

# Documentação Microsserviços E-Commerce

## Visão Geral

Este documento detalha os requisitos funcionais e não funcionais, arquitetura, endpoints, modelos de dados e outras informações relevantes para o desenvolvimento e implantação de microsserviços para o e-commerce de produtos digitais.

## Objetivo

O objetivo desta aplicação é fornecer uma interface contendo recursos necessários para que clientes possam realizar a aquisição de produtos pela internet, através das opções disponíveis no sistema, como cadastro e autenticação de usuário, carrinho de compras, pagamentos digitais e outras funcionalidades necessárias para atender às necessidades dos clientes.

## Escopo

Os microsserviços foram divididos em 4 partes importantes do sistema, entendidas por:

- Usuários (Cadastro de Usuários)
- Produtos (Cadastro de Produtos)
- Pedidos (Pedidos de Vendas)
- Pagamentos (Checkout de Pagamentos)

Cada microsserviço será responsável por gerenciar recursos específicos do sistema, fornecendo endpoints para criação, leitura, atualização e exclusão (CRUD) desses recursos. Além disso, também serão responsáveis pela aplicação de políticas de segurança.

# Requisitos Funcionais

## Cadastro de Usuários

1. **Cadastro:** Permitir que os usuários se registrem no sistema informando nome de usuário, email e senha. O sistema deve criptografar a senha antes de armazená-la no banco de dados.
2. **Alterar o status do usuário:** Permitir aos administradores alterar o status de um usuário, como bloquear, desbloquear, em caso de atividade suspeita ou desligamento.
3. **Busca por status do usuário** (somente para administradores)
4. **Busca por id do usuário** (somente para administradores)
5. **Registro de Log de acesso:** Permitir o registro de acesso dos usuários.
6. **Tratamento de erros:** O sistema deve fornecer mensagens de erro claras e descritivas em caso de falha durante o processamento de uma solicitação. Os códigos de status HTTP apropriados devem ser retornados em todas as respostas.

## Cadastro de Produtos

1. **Cadastro de produto:** Permitir que os usuários registrados no sistema com perfil de moderador ou administrador possam cadastrar produtos no sistema informando os dados necessários para o seu cadastro.
2. **Alteração de produto:** Permitir que os usuários registrados no sistema com perfil de moderador ou administrador possam alterar produtos que já tenham sido cadastrados no sistema, informando os dados necessários para sua alteração, bem como ativar/inativar produtos.
3. **Busca de produtos por id**
4. **Busca de produtos por preço**

## Pedidos de Vendas

1. **Formas de pagamento:** Os pedidos deverão ter as formas de pagamento "boleto" ou "cartão" e é permitido somente uma forma de pagamento.
2. **Status de pagamento:** O sistema deverá atualizar o status do pedido de acordo com a situação de pagamento. caso o pedido tenha sido criado e ainda não foi pago, deverá constar o status "AGUARDANDO". caso já tenha sido pago. terá o status "CONFIRMADO".

## Pagamentos (Checkout de Vendas)

1. **Comunicação com Gateway de pagamentos:** O sistema deve possibilitar o usuário se comunicar de forma transparente com o gateway de pagamentos, informando seus dados de pagamento para serem enviados ao gateway. Após a confirmação do gateway, o sistema deverá importar os dados de retorno do gateway para o pedido e alterar o seu status para "CONFIRMADO".
2. **Alteração do status de aprovação:** O sistema deve possibilitar a alteração do status no pedido de venda para "CONFIRMADO".

## Requisitos Não Funcionais

### Segurança

Todas as operações devem ser protegidas por autenticação e autorização. Deve ser utilizada uma estratégia de autenticação segura, como JWT (JSON Web Tokens) e O-Auth para autenticação via API de terceiros.

### Desempenho

O microserviço deve ser capaz de lidar com um grande número de solicitações simultâneas.  
Deve ser otimizado para um tempo de resposta rápido.

### Escalabilidade

O microserviço deve ser facilmente escalável, permitindo adicionar mais instâncias conforme necessário para lidar com aumento de carga.

### Documentação da API

A API deve ser bem documentada, incluindo descrições claras de cada endpoint, parâmetros necessários, códigos de status HTTP retornados e exemplos de solicitações e respostas.  
Deve ser utilizado um padrão de documentação, como o Swagger ou o OpenAPI.

## Monitoramento e Registro

Deve ser implementado um sistema de monitoramento para acompanhar o desempenho e a disponibilidade do microserviço.

Todas as solicitações e respostas devem ser registradas para fins de auditoria e depuração.

## Padrões de Codificação

O código do microserviço deve seguir os padrões de codificação baseado nas boas práticas de programação, incluindo DDD, Arquitetura Hexagonal, estabelecidos pela equipe.

Deve ser realizada revisão de código para garantir a qualidade e a consistência do código-fonte.

## Arquitetura

Cada microserviço será desenvolvido utilizando algumas das seguintes tecnologias e componentes:

Spring Boot: Para criação rápida de API's Rest e microserviços.

Spring Security: Para autenticação e autorização.

Spring Data JPA: Para acesso a banco de dados com mapeamento objeto-relacional (ORM).

Banco de Dados Relacional: Como MySQL, PostgreSQL ou similar.

Banco de Dados Não-Relacional: Como MongoDB, Redis ou similar.

Conteinerização: Docker.

## Endpoints - Cadastro de Usuários

POST /api/v1/auth/register - Cadastro

POST /api/v1/auth/authenticate - Autenticação

POST /api/v1/admin/moderator - Criar moderador

PUT /api/v1/admin/moderator/{id} - Atualizar moderador

GET /api/v1/admin/moderators - Retornar todos os moderadores

DELETE /api/v1/admin/moderators/{id} - Deletar moderador

## Endpoints - Cadastro de Produtos

POST /api/v1/produtos - Criar produto

GET /api/v1/produtos?page=1&offset=20 - Retornar a busca de produtos paginada

GET /api/v1/produtos/{id} - Retornar produto por Id

PUT /api/v1/produtos/{id} - Alterar produto

DELETE /api/v1/produtos/{id} - Excluir produto

## Endpoint - Pedido de Vendas

POST /api/v1/pedidos - Criar pedido

GET /api/v1/pedidos/{id} - Retornar pedido por Id

GET /api/v1/pedidos/{usuarioid} - Retornar pedidos por Id de usuário

PUT /api/v1/pedidos/{id} - Atualizar pedido

PUT /api/v1/pedidos/{id}/status - Atualizar status do pedido

DELETE /api/v1/pedidos/{id} - Excluir pedido

## Endpoints - Checkout Pagamentos

POST /api/v1/pagamentos - Incluir pagamento no checkout

GET /api/v1/pagamento/{id} - Obter status do pagamento

# Modelos de Dados

Os seguintes modelos de dados serão utilizados pelos microserviços:

## **Usuario**

id: Long  
nome: String  
email: String  
senha: String

## **Produto**

id: Long  
nome: String  
descrição: String  
preço: Double  
categoria: String

## **Pedido**

id: Long  
usuario\_id: Long  
status: String  
forma\_pagamento: String  
dt\_hora\_pedido: Date  
valor\_total\_pedido: Double

## **ItensPedido**

id: Long  
pedido\_id: Long  
produto\_id: Long  
preço: Double  
desconto: Double

## **Pagamento**

id: Long  
pedido\_id: Long  
forma\_pagamento: String  
status\_pagamento: String  
valor\_pagamento: Double  
dt\_hora\_pagamento: Date

## Considerações de Segurança

Será utilizado o esquema de autenticação JWT para proteger os endpoints. Serão implementadas políticas de autorização para restringir o acesso a determinados endpoints com base nos papéis do usuário.

## Conclusão

Este documento define os requisitos funcionais e não funcionais, arquitetura, endpoints e modelos de dados para o desenvolvimento de microsserviços para o e-commerce de produtos digitais. A implementação bem-sucedida desses requisitos garantirá a segurança, desempenho e escalabilidade do sistema, atendendo às necessidades dos clientes e usuários finais.