

Desenvolvimento de um Sistema de E-commerce Escalável e Seguro

Introdução

O desenvolvimento deste projeto envolve a criação de um e-commerce escalável e seguro, aplicando conhecimentos adquiridos nas disciplinas do semestre vigente. O estudo de caso aborda a ShopOnline, uma varejista digital que enfrenta desafios operacionais devido ao rápido crescimento de sua base de clientes e ao aumento do volume de transações, especialmente durante eventos promocionais. Problemas como lentidão no sistema, inconsistências no estoque e dificuldades em gerenciar picos de acesso comprometem a experiência do usuário e o crescimento da empresa.

Para resolver essas questões, a TechCommerce foi contratada para desenvolver um sistema modular e escalável, integrando desde cadastro de usuários até gerenciamento logístico de pedidos, com suporte a parceiros externos. O sistema deve atender a requisitos rigorosos de desempenho, usabilidade e segurança, garantindo alta disponibilidade sob demanda elevada. Seu banco de dados precisa assegurar consistência em operações críticas (estoque, pagamentos e buscas), enquanto a arquitetura permitirá escalabilidade horizontal e uso de infraestrutura ágil. Adicionalmente, a solução será desenvolvida com design orientado a objetos para promover modularidade e facilitar manutenções futuras.

A implementação visa melhorar a experiência do cliente, aumentar a eficiência operacional e suportar o crescimento contínuo da ShopOnline.

Desenvolvimento

O projeto será estruturado por seções, sendo elas Análise Orientada a Objetos, Modelagem de Dados, Programação e Desenvolvimento de Banco de Dados e Infraestrutura Ágil. Matérias pertinentes ao curso de DevOps.

Análise Orientada a Objetos

Para esse projeto será considerado o desenvolvimento de um diagrama de classes para o sistema de e-commerce da ShopOnline, contemplando as seguintes funcionalidades:

Gerenciamento de Clientes

- Cadastro de clientes com dados pessoais (nome, e-mail, senha).
- Login seguro via e-mail e senha.
- Atualização e exclusão de contas.

Gerenciamento de Produtos

- Cadastro de produtos com descrição, preço e outras especificações.
- Busca de produtos para adição ao carrinho.

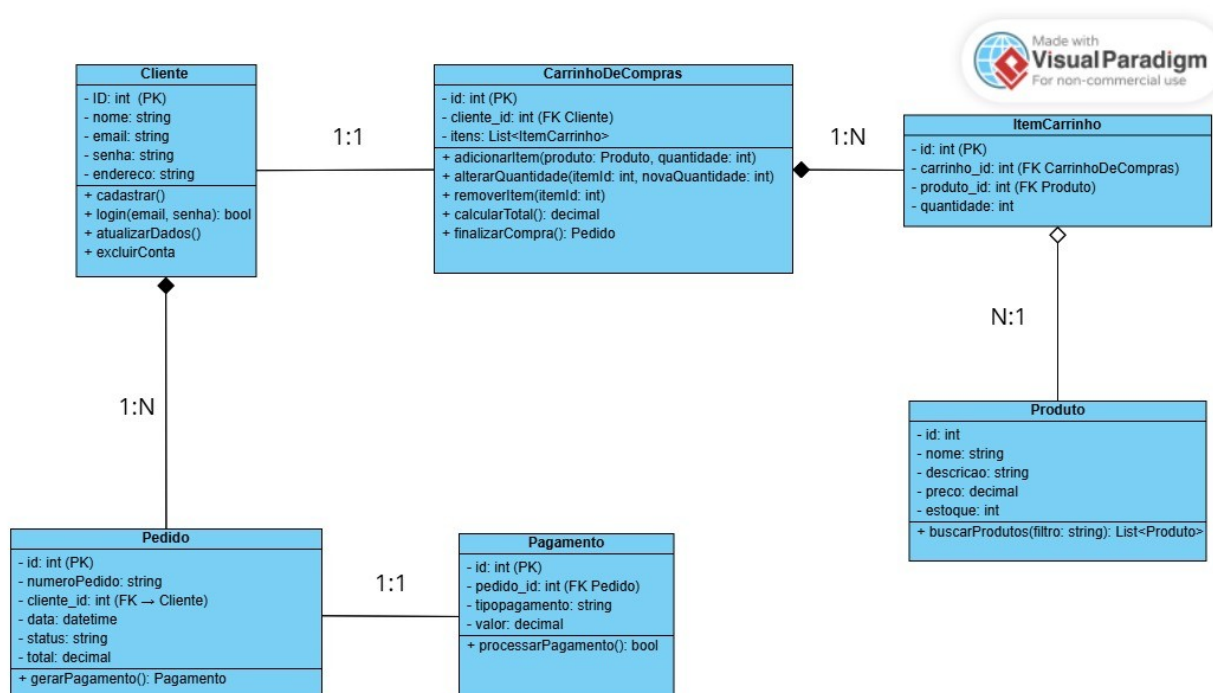
Carrinho de Compras

- Adição, remoção e ajuste de quantidades de produtos.
- Finalização da compra.

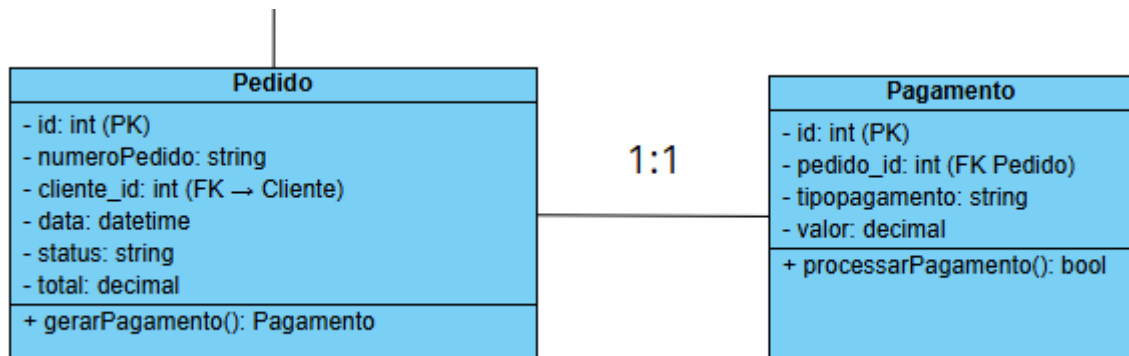
Pedidos e Pagamentos

- Cada pedido estará vinculado a um processo de pagamento.

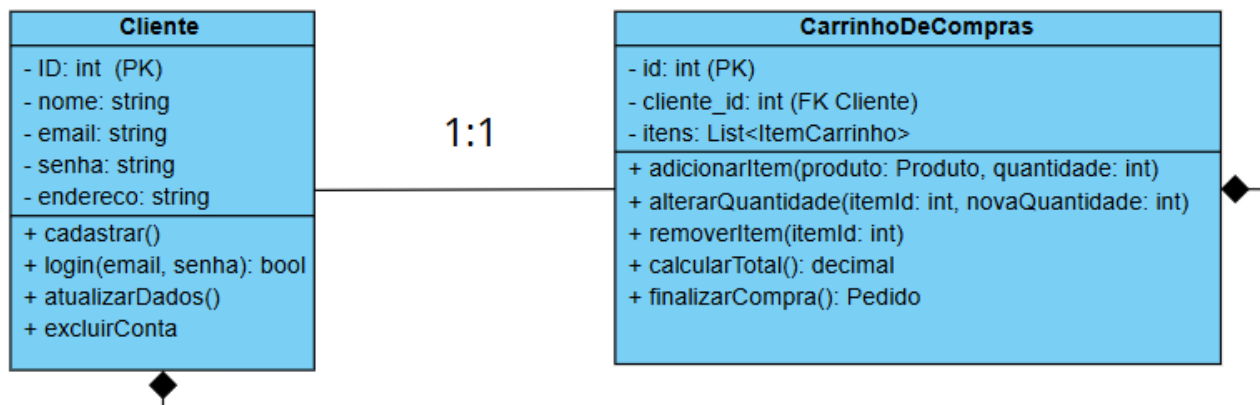
Será utilizado o visual paradigm para a elaboração do diagrama de classes. Por se tratar de uma ferramenta versátil e de fácil uso para essa etapa do projeto. A imagem a seguir demonstrar como ficou o diagrama:



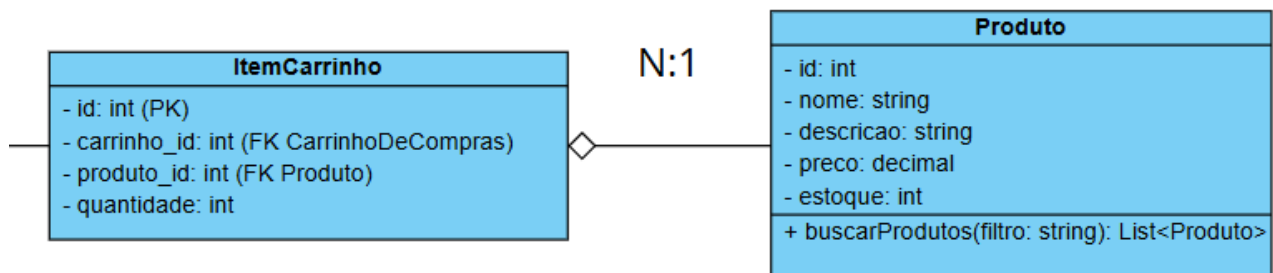
Cada classe tem seu próprio nome, método e relacionamentos. Por exemplo as classes “Pedido” e “Pagamento”, possuem relacionamento de um para um (1:1), devido a cada pedido gerar um pagamento. Observe abaixo:



Outras classes que possui o mesmo relacionamento de 1:1 é “Cliente” e “CarrinhoDeCompra”. Veja a seguir:



As classes “ItemCarrinho” e “Produto” possuem um relacionamento de N para 1 (N:1), por que cada item referencia um produto. Observe abaixo:



As classes que possuem relacionamentos de um para N (1:N) são “Cliente” e “Pedido”, já que um cliente pode ter vários pedidos. E “CarrinhoDeCompra” e “ItemCarrinho” devido a um carrinho contém múltiplos itens.

Modelagem de Dados

Para a modelagem de dados deste projeto, optou-se pelo MySQL Workbench como ferramenta principal, devido à sua ampla adoção no mercado, interface intuitiva e recursos robustos para projetos de bancos de dados relacionais. Sua usabilidade simplifica a criação de diagramas ER (Entidade-Relacionamento), a definição de esquemas e a implementação de consultas SQL, alinhando-se perfeitamente às necessidades técnicas do sistema da ShopOnline.

A modelagem de dados do projeto requer atenção especial a aspectos críticos, como:

- Estruturação das tabelas fundamentais:

Usuários, Produtos, Pedidos, Pagamentos e ItensPedido.

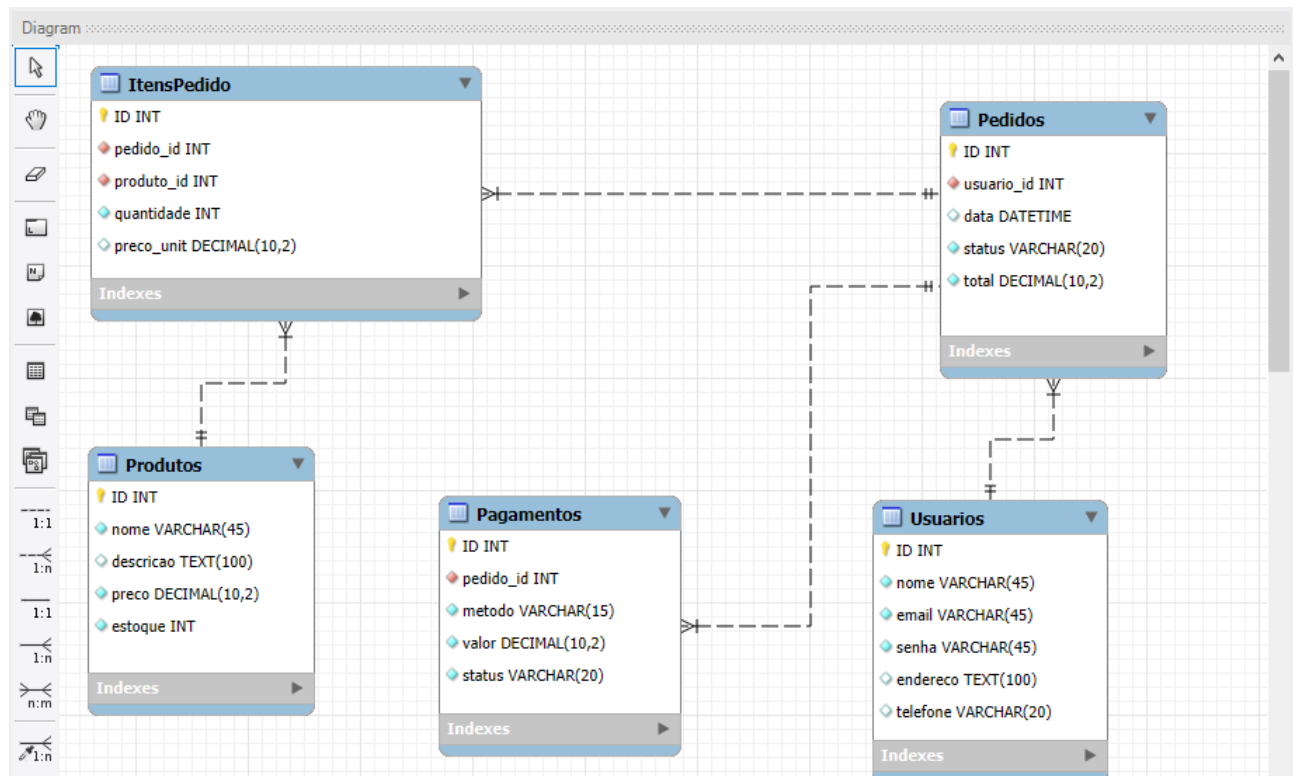
- Relacionamentos entre entidades:

A conexão entre Pedidos e Produtos deve ser implementada por meio da tabela ItensPedido, garantindo a rastreabilidade dos itens vendidos.

- Atributos essenciais na tabela de relacionamento:

Campos como quantidade e preço unitário são indispensáveis em ItensPedido para manter a integridade financeira e logística.

A seguir podemos observar com ficou a estruturar da modelagem de dados e suas ligações.



Esses elementos asseguram que o banco de dados atenda às necessidades do sistema de e-commerce, como escalabilidade e consistência nas operações.

Relacionamentos

A tabela a seguir classifica e descreve os relacionamentos entre as entidades do banco de dados, definindo a cardinalidade e as regras de associação.

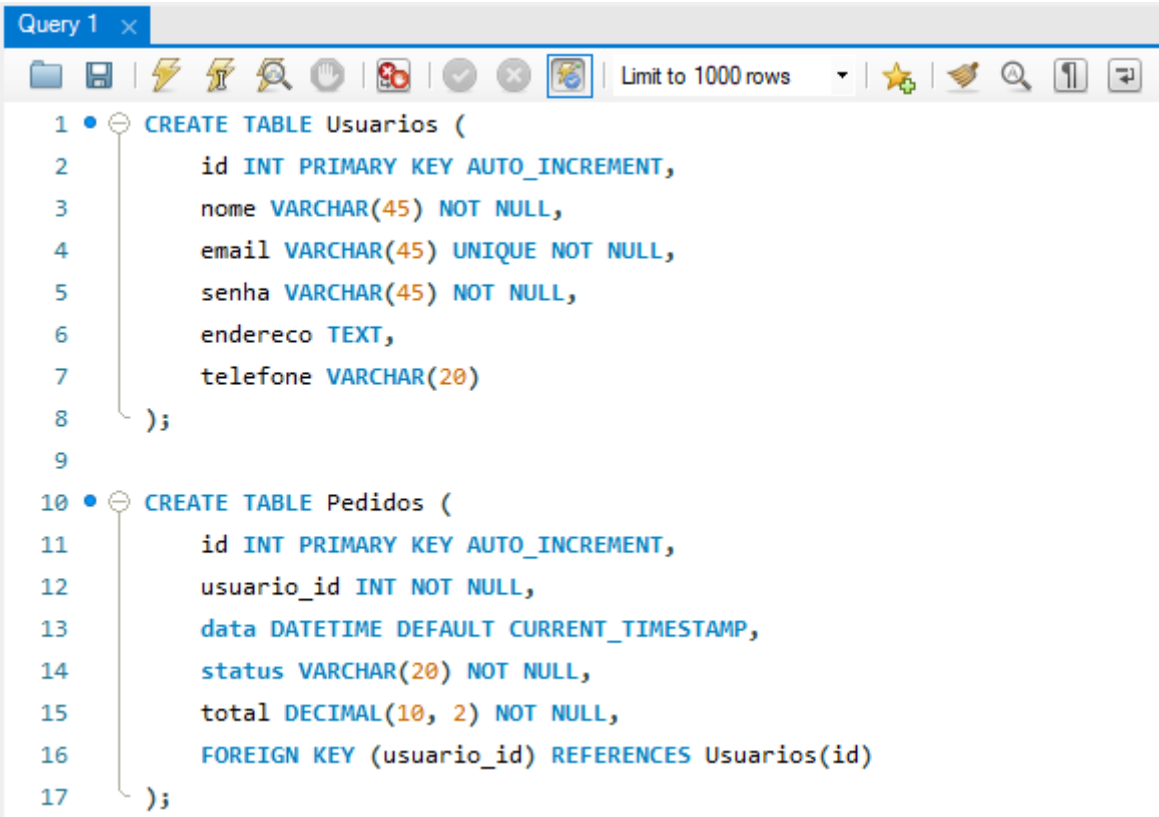
Tabelas	Tipo	Descrição
Usuários → Pedidos	1:N	Um usuário pode ter vários pedidos.
Pedidos → ItensPedido	1:N	Um pedido pode conter múltiplos itens.
Produtos → ItensPedido	1:N	Um produto pode aparecer em vários itens de pedidos.
Pedidos → Pagamentos	1:1	Cada pedido tem exatamente um pagamento.

Programação e Desenvolvimento de Banco de Dados

Para a implementação do banco de dados do sistema de e-commerce, optou-se pela utilização do SGBD MySQL. Serão desenvolvidos scripts SQL completos que demonstrarão não apenas o domínio técnico da linguagem, mas também sua aplicação prática no contexto do projeto, garantindo a funcionalidade e eficiência do sistema.

Entre os requisitos essenciais desta atividade, destaca-se a implementação das tabelas do banco de dados conforme modelagem proposta, assegurando que a tabela intermediária ItensPedido contenha os atributos fundamentais: ID, pedido_id, produto_id, quantidade e preco_unit, garantindo assim a integridade e rastreabilidade dos dados nas relações entre pedidos e produtos.

A imagem a seguir ilustra a criação das tabelas Usuários e Pedidos, elementos essenciais para a estruturação e funcionamento do sistema proposto.



```
Query 1 x
Limit to 1000 rows

1 • CREATE TABLE Usuarios (
2     id INT PRIMARY KEY AUTO_INCREMENT,
3     nome VARCHAR(45) NOT NULL,
4     email VARCHAR(45) UNIQUE NOT NULL,
5     senha VARCHAR(45) NOT NULL,
6     endereco TEXT,
7     telefone VARCHAR(20)
8 );
9
10 • CREATE TABLE Pedidos (
11     id INT PRIMARY KEY AUTO_INCREMENT,
12     usuario_id INT NOT NULL,
13     data DATETIME DEFAULT CURRENT_TIMESTAMP,
14     status VARCHAR(20) NOT NULL,
15     total DECIMAL(10, 2) NOT NULL,
16     FOREIGN KEY (usuario_id) REFERENCES Usuarios(id)
17 );
```

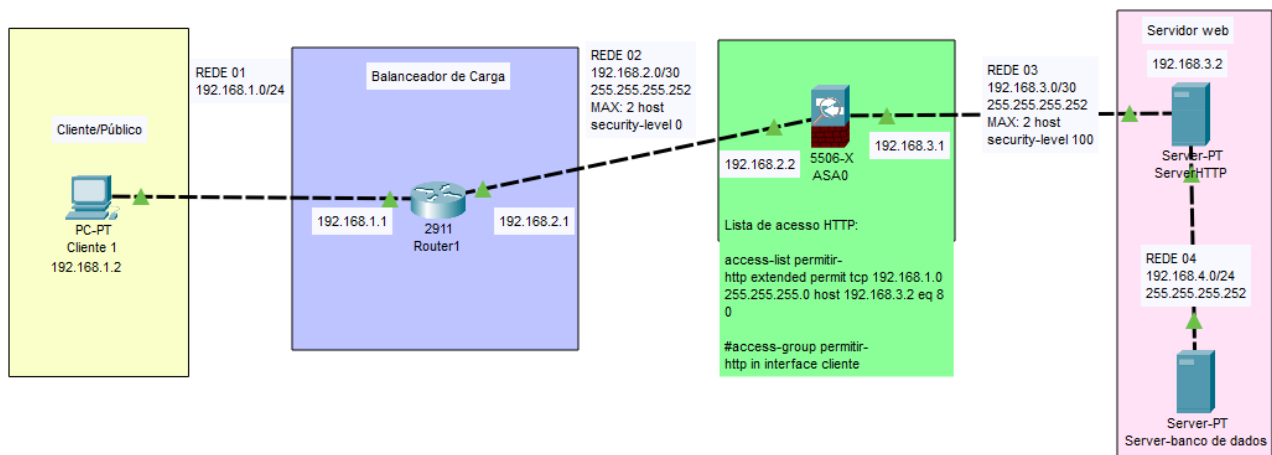
A imagem abaixo apresenta a modelagem das tabelas Pagamentos e Produtos, componentes fundamentais para a arquitetura e operação do sistema, garantindo a gestão eficiente de transações financeiras e o controle de itens comercializados.

```
28
29 • ○ CREATE TABLE Pagamentos (
30     id INT PRIMARY KEY AUTO_INCREMENT,
31     pedido_id INT UNIQUE NOT NULL,
32     metodo VARCHAR(15) NOT NULL,      -- Pagamento
33     valor DECIMAL(10, 2) NOT NULL,
34     status VARCHAR(20) NOT NULL,
35     FOREIGN KEY (pedido_id) REFERENCES Pedidos(id)
36 );
37
38 • ○ CREATE TABLE Produtos (
39     id INT PRIMARY KEY AUTO_INCREMENT,
40     nome VARCHAR(45) NOT NULL,
41     descricao TEXT,
42     preco DECIMAL(10, 2) NOT NULL,
43     estoque INT NOT NULL
44 );
45
```

Infraestrutura Ágil

Este trabalho propõe um plano estratégico para implementar uma infraestrutura ágil na ShopOnline, alinhada aos princípios de escalabilidade, modularidade e segurança, visando melhorar a eficiência operacional e a experiência do usuário. O objetivo central é superar os desafios atuais da empresa, como a demanda em picos de acesso, por meio de uma arquitetura escalável baseada em nuvem ou modelo híbrido, com estratégias de escalabilidade horizontal, balanceamento de carga e redundância. A modularidade do sistema será detalhada por componente (ex.: autenticação, banco de dados, integrações), definindo responsabilidades, alta disponibilidade e confiabilidade. Além disso, serão apresentadas medidas de segurança robustas para proteger dados e transações, incluindo defesas contra ataques cibernéticos e conformidade com regulamentações de proteção de dados.

Para a implementação do e-commerce da ShopOnline, utilizaremos o Cisco Packet Tracer para simular a infraestrutura de rede. A topologia adotada no projeto segue a estrutura: cliente (REDE 01) → roteador (atuando como balanceador) → firewall (ASA 5506-X) → servidor web (REDE 03), conforme ilustrado na imagem abaixo. O tráfego HTTP é permitido do cliente para o servidor por meio de uma lista de acesso no firewall.



A partir do desenvolvimento desse modelo básico, foram identificados pontos essenciais para alcançar o objetivo final do projeto. A infraestrutura recomendada consiste em um modelo de nuvem híbrida, utilizando nuvem pública (AWS/Azure/GCP) para o front-end, balanceamento de carga e CDN, garantindo alta escalabilidade e desempenho, enquanto ambientes privados (On-Premises ou Nuvem Privada) serão responsáveis por hospedar bancos de dados críticos, como informações de clientes e transações financeiras, assegurando maior controle e segurança dos dados sensíveis. Essa abordagem estratégica permite combinar a flexibilidade e a escalabilidade da nuvem pública com a proteção e a governança oferecidas pelo ambiente privado, atendendo assim às demandas técnicas e regulatórias do projeto.

Para garantir uma escalabilidade eficiente, o sistema adotará um modelo horizontal, utilizando containers Docker gerenciados pelo Kubernetes para os servidores web e de aplicação, permitindo escalonamento dinâmico durante picos de demanda. Além disso, o banco de dados implementará a estratégia de Read Replicas para otimizar o desempenho em operações de consulta, assegurando maior disponibilidade e distribuição equilibrada da carga. Essa abordagem combinada proporciona uma infraestrutura flexível e robusta, capaz de se adaptar às variações de tráfego sem comprometer a performance do sistema.

Modularidade do Sistema

Módulo	Responsabilidades	Alta Disponibilidade
Autenticação	Login, OAuth 2.0, JWT, MFA	Redis para sessões + DDoS
Banco de Dados	Transações, pedidos, estoque	Replicação síncrona + Backups
Pagamentos	Integração com gateways	Processamento assíncrono.
Catálogo	Busca, filtros, recomendações	Elasticsearch + CDN

Logística	Rastreamento de pedidos	Filas SQS para retentativas
-----------	-------------------------	-----------------------------

Para garantir a segurança das transações, é essencial implementar medidas robustas, como um Web Application Firewall (WAF), que protege contra ataques comuns, incluindo SQL Injection e Cross-Site Scripting (XSS) – soluções como o AWS WAF são amplamente utilizadas. Além disso, a proteção contra DDoS, oferecida por serviços como Cloudflare ou AWS Shield, combinada com um monitoramento contínuo, assegura a disponibilidade e a integridade dos sistemas, mitigando riscos e mantendo a operação estável.

Conclusão

O projeto de e-commerce para a ShopOnline integrou arquitetura escalável, segurança e modularidade para superar desafios como picos de acesso e proteção de dados. A modelagem em MySQL e a estrutura orientada a objetos garantiram consistência nas transações e estoque, enquanto a infraestrutura híbrida (nuvem + on-premises) equilibrou desempenho e controle de dados sensíveis. Tecnologias como Kubernetes, Redis e WAF asseguraram alta disponibilidade e defesa contra ataques. A simulação inicial no Packet Tracer validou a topologia básica, mas a solução evoluiu para incluir redundância e auto-scaling. O resultado é um sistema pronto para crescimento, com experiência otimizada para o cliente e conformidade regulatória, consolidando a ShopOnline como uma plataforma robusta e adaptável no mercado digital.