# Documento de Arquitetura do MVP do e-Store

## 1. Introdução

### 1.1 Propósito

O propósito deste documento é descrever a arquitetura do MVP do sistema e-Store, destacando sua estrutura, componentes, tecnologias e fluxos de dados. Este MVP servirá como uma versão inicial do sistema de gestão para lojas, focando nas funcionalidades essenciais para atender as necessidades dos usuários-alvo.

### 1.2 Escopo

O MVP incluirá funcionalidades básicas como:

* Controle de estoque
* Registro de vendas (PDV)
* Geração de relatórios financeiros
* Cadastro de produtos e clientes
* Gestão de usuários

## 2. Visão Geral da Arquitetura

A arquitetura do MVP do e-Store será baseada em uma abordagem de microsserviços, que permitirá uma melhor escalabilidade e manutenção. O sistema será dividido em módulos que se comunicam entre si através de APIs RESTful.

### 2.1 Estrutura Geral

A arquitetura é composta por:

* **Frontend:** Aplicação web que interage com os usuários.
* **Backend:** Conjunto de microsserviços que processam a lógica de negócios.
* **Banco de Dados:** Armazenamento persistente de dados do sistema.
* **Integrações:** APIs de terceiros para pagamento e emissão de notas fiscais.

### 2.2 Diagrama de Arquitetura

+---------------------------------+

| Frontend |

| (Aplicação Web - React/Vue) |

+---------------------------------+

|

| API (REST)

|

+------------------------------+

| Backend |

| +---------+ +--------------+ |

| | PDV | | Estoque | |

| | Serviço | | Serviço | |

| +---------+ +--------------+ |

| | | |

| +--------------+ |

| | |

| +-------------------------+ |

| | Relatórios Financeiros | |

| | Serviço | |

| +-------------------------+ |

| | |

| +-------------------+

| |

| +-----------------+ +----------------------+

| | Autenticação | | Gestão de Usuários |

| | Serviço | | Serviço |

| +-----------------+ +----------------------+

+-------------------------------+

|

| Banco de Dados (PostgreSQL/MySQL)

|

+------------------+

| Banco de Dados |

| (SQL) |

+------------------+

## 3. Componentes do Sistema

### 3.1 Frontend

* **Tecnologias:** React ou Vue.js para desenvolvimento de uma interface amigável e responsiva.
* **Funcionalidades:** Dashboard, telas de cadastro, visualização de relatórios e PDV.

### 3.2 Backend

* **Tecnologias:** Node.js com Express ou Python com Flask/Django.
* **Microserviços:**
  + **Serviço de PDV:** Gerencia vendas e emite notas fiscais.
  + **Serviço de Estoque:** Controla entradas e saídas de produtos, incluindo alertas de estoque baixo.
  + **Serviço de Relatórios Financeiros:** Gera relatórios de vendas e despesas.
  + **Serviço de Autenticação:** Gerencia login e segurança de usuários.
  + **Serviço de Gestão de Usuários:** Administra dados de funcionários e permissões.

### 3.3 Banco de Dados

* **Tecnologia:** PostgreSQL ou MySQL, dependendo das necessidades de escalabilidade.
* **Estrutura:** Tabelas para produtos, vendas, usuários, estoque e relatórios.

### 3.4 Integrações

* **Gateways de Pagamento:** Integração com APIs como PagSeguro ou Stripe para processamento de pagamentos.
* **Nota Fiscal Eletrônica:** Integração com sistemas de emissão de NF-e.

## 4. Fluxos de Dados

### 4.1 Fluxo de Venda

1. O vendedor registra a venda no PDV.
2. O serviço de PDV processa a transação e atualiza o banco de dados.
3. O estoque é atualizado automaticamente pelo serviço de estoque.
4. Um recibo é gerado e enviado ao cliente.

### 4.2 Fluxo de Relatórios

1. O gestor solicita relatórios financeiros através do frontend.
2. O serviço de relatórios financeiros consulta o banco de dados.
3. Os dados são processados e o relatório é enviado ao frontend.

## 5. Considerações de Segurança

* **Autenticação:** Implementação de autenticação JWT para segurança nas APIs.
* **Criptografia:** Dados sensíveis, como informações de pagamento, devem ser criptografados.
* **Segurança de Dados:** Acesso restrito aos dados com base em permissões de usuário.

## 6. Conclusão

Este documento fornece uma visão clara da arquitetura do MVP do e-Store, delineando os componentes principais, tecnologias e fluxos de dados. A estrutura modular permitirá uma implementação ágil e uma base sólida para futuras melhorias e expansões.