



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 de pedidos e usuários com diferentes níveis de acesso.
- Gestão de notas de empenhos.
- Disparo de e-mails de cobrança para os fornecedores.



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.
- Cadastro e gestão de notas de empenho.
- 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:

- Auth - Autenticação e gerenciamento de usuários
- MerchandiseTypes - Gestão de tipos de mercadorias
- Merchandise - Gestão de mercadorias (produtos individuais)
- Orders - Gestão de pedidos
- Sections - Gestão de seções
- Stock - Gestão de estoques
- Fornecedores (Suppliers) - Gestão de fornecedores
- Reports - Relatórios e dashboards

7.3 Endpoints

AUTH

- **Caminho:** /auth/user-data/:email

Método: GET

Parâmetros: email (path)

Response: (200, OK)

- **Caminho:** /auth/register

Método: POST



Parâmetros: Nenhum

Response: (201, Created)

- **Caminho:** /auth/login

Método: POST

Parâmetros: Nenhum

Response: (200, OK)

- **Caminho:** /auth/forgot-password

Método: PUT

Parâmetros: Nenhum

Response: (200, OK)

- **Caminho:** /auth/users

Método: GET

Parâmetros: Nenhum

Response: (200, OK)

- **Caminho:** /auth/users/:id

Método: PUT

Parâmetros: id (path)

Response: (200, OK)

- **Caminho:** /auth/users/:id

Método: DELETE

Parâmetros: id (path)



Response: (200, OK)

- **Caminho:** /auth/users/:userId/stocks

Método: POST

Parâmetros: userId (path)

Response: (200, OK)

- **Caminho:** /auth/users/:userId/stocks/:stockId

Método: DELETE

Parâmetros: userId (path), stockId (path)

Response: (200, OK)

- **Caminho:** /auth/users/:userId/stocks

Método: GET

Parâmetros: userId (path)

Response: (200, OK)

- **Caminho:** /auth/change-password

Método: PUT

Parâmetros: Nenhum

Response: (200, OK)

MERCHANDISETYPES

- **Caminho:** /merchandise-types/:id/logs

Método: GET

Parâmetros: id (path)



Response: (200, OK)

- **Caminho:** /merchandise-types/:id/merchandises

Método: GET

Parâmetros: id (path)

Response: (200, OK)

- **Caminho:** /merchandise-types/:id/quantity-total

Método: PATCH

Parâmetros: id (path)

Response: (200, OK)

- **Caminho:** /merchandise-types/:stockId

Método: GET

Parâmetros: stockId (path)

Response: (200, OK)

- **Caminho:** /merchandise-types/details/:id

Método: GET

Parâmetros: id (path)

Response: (200, OK)

- **Caminho:** /merchandise-types

Método: POST

Parâmetros: Nenhum

Response: (201, Created)



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

Método: PUT

Parâmetros: id (path)

Response: (200, OK)

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

Método: DELETE

Parâmetros: id (path)

Response: (204, No Content)

MERCHANDISE

- **Caminho:** /merchandise

Método: GET

Parâmetros: Nenhum

Response: (200, OK)

- **Caminho:** /merchandise/stock-alerts

Método: GET

Parâmetros: Nenhum

Response: (200, OK)

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

Método: GET

Parâmetros: id (path)

Response: (200, OK)



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

Método: GET

Parâmetros: id (path)

Response: (200, OK)

- **Caminho:** /merchandise

Método: POST

Parâmetros: Nenhum

Response: (201, Created)

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

Método: PUT

Parâmetros: id (path)

Response: (200, OK)

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

Método: DELETE

Parâmetros: id (path)

Response: (204, No Content)

ORDERS

- **Caminho:** /orders/:stockId

Método: GET

Parâmetros: stockId (path)

Response: (200, OK)



- **Caminho:** /orders/:id
Método: GET
Parâmetros: id (path)
Response: (200, OK)
- **Caminho:** /orders
Método: POST
Parâmetros: Nenhum
Response: (201, Created)
- **Caminho:** /orders/:id
Método: PUT
Parâmetros: id (path)
Response: (200, OK)
- **Caminho:** /orders/:id/status
Método: PATCH
Parâmetros: id (path)
Response: (200, OK)
- **Caminho:** /orders/:id/withdrawal-date
Método: PATCH
Parâmetros: id (path)
Response: (200, OK)
- **Caminho:** /orders/:id
Método: DELETE



Parâmetros: id (path)

Response: (204, No Content)

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: (201, Created)

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

Método: PUT

Parâmetros: id (path)

Response: (200, OK)

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

Método: DELETE

Parâmetros: id (path)



Response: (204, No Content)

- **Caminho:** /sections/:id/consumption-average

Método: GET

Parâmetros: id (path), startDate (query), endDate (query)

Response: (200, OK)

STOCK

- **Caminho:** /stock/:userId

Método: GET

Parâmetros: userId (path)

Response: (200, OK)

- **Caminho:** /stock/historico/:merchandiseId

Método: GET

Parâmetros: merchandiseId (path), tipo (query), inicio (query), fim (query)

Response: (200, OK)

- **Caminho:** /stock

Método: POST

Parâmetros: Nenhum

Response: (201, Created)

- **Caminho:** /stock/:stockId

Método: PUT

Parâmetros: stockId (path)



Response: (200, OK)

- **Caminho:** /stock/:stockId

Método: DELETE

Parâmetros: stockId (path)

Response: (204, No Content)

SUPPLIERS

- **Caminho:** /suppliers

Método: POST

Parâmetros: Nenhum

Response: (201, Created)

- **Caminho:** /suppliers

Método: GET

Parâmetros: search (query), isActive (query), page (query), limit (query), sortBy (query), sortOrder (query)

Response: (200, OK)

- **Caminho:** /suppliers/:id

Método: GET

Parâmetros: id (path)

Response: (200, OK)

- **Caminho:** /suppliers/:id

Método: PUT

Parâmetros: id (path)



Response: (200, OK)

- **Caminho:** /suppliers/:id

Método: DELETE

Parâmetros: id (path)

Response: (200, OK)

REPORTS

- **Caminho:** /reports/dashboard

Método: GET

Parâmetros: format (query), stockId (query), startDate (query), endDate (query), includeOrders (query), includeMerchandise (query), includeStock (query)

Response: (200, OK)

- **Caminho:** /reports/dashboard/summary

Método: GET

Parâmetros: stockId (query), startDate (query), endDate (query)

Response: (200, OK)

- **Caminho:** /reports/dashboard/orders-by-period

Método: GET

Parâmetros: stockId (query), period (query), startDate (query), endDate (query)

Response: (200, OK)

- **Caminho:** /reports/dashboard/product-status

Método: GET

Parâmetros: stockId (query)



Response: (200, OK)

- **Caminho:** /reports/dashboard/orders-by-section

Método: GET

Parâmetros: stockId (query), startDate (query), endDate (query)

Response: (200, OK)

- **Caminho:** /reports/dashboard/top-products

Método: GET

Parâmetros: stockId (query), startDate (query), endDate (query)

Response: (200, OK)

- **Caminho:** /reports/dashboard/stock-alerts

Método: GET

Parâmetros: stockId (query)

Response: (200, OK)

- **Caminho:** /reports/dashboard/complete

Método: GET

Parâmetros: stockId (query), period (query), startDate (query), endDate (query)

Response: (200, OK)

- **Caminho:** /reports/dashboard/complete/report

Método: GET

Parâmetros: format (query), stockId (query), period (query), startDate (query),
endDate (query)



Response: (200, OK)