

AZ SHOP

Documento de Arquitetura e Requisitos do Sistema

Idealizado e Desenvolvido por: Daniel Azevedo de Oliveira Maia

23 de fevereiro de 2026

Conteúdo

1	Visão Geral do Projeto	2
1.1	Descrição do Produto	2
1.2	Objetivo Estratégico	2
2	Escopo do MVP	4
2.1	Funcionalidades Incluídas	4
2.2	Funcionalidades Fora do Escopo Inicial	4
3	Arquitetura do Sistema	5
3.1	Modelo Arquitetural	5
3.2	Componentes Principais	5
4	Modelo de Domínio Inicial	6
4.1	Entidades Principais	6
5	Estratégia de Persistência	7
5.1	Banco de Dados	7
6	Estratégia de Evolução	8
7	Conclusão	9

1 Visão Geral do Projeto

1.1 Descrição do Produto

O AZ Shop é um sistema de e-commerce desenvolvido com o objetivo principal de servir como base prática para estudo aprofundado de arquitetura de software, modelagem de domínio, persistência relacional, segurança e boas práticas em aplicações backend.

O sistema permitirá:

- Cadastro e autenticação de usuários;
- Diferenciação de perfis (Administrador e Cliente);
- Cadastro e gerenciamento de produtos e categorias;
- Controle de estoque;
- Carrinho de compras;
- Criação e gerenciamento de pedidos;
- Processamento de pagamento simulado.

O foco inicial será exclusivamente o backend, implementado em Java utilizando Spring Boot.

1.2 Objetivo Estratégico

O objetivo central do AZ Shop é proporcionar um ambiente completo para prática dos seguintes conceitos técnicos:

- Modelagem de entidades e relacionamentos (1:1, 1:N, N:1);
- Uso de chaves primárias, estrangeiras e restrições;
- Persistência com JPA e SQL;
- Camadas bem definidas (Controller, Service, Repository);

- Paginação, ordenação e filtros;
- Serialização e desserialização JSON;
- MapStruct para mapeamento DTO;
- Controle transacional;
- Autenticação e autorização com JWT;
- Arquitetura monolítica modular escalável.

2 Escopo do MVP

2.1 Funcionalidades Incluídas

- Cadastro e autenticação de usuários com JWT;
- Perfis distintos (ADMIN e CUSTOMER);
- Cadastro de categorias e produtos (ADMIN);
- Listagem de produtos com paginação e filtros;
- Controle de estoque;
- Criação e manutenção de carrinho de compras;
- Conversão do carrinho em pedido;
- Registro de itens do pedido;
- Processamento de pagamento simulado;
- Controle de status do pedido.

2.2 Funcionalidades Fora do Escopo Inicial

- Integração com gateway real de pagamento;
- Sistema de avaliações de produtos;
- Marketplace com múltiplos vendedores;
- Sistema de cupons e promoções avançadas;
- Integração com sistema de frete externo.

3 Arquitetura do Sistema

3.1 Modelo Arquitetural

A arquitetura adotada será:

Monólito modular baseado em camadas, utilizando Spring Boot.

Motivos:

- Facilidade de aprendizado e manutenção;
- Clareza estrutural para estudo de domínio;
- Separação lógica por módulos;
- Evolução futura para microsserviços.

3.2 Componentes Principais

Componente	Descrição
Backend	Aplicação Java com Spring Boot responsável por toda a lógica de negócio.
Banco de Dados	PostgreSQL para armazenamento relacional estruturado.
Autenticação	Spring Security com JWT stateless.
Mapeamento	MapStruct para conversão entre entidades e DTOs.

4 Modelo de Domínio Inicial

4.1 Entidades Principais

- User
- Address
- Category
- Product
- Inventory
- Cart
- CartItem
- Order
- OrderItem
- Payment

Essas entidades permitirão a prática completa de relacionamentos e restrições de integridade.

5 Estratégia de Persistência

5.1 Banco de Dados

O sistema utilizará PostgreSQL com modelagem relacional estruturada.

Serão praticados:

- Chaves primárias auto geradas;
- Chaves estrangeiras;
- Unique constraints;
- Índices;
- Controle de versão para concorrência otimista;
- Transações.

6 Estratégia de Evolução

Após consolidação do monólito modular, o sistema poderá evoluir para:

- Separação por microsserviços;
- Comunicação entre serviços;
- Mensageria;
- Estratégias de consistência distribuída.

7 Conclusão

O AZ Shop é um projeto educacional estruturado para prática avançada de arquitetura backend moderna.

A arquitetura definida:

- Permite aprendizado profundo;
- É modular e escalável;
- Possui fundamentos sólidos;
- Pode evoluir para arquiteturas distribuídas.

Este documento serve como referência oficial inicial para o desenvolvimento do sistema.