

Relatório: Loja Virtual

Thiago Freire

Dezembro 2024

Introdução

O projeto **Loja Virtual** simula um sistema básico de compras, permitindo a gestão de produtos, pedidos e um catálogo. Ele foi desenvolvido com foco em modularidade e testabilidade, utilizando boas práticas de programação para facilitar manutenção e expansão futura.

Escolhas Técnicas

As principais escolhas técnicas feitas no desenvolvimento foram motivadas por critérios de desempenho, escalabilidade e facilidade de manutenção:

- **C++ com Padrões Modernos:** A escolha por C++ foi motivada pela necessidade de controle de baixo nível e alto desempenho, essenciais para sistemas de gerenciamento e processamento de dados. Foram utilizados padrões modernos para aproveitar recursos como `smart pointers` e `std::vector`.
- **CMake:** O CMake foi selecionado como ferramenta de build por facilitar a integração com diversas plataformas. Sua flexibilidade permite configurar dependências externas, como o GoogleTest e o `spdlog`, além de simplificar o gerenciamento de projetos complexos.
- **GoogleTest:** Este framework foi escolhido criação de testes unitários e de integração.
- **spdlog:** O `spdlog` foi adotado para registrar logs do sistema, incluindo mensagens de informação e erros. Sua eficiência em tempo de execução e suporte para rotação de arquivos o tornam ideal para sistemas que precisam de rastreamento contínuo de eventos.
- **Estrutura Modular:** A divisão do sistema em classes independentes, como `Produto`, `Catalogo` e `Pedido`, seguiu o princípio da responsabilidade única, facilitando a manutenção e o teste isolado de componentes. Essa modularidade também permite escalabilidade futura, possibilitando a adição de novas funcionalidades sem grandes alterações no código existente.

Funcionamento Geral do Sistema

O sistema é dividido em três componentes principais:

- **Produto:** Representa os itens disponíveis para compra, contendo informações como ID, nome e preço.
- **Catalogo:** Gerencia uma lista de produtos disponíveis. Permite adicionar produtos e buscar por ID.
- **Pedido:** Controla os produtos comprados por um cliente e calcula o valor total do pedido.

Essas classes interagem entre si para simular o funcionamento de uma loja virtual, dividido em três etapas:

Etapas 1: Os produtos são criados e adicionados ao catálogo.

Etapas 2: Um cliente seleciona produtos do catálogo e adiciona ao pedido.

Etapas 3: O sistema calcula o valor total do pedido e exibe o resultado.

Testes e Depuração

Para garantir a qualidade do código, foram implementados testes automatizados com o GoogleTest:

- **Testes Unitários:** Validam funcionalidades individuais, como a adição de produtos ao catálogo e o cálculo do total do pedido.
- **Testes de Integração:** Simulam o uso completo do sistema, verificando a interação entre as classes.

Durante a execução, eventos importantes são registrados via `spdlog`. Exemplo de mensagens de log:

```
[INFO] Iniciando o sistema Loja Virtual...  
[INFO] Produto 'Notebook' adicionado ao catálogo.  
[ERROR] Produto com ID 5 não encontrado.
```

Conclusão

O uso de testes automatizados e logs contribuiu para identificar e corrigir erros rapidamente, além de facilitar a manutenção futura.