Sistema de Loja Online

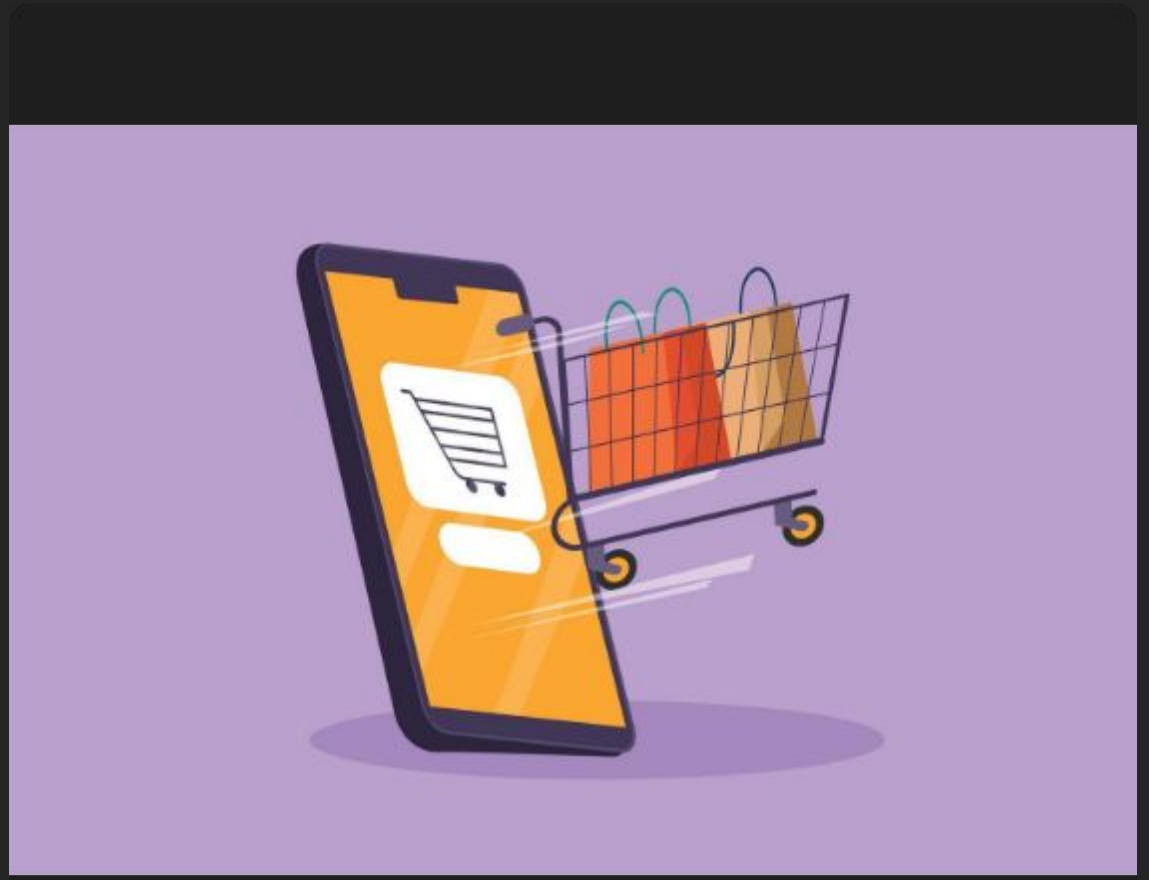
Desenvolvimento Orientado a Objetos com C++17 e API REST

Ana Julia Damasceno
Bruno Simoes Amorim
Helio Emanuel Pereira
Henrique Gabriel Fonseca
Miguel Angel Roa
Leo Felipe dos Santos

1. Introdução

Este projeto consiste no desenvolvimento de um **Sistema de Loja Online** completo, utilizando a linguagem C++ e seguindo rigorosamente os princípios da Programação Orientada a Objetos (POO).

- Simulação de funcionalidades reais de e-commerce.
- Sistema modular e escalável.
- Integração externa via **API REST** (cpp-httplib).
- Modelagem de dados robusta baseada em UML.



2. Objetivos do Projeto

- **Construção Sólida:** Implementar um sistema funcional respeitando os pilares da POO.
- **Modelagem:** Representar o domínio do problema através de diagramas UML claros.
- **Modularidade:** Criar classes reutilizáveis e bem estruturadas (Baixo acoplamento).
- **Integração:** Implementar API REST para operações remotas (Cadastro, Consultas).

3. Stack Tecnológica

C++17

Linguagem core do sistema, oferecendo alta performance e recursos modernos de memória.

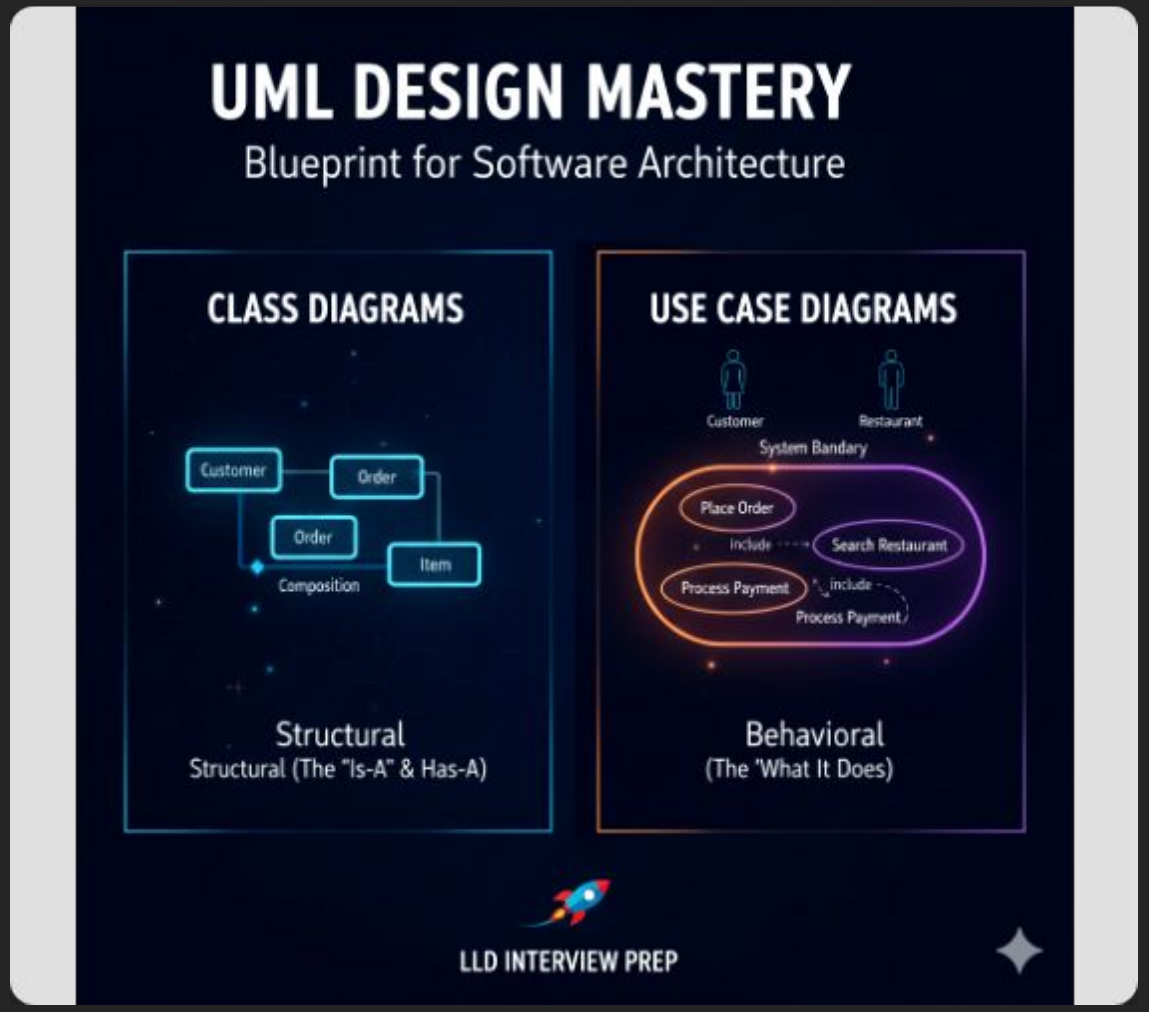
cpp-httplib

Biblioteca header-only para implementação leve e rápida do servidor API REST.

JSON & PlantUML

nlohmann::json para manipulação de dados e PlantUML para modelagem visual.

4. Estrutura do Sistema (UML)



Classes Principais

- **Produto:** Entidade base (ID, Nome, Preço, Estoque).
- **Carrinho:** Agregação de produtos selecionados pelo cliente.
- **Cliente:** Usuário que interage com o sistema e possui um carrinho.
- **Loja:** Controlador central (Manager) que gerencia estoque e catálogo.

5.1 Desenvolvimento: Classe Produto

A classe Produto utiliza encapsulamento para proteger os dados. Atributos privados são acessados apenas via métodos públicos.

Este design permite validação de dados (ex: preço não pode ser negativo) antes da atribuição.

```
class Produto
private
    int
    std::string
    double
    int

public
    Produto int      string        double


    // Getters e Setters
    double getPreco    const    return
    void atualizarEstoque  int
```

5.2 Lógica do Carrinho

```
void Carrinho   adicionarItem  Produto     int
    if    getEstoque
              push_back
                   getPreco
          atualizarEstoque
        std::cout    "Item adicionado!"
      else
        std::cout    "Estoque insuficiente."
```

Gerenciamento de Estado

O Carrinho interage diretamente com objetos Produto.

- Verificação de disponibilidade em tempo real.
- Atualização dinâmica do valor total.
- Utilização de std::vector para armazenar itens dinamicamente.

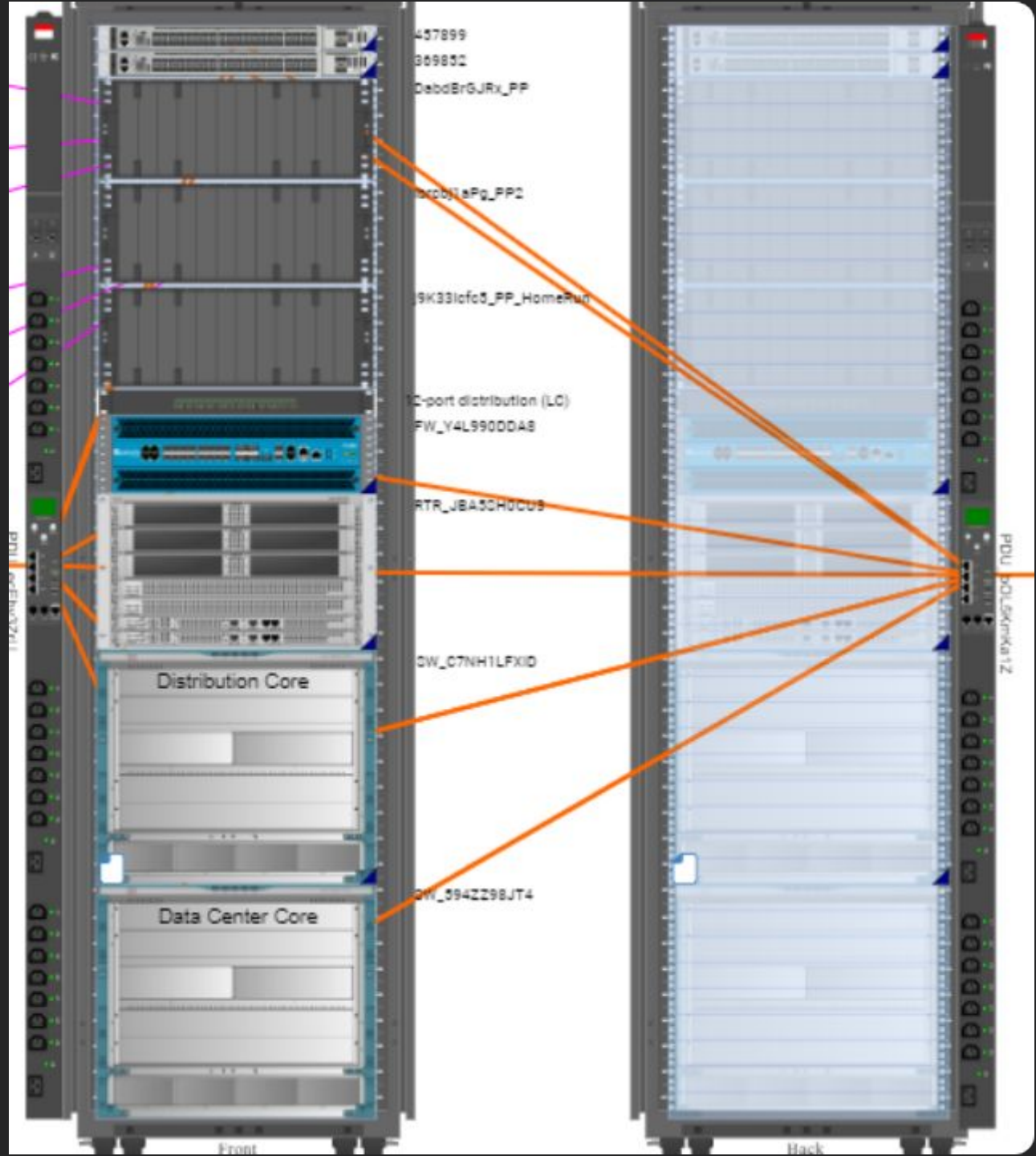
5.3 Integração API REST

Utilizando httplib, expomos endpoints para comunicação externa.

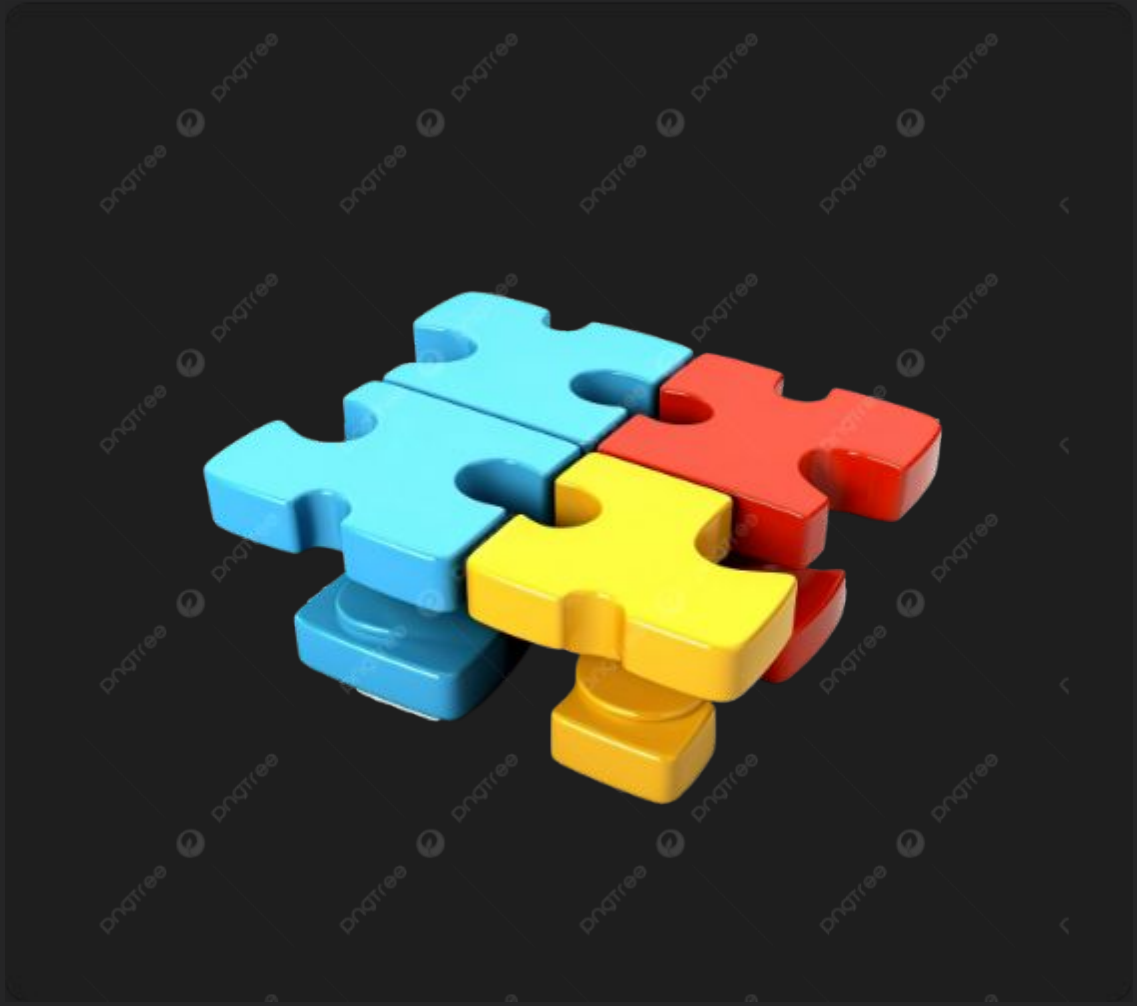
```
httplib::Server

// Endpoint GET para listar produtos
    Get "/produtos"      const Request        Response
    json           toJSON
        set_content    dump    "application/json"


// Inicia o servidor na porta 8080
    listen "localhost"
```



6. Conceitos de POO Aplicados



- **Encapsulamento:** Proteção de atributos sensíveis (preço, estoque) dentro das classes.
- **Abstração:** Simplificação de entidades complexas do mundo real (Loja, Cliente) em modelos de software.
- **Herança:** Estrutura preparada para expansão (Ex: ProdutoDigital herdando de Produto).
- **Polimorfismo:** Flexibilidade para métodos (ex: calcularFrete()) comportarem-se diferentemente em subclasses.

7. Fluxo de Execução

1. Cliente

Envia requisição POST para /carrinho selecionando produtos.

2. API Controller

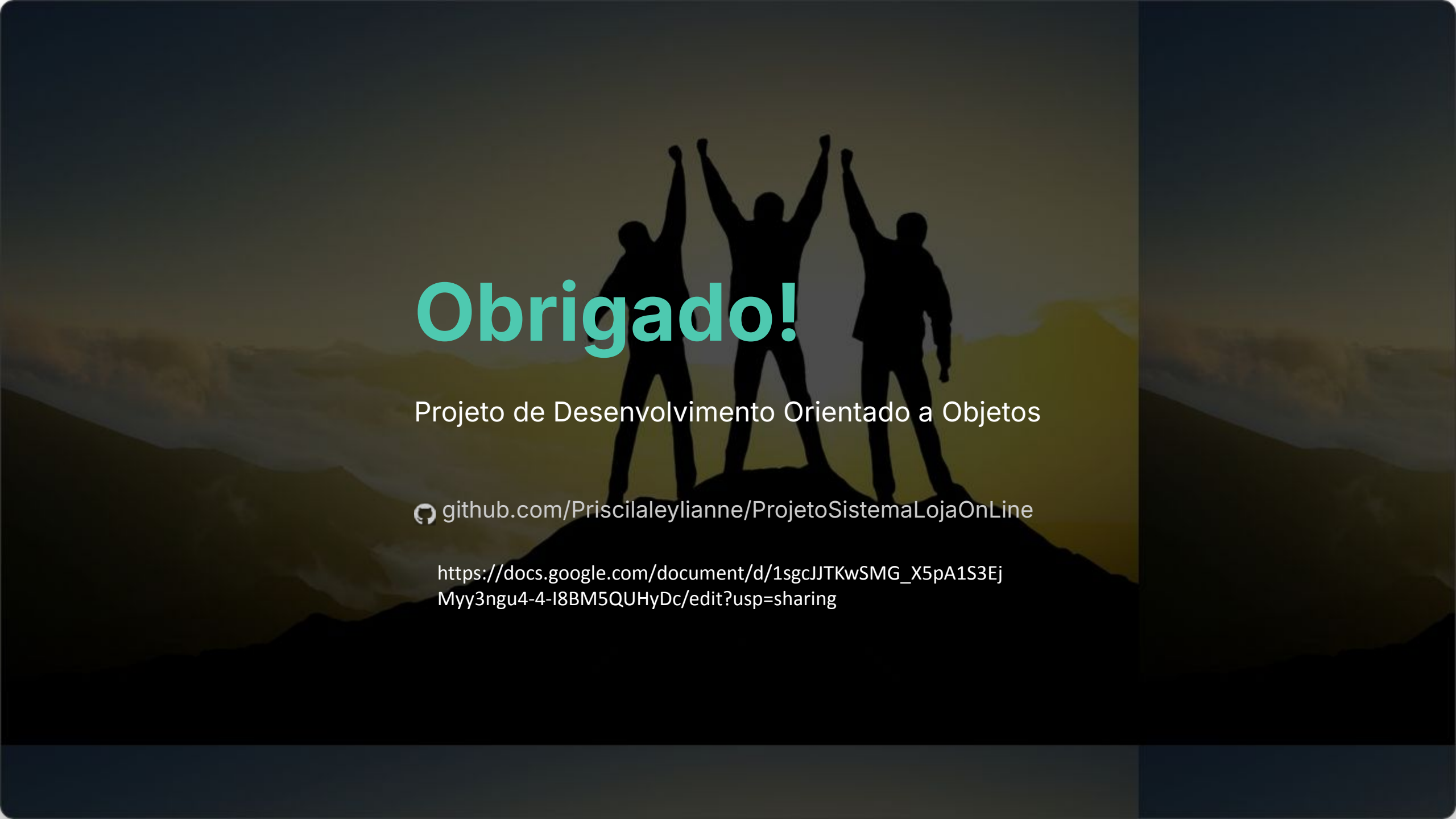
Recebe JSON, valida dados e chama métodos da classe Loja.

3. Estoque

Sistema verifica disponibilidade, atualiza quantidade e confirma compra.

8. Referências Bibliográficas

- STROUSTRUP, Bjarne. *The C++ Programming Language*. 4th Edition. Addison-Wesley, 2013.
- CPP-HTTPLIB. *A C++11 single-file header-only cross platform HTTP/HTTPS library*. Disponível em GitHub.
- JSON FOR MODERN C++. *Niels Lohmann*. Documentação oficial da biblioteca.
- GAMMA, Erich et al. *Design Patterns: Elements of Reusable Object-Oriented Software*.



Obrigado!

Projeto de Desenvolvimento Orientado a Objetos

github.com/Priscilaleylianne/ProjetoSistemaLojaOnLine

https://docs.google.com/document/d/1sgcJJTKwSMG_X5pA1S3EjMyy3ngu4-4-I8BM5QUHyDc/edit?usp=sharing

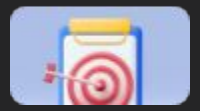
Image Sources

https://img.freepik.com/premium-photo/abstract-blue-technology-background-with-polygonal-network-connection-dots-lines-dark-blue_1108670-5764.jpg

Source: www.freepik.com

https://static.vecteezy.com/system/resources/previews/016/998/454/non_2x/cartoon-flat-style-drawing-shopping-cart-out-of-smartphone-sale-digital-lifestyle-and-consumerism-concept-e-commerce-digital-marketing-online-store-technology-graphic-design-illustration-vector.jpg

Source: www.vecteezy.com



https://static.vecteezy.com/system/resources/previews/035/858/560/non_2x/3d-paper-clipboard-and-target-with-arrow-render-paper-sheet-dartboard-and-arrow-work-project-plan-productivity-checklist-goal-task-management-to-do-list-illustration-vector.jpg

Source: www.vecteezy.com

https://miro.medium.com/1*r24UX6yiFTH1h54UDsAPyQ.png

Source: blog.stackademic.com

https://graphicalnetworks.com/wp-content/uploads/2023/08/rack-cabinet-diagram-example.png

Source: graphicalnetworks.com

https://png.pngtree.com/png-clipart/20240314/original/pngtree-d-jigsaw-puzzle-pieces-isolated-problem-solving-business-connecting-cooperation-partnership-png-image_14581852.png

Source: pngtree.com

Image Sources

https://miro.medium.com/1*tFUxiPEPqTNtAW96oQwIqw.jpeg

Source: medium.com