



# **Sistema de gestão administrativa para almoxarifado militar**





## **Sumário**

1. Introdução
2. Finalidade
3. Objetivo
4. Escopo
5. Visão Geral
6. Visão de Casos de Uso
7. Documentação API
  - 7.1 Objetivo
  - 7.2 Tags
  - 7.3 Endpoints



## 1. Introdução

Este documento descreve o desenvolvimento do sistema de gerenciamento de estoque da Base Administrativa da Guarnição de Caçapava, com foco no **almoxarifado** e na **farmácia militar**. O objetivo é substituir os processos manuais que hoje geram retrabalho, atrasos e riscos de erros, implementando uma solução tecnológica moderna, rastreável e eficiente.

## 2. Finalidade

A finalidade do projeto é oferecer um aplicativo web e mobile que possibilite:

- Controle centralizado do almoxarifado e da farmácia.
- Identificação rápida de itens via **QR Code**.
- Emissão de relatórios estratégicos para suporte logístico.
- Prevenção de perdas e desabastecimento por meio de alertas automáticos.
- Gestão eficiente de pedidos e usuários com diferentes níveis de acesso.



### 3. Objetivo

O objetivo principal é **garantir agilidade, segurança e precisão no gerenciamento de materiais** da Base, otimizando tanto o processo de requisição quanto o acompanhamento dos estoques.

### 4. Escopo

O sistema abrangerá as seguintes funcionalidades:

- Cadastro de produtos manual e via QR Code.
- Geração automática de QR Codes para impressão.
- Cadastro e gestão de pedidos.
- Controle de inventário com filtros avançados.
- Ajuste manual e exclusão de itens.
- Emissão de relatórios estratégicos.
- Estimativa de demanda futura.
- Gerenciamento de usuários com perfis distintos.
- Envio automático de e-mails de cobrança para fornecedores.

### 5. Visão Geral

A solução contará com uma interface **simples, responsiva e segura**, acessível por múltiplos dispositivos. Os principais recursos incluem:

- **Autenticação segura via login e senha.**
- **Leitura e impressão de QR Code.**



- **Dashboard interativo** com histórico de movimentações, inventário e relatórios.
- **Notificações inteligentes** para alertas de estoque.
- **Gestão multiusuário** com diferentes níveis de acesso.

## 6. Visão de Casos de Uso

- **Administrador:**
  - Cadastrar/editar produtos, pedidos, usuários e seções.
  - Gerar relatórios e exportar dados em PDF/Excel.
  - Configurar alertas de estoque mínimo.
  - Enviar cobranças por e-mail a fornecedores.
- **Usuário (comum):**
  - Visualizar estoque e histórico de movimentações.
  - Solicitar materiais (pedidos).
  - Atualizar e visualizar dados de perfil.
- **Paciente (na farmácia):**
  - Consultar agendamentos de exames.
  - Cancelar agendamentos.
  - Acompanhar retiradas e instruções de preparo.



## 7. Documentação API

### 7.1 Objetivo

O objetivo deste documento tem como informar ao leitor sobre as mecânicas que foram desenvolvidas em nossa aplicação.

### 7.2 Tags

Abaixo estão listadas quais foram as seguintes tags utilizadas:

- MerchandiseTypes
- Auth
- Merchandise
- Orders
- Sections
- Stock

### 7.3 Endpoints

#### MerchandiseTypes

- **Caminho:** /merchandise-types  
**Método:** GET  
**Parâmetros:** Nenhum  
**Response:** (200, OK)
- **Caminho:** /merchandise-types/:id  
**Método:** GET  
**Parâmetros:** id (path)  
**Response:** (200, OK)



- **Caminho:** /merchandise-types  
**Método:** POST  
**Parâmetros:** Nenhum  
**Response:** (200, OK)
- **Caminho:** /merchandise-types/:id  
**Método:** PUT  
**Parâmetros:** id (path)  
**Response:** (200, OK)
- **Caminho:** /merchandise-types/:id  
**Método:** DELETE  
**Parâmetros:** id (path)  
**Response:** (200, OK)

## Auth

- **Caminho:** /auth/register  
**Método:** POST  
**Parâmetros:** Nenhum  
**Response:** (200, OK)
- **Caminho:** /auth/login  
**Método:** POST  
**Parâmetros:** Nenhum  
**Response:** (200, OK)
- **Caminho:** /auth/logout  
**Método:** POST  
**Parâmetros:** Nenhum  
**Response:** (200, OK)
- **Caminho:** /auth/me  
**Método:** GET



**Parâmetros:** Nenhum

**Response:** (200, OK)

- **Caminho:** /auth/refresh

**Método:** POST

**Parâmetros:** Nenhum

**Response:** (200, OK)

## Merchandise

- **Caminho:** /merchandise

**Método:** GET

**Parâmetros:** Nenhum

**Response:** (200, OK)

- **Caminho:** /merchandise/:id

**Método:** GET

**Parâmetros:** id (path)

**Response:** (200, OK)

- **Caminho:** /merchandise

**Método:** POST

**Parâmetros:** Nenhum

**Response:** (200, OK)

- **Caminho:** /merchandise/:id

**Método:** PUT

**Parâmetros:** id (path)

**Response:** (200, OK)

- **Caminho:** /merchandise/:id

**Método:** DELETE

**Parâmetros:** id (path)

**Response:** (200, OK)





## Orders

- **Caminho:** /orders  
**Método:** GET  
**Parâmetros:** Nenhum  
**Response:** (200, OK)
- **Caminho:** /orders/:id  
**Método:** GET  
**Parâmetros:** id (path)  
**Response:** (200, OK)
- **Caminho:** /orders  
**Método:** POST  
**Parâmetros:** Nenhum  
**Response:** (200, OK)
- **Caminho:** /orders/:id  
**Método:** PUT  
**Parâmetros:** id (path)  
**Response:** (200, OK)
- **Caminho:** /orders/:id  
**Método:** DELETE  
**Parâmetros:** id (path)  
**Response:** (200, OK)

## Sections

- **Caminho:** /sections  
**Método:** GET



**Parâmetros:** Nenhum

**Response:** (200, OK)

- **Caminho:** /sections/:id

**Método:** GET

**Parâmetros:** id (path)

**Response:** (200, OK)

- **Caminho:** /sections

**Método:** POST

**Parâmetros:** Nenhum

**Response:** (200, OK)

- **Caminho:** /sections/:id

**Método:** PUT

**Parâmetros:** id (path)

**Response:** (200, OK)

- **Caminho:** /sections/:id

**Método:** DELETE

**Parâmetros:** id (path)

**Response:** (200, OK)

## Stock

- **Caminho:** /stock

**Método:** GET

**Parâmetros:** Nenhum

**Response:** (200, OK)

- **Caminho:** /stock/:id

**Método:** GET

**Parâmetros:** id (path)

**Response:** (200, OK)



- **Caminho:** /stock  
**Método:** POST  
**Parâmetros:** Nenhum  
**Response:** (200, OK)
- **Caminho:** /stock/:id  
**Método:** PUT  
**Parâmetros:** id (path)  
**Response:** (200, OK)
- **Caminho:** /stock/:id  
**Método:** DELETE  
**Parâmetros:** id (path)  
**Response:** (200, OK)