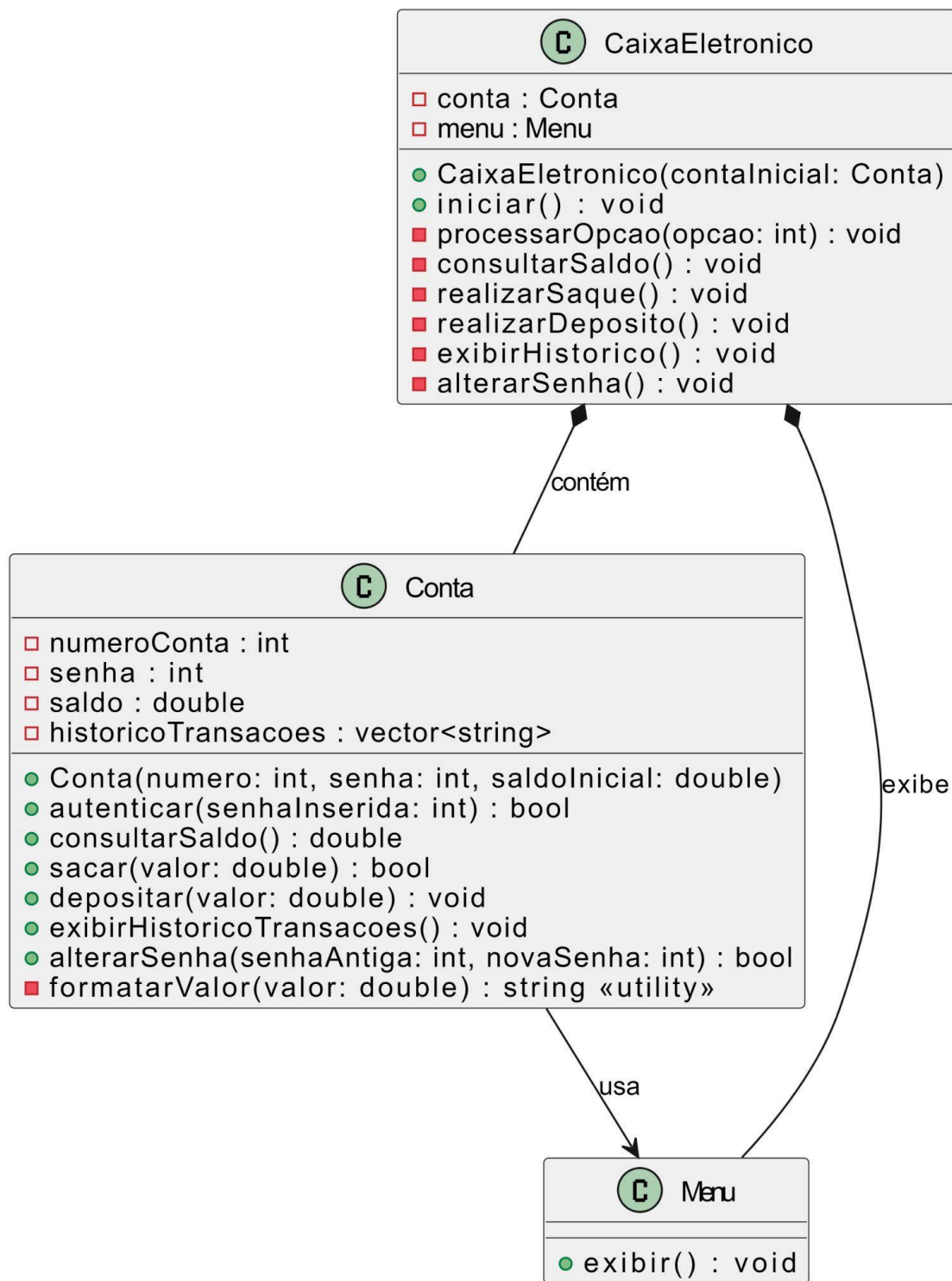
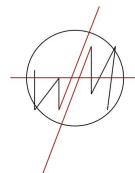
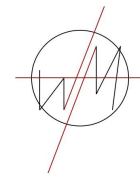


Imagem - Diagrama de Classe. Elaborado pelo autor.



## 7. - Referências Bibliográficas

ALENCAR, José. Introdução à programação orientada a objetos em C++. São Paulo: Editora Novatec, 2020. Acesso em: 31 de out. 2024.

BARROS, Carla; SOUZA, Felipe. Segurança em sistemas bancários: práticas e tecnologias. 2. ed. Rio de Janeiro: Alta Books, 2019. Acesso em: 31 de out. 2024.

DEITEL, H. M.; DEITEL, P. J. Como programar em C++. 5. ed. São Paulo: Pearson Prentice Hall, 2011. Acesso em: 31 de out. 2024.

MIRANDA, Laura. Desenvolvimento de sistemas seguros: teoria e prática. Porto Alegre: Bookman, 2018. Acesso em: 31 de out. 2024.

NORTON, Adam; COSTA, Roberto. Estrutura de dados e algoritmos em C++. Campinas: Editora Unicamp, 2022. Acesso em: 31 de out. 2024.

SILVA, João Pedro; NASCIMENTO, Ana Paula. Práticas de programação para iniciantes. Brasília: Editora do Brasil, 2021. Acesso em: 31 de out. 2024.