



Sistema de gestão administrativa para almoxarifado militar



2025.2



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.
- Cadastro e gestão de exames
- Cadastro e gestão de pacientes
- Gestão de agendamentos
- Envio automático de e-mails para os pacientes.



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 estoque:**
 - 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.
- **Administrador agendamento:**
 - Cadastrar/editar tipos de exame.
 - Cadastrar/editar pacientes.
 - Gerenciar agendamento de exames.
- **Usuário (comum):**
 - Visualizar estoque e histórico de movimentações.
 - Solicitar materiais (pedidos).



- Atualizar e visualizar dados de perfil.

- **Paciente:**

- Consultar agendamentos de exames.
- 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

Tags Utilizadas

Abaixo estão listadas as seguintes tags utilizadas:

- **AUTH** - Autenticação e gerenciamento de usuários
- **APPOINTMENTS** - Agendamentos de pacientes
- **COMMITMENT NOTES** - Notas de empenho (pedidos administrativos)
- **EMAIL LOGS** - Logs de envio de emails
- **EMAIL TEMPLATES** - Templates de emails para disparos automáticos
- **EXAM PREPARATIONS** - Instruções de preparo de exames
- **EXAM TYPES** - Tipos de exames
- **MERCHANDISE** - Gestão de produtos individuais
- **MERCHANDISE TYPES** - Gestão de tipos/categorias de produtos
- **ORDERS** - Gestão de pedidos
- **REPORTS** - Relatórios e dashboards
- **SECTIONS** - Gestão de seções/departamentos
- **STOCK** - Gestão de estoques
- **FORNECEDORES** - Gestão de fornecedores



7.3 Endpoints

AUTH - Autenticação e Gerenciamento de Usuários

- Caminho: /auth/user-data/{email}

Método: GET

Parâmetros: email (path, obrigatório)

Response: (200, OK), (404, Usuário não encontrado)

- Caminho: /auth/register

Método: POST

Parâmetros: email, name, firebaseUid (obrigatórios), role (opcional)

Response: (201, Criado), (400, Dados inválidos)

- Caminho: /auth/users

Método: POST

Parâmetros: email, name, firebaseUid (obrigatórios em body)

Response: (201, Usuário criado com sucesso)

- Caminho: /auth/login

Método: POST

Parâmetros: email, firebaseUid (obrigatórios)

Response: (200, OK), (401, Credenciais inválidas)

- Caminho: /auth/forgot-password

Método: PUT

Parâmetros: email (obrigatório)

Response: (200, OK), (404, Email não encontrado)

- Caminho: /auth/users

Método: GET

Parâmetros: Nenhum

Response: (200, OK)

- Caminho: /auth/users/{id}

Método: PUT

Parâmetros: id (path, obrigatório), name, email, role, isActive (body, opcionais)

Response: (200, OK)



- Caminho: /auth/users/{id}

Método: DELETE

Parâmetros: id (path, obrigatório)

Response: (200, OK)

- Caminho: /auth/users/{userId}/stocks

Método: POST

Parâmetros: userId (path, obrigatório), stockId, responsibility (body, obrigatórios)

Response: (200, OK)

- Caminho: /auth/users/{userId}/stocks/{stockId}

Método: DELETE

Parâmetros: userId, stockId (path, obrigatórios)

Response: (200, OK)

- Caminho: /auth/users/{userId}/stocks

Método: GET

Parâmetros: userId (path, obrigatório)

Response: (200, OK), (404, Usuário não encontrado)

- Caminho: /auth/change-password

Método: PUT

Parâmetros: email, currentPassword, newPassword (obrigatórios)

Response: (200, OK), (400, Dados inválidos), (404, Usuário não encontrado)

APPOINTMENTS - Agendamentos de Pacientes

- Caminho: /appointments

Método: GET

Parâmetros: Nenhum

Response: (200, OK)

- Caminho: /appointments

Método: POST

Parâmetros: (obrigatórios conforme especificação)

Response: (201, Criado)



- Caminho: /appointments/{id}

Método: PATCH

Parâmetros: id (path, obrigatório)

Response: (200, OK)

- Caminho: /appointments/{id}

Método: DELETE

Parâmetros: id (path, obrigatório)

Response: (200, OK)

- Caminho: /appointments/{id}/receipt

Método: GET

Parâmetros: id (path, obrigatório)

Response: (200, OK)

- Caminho: /appointments/report

Método: GET

Parâmetros: Nenhum

Response: (200, OK)

- Caminho: /appointments/patients

Método: GET

Parâmetros: Nenhum

Response: (200, OK)

COMMITMENT NOTES - Notas de Empenho

- Caminho: /commitment-notes

Método: GET

Parâmetros: Nenhum

Response: (200, OK)

- Caminho: /commitment-notes/{id}

Método: GET

Parâmetros: id (path, UUID, obrigatório)

Response: (200, OK), (404, Não encontrado)



- Caminho: /commitment-notes

Método: POST

Parâmetros: supplierId, valor, numeroNota, dataNota, ug, razaoSocial, cnpj
(obrigatórios)

Response: (201, Criado)

- Caminho: /commitment-notes/{id}

Método: PUT

Parâmetros: id (path, obrigatório)

Response: (200, OK)

- Caminho: /commitment-notes/{id}/admin-fields

Método: PATCH

Parâmetros: id (path, obrigatório)

Response: (200, OK)

- Caminho: /commitment-notes/{id}/finalize

Método: POST

Parâmetros: id (path, obrigatório)

Response: (200, OK)

- Caminho: /commitment-notes/{id}

Método: DELETE

Parâmetros: id (path, obrigatório)

Response: (200, OK)

EMAIL LOGS - Logs de Emails

- Caminho: /email-logs/commitment-notes/{id}

Método: GET

Parâmetros: id (path, obrigatório)

Response: (200, OK)

- Caminho: /email-logs/suppliers/{id}

Método: GET

Parâmetros: id (path, obrigatório)

Response: (200, OK)



EMAIL TEMPLATES - Templates de Emails

- Caminho: /email-templates/allowed-vars

Método: GET

Parâmetros: Nenhum

Response: (200, OK)

- Caminho: /email-templates

Método: GET

Parâmetros: Nenhum

Response: (200, OK)

- Caminho: /email-templates/{id}

Método: GET

Parâmetros: id (path, obrigatório)

Response: (200, OK), (404, Não encontrado)

- Caminho: /email-templates/{id}/versions

Método: GET

Parâmetros: id (path, obrigatório)

Response: (200, OK)

- Caminho: /email-templates

Método: POST

Parâmetros: (obrigatórios conforme especificação)

Response: (201, Criado)

- Caminho: /email-templates/{id}

Método: PATCH

Parâmetros: id (path, obrigatório)

Response: (200, OK)

- Caminho: /email-templates/{id}/preview

Método: POST

Parâmetros: id (path, obrigatório)

Response: (200, OK)



EXAM PREPARATIONS - Instruções de Preparo

- Caminho: /exam-preparations/exam-type/{examTypeId}

Método: GET

Parâmetros: examTypeId (path, UUID, obrigatório)

Response: (200, OK)

- Caminho: /exam-preparations/{id}

Método: GET

Parâmetros: id (path, UUID, obrigatório)

Response: (200, OK), (404, Não encontrado)

- Caminho: /exam-preparations

Método: POST

Parâmetros: examTypeId, titulo, conteudo (obrigatórios)

Response: (201, Criado)

- Caminho: /exam-preparations/{id}

Método: PATCH

Parâmetros: id (path, obrigatório)

Response: (200, OK)

- Caminho: /exam-preparations/{id}

Método: DELETE

Parâmetros: id (path, obrigatório)

Response: (200, OK)

EXAM TYPES - Tipos de Exames

- Caminho: /exam-types

Método: GET

Parâmetros: Nenhum

Response: (200, OK)

- Caminho: /exam-types

Método: POST

Parâmetros: (obrigatórios conforme especificação)

Response: (201, Criado)



- Caminho: /exam-types/{id}

Método: PATCH

Parâmetros: id (path, obrigatório)

Response: (200, OK)

- Caminho: /exam-types/{id}

Método: DELETE

Parâmetros: id (path, obrigatório)

Response: (200, OK)

MERCHANDISE - Gestão de Produtos

- Caminho: /merchandise

Método: GET

Parâmetros: Nenhum

Response: (200, OK), (401, Não autorizado), (403, Acesso negado)

- Caminho: /merchandise/stock-alerts

Método: GET

Parâmetros: Nenhum

Response: (200, OK)

- Caminho: /merchandise/{id}

Método: GET

Parâmetros: id (path, UUID, obrigatório)

Response: (200, OK), (404, Não encontrado), (401, Não autorizado), (403, Acesso negado)

- Caminho: /merchandise/{id}/qrcode

Método: GET

Parâmetros: id (path, UUID, obrigatório)

Response: (200, PNG), (404, Não encontrado), (401, Não autorizado), (403, Acesso negado)

- Caminho: /merchandise

Método: POST

Parâmetros: batchId, typeId, quantity, status (obrigatórios)



Response: (201, Criado), (400, Dados inválidos), (401, Não autorizado), (403, Acesso negado)

- Caminho: /merchandise/{id}

Método: PUT

Parâmetros: id (path, obrigatório)

Response: (200, OK), (404, Não encontrado), (400, Dados inválidos), (401, Não autorizado), (403, Acesso negado)

- Caminho: /merchandise/{id}

Método: DELETE

Parâmetros: id (path, UUID, obrigatório)

Response: (204, Sem conteúdo), (404, Não encontrado), (401, Não autorizado), (403, Acesso negado)

MERCHANDISE TYPES - Gestão de Tipos de Produtos

- Caminho: /merchandise-types/{id}/logs

Método: GET

Parâmetros: id (path, obrigatório)

Response: (200, OK)

- Caminho: /merchandise-types/{id}/entry-history

Método: GET

Parâmetros: id (path, obrigatório)

Response: (200, OK), (400, Dados inválidos), (404, Não encontrado)

- Caminho: /merchandise-types/{id}/merchandises

Método: GET

Parâmetros: id (path, obrigatório)

Response: (200, OK), (400, Dados inválidos), (404, Não encontrado), (401, Não autorizado)

- Caminho: /merchandise-types/{id}/quantity-total

Método: PATCH

Parâmetros: id (path, obrigatório), quantityTotal (body, obrigatório)

Response: (200, OK), (400, Dados inválidos), (401, Não autorizado), (403, Acesso negado)



- Caminho: /merchandise-types/{stockId}

Método: GET

Parâmetros: stockId (path, obrigatório)

Response: (200, OK), (401, Não autorizado), (403, Acesso negado)

- Caminho: /merchandise-types/details/{id}

Método: GET

Parâmetros: id (path, UUID, obrigatório)

Response: (200, OK), (404, Não encontrado), (401, Não autorizado), (403, Acesso negado)

- Caminho: /merchandise-types

Método: POST

Parâmetros: name, recordNumber, unitOfMeasure, controlled, group, minimumStock, stockId (obrigatórios)

Response: (201, Criado), (400, Dados inválidos), (401, Não autorizado), (403, Acesso negado)

- Caminho: /merchandise-types/{id}

Método: PUT

Parâmetros: id (path, obrigatório)

Response: (200, OK), (404, Não encontrado), (400, Dados inválidos), (401, Não autorizado), (403, Acesso negado)

- Caminho: /merchandise-types/{id}

Método: DELETE

Parâmetros: id (path, UUID, obrigatório)

Response: (204, Sem conteúdo), (404, Não encontrado), (401, Não autorizado), (403, Acesso negado)

ORDERS - Gestão de Pedidos

- Caminho: /orders/{stockId}

Método: GET

Parâmetros: stockId (path, obrigatório)

Response: (200, OK)



- Caminho: /orders/{id}

Método: GET

Parâmetros: id (path, UUID, obrigatório)

Response: (200, OK), (404, Não encontrado)

- Caminho: /orders

Método: POST

Parâmetros: sectionId (obrigatório), withdrawalDate, status, orderItems (opcionais)

Response: (201, Criado), (400, Dados inválidos)

- Caminho: /orders/{id}

Método: PUT

Parâmetros: id (path, obrigatório)

Response: (200, OK), (404, Não encontrado), (400, Dados inválidos)

- Caminho: /orders/{id}/status

Método: PATCH

Parâmetros: id (path, obrigatório), status (body, obrigatório)

Response: (200, OK), (400, Dados inválidos), (404, Não encontrado)

- Caminho: /orders/{id}/withdrawal-date

Método: PATCH

Parâmetros: id (path, obrigatório), withdrawalDate (body, opcional)

Response: (200, OK), (400, Dados inválidos), (404, Não encontrado)

- Caminho: /orders/{id}

Método: DELETE

Parâmetros: id (path, UUID, obrigatório)

Response: (204, Sem conteúdo), (404, Não encontrado)

REPORTS - Relatórios e Dashboards

- Caminho: /reports/dashboard

Método: GET

Parâmetros: format (query, obrigatório), stockId (query, obrigatório), startDate, endDate (query, opcionais)

Response: (200, OK), (400, Parâmetros inválidos), (500, Erro interno)



- Caminho: /reports/dashboard/summary
Método: GET
Parâmetros: stockId (query, obrigatório), startDate, endDate (query, opcionais)
Response: (200, OK), (400, Parâmetros inválidos), (500, Erro interno)
- Caminho: /reports/dashboard/orders-by-period
Método: GET
Parâmetros: stockId (query, obrigatório), period (query, opcional)
Response: (200, OK), (400, Parâmetros inválidos), (500, Erro interno)
- Caminho: /reports/dashboard/product-status
Método: GET
Parâmetros: stockId (query, obrigatório)
Response: (200, OK), (400, Parâmetros inválidos), (500, Erro interno)
- Caminho: /reports/dashboard/orders-by-section
Método: GET
Parâmetros: stockId (query, obrigatório), startDate, endDate (query, opcionais)
Response: (200, OK), (400, Parâmetros inválidos), (500, Erro interno)
- Caminho: /reports/dashboard/top-products
Método: GET
Parâmetros: stockId (query, obrigatório), startDate, endDate (query, opcionais)
Response: (200, OK), (400, Parâmetros inválidos), (500, Erro interno)
- Caminho: /reports/dashboard/stock-alerts
Método: GET
Parâmetros: stockId (query, obrigatório)
Response: (200, OK), (400, Parâmetros inválidos), (500, Erro interno)
- Caminho: /reports/dashboard/complete
Método: GET
Parâmetros: stockId (query, obrigatório), period (query, opcional)
Response: (200, OK), (400, Parâmetros inválidos), (500, Erro interno)
- Caminho: /reports/dashboard/complete/report
Método: GET
Parâmetros: format (query, obrigatório), stockId (query, obrigatório), period (query,



opcional)

Response: (200, OK), (400, Parâmetros inválidos), (500, Erro interno)

- Caminho: /reports/forecast/balance

Método: GET

Parâmetros: months (query, opcional)

Response: (200, OK), (500, Erro interno)

SECTIONS - Gestão de Seções

- Caminho: /sections

Método: GET

Parâmetros: Nenhum

Response: (200, OK)

- Caminho: /sections/{id}

Método: GET

Parâmetros: id (path, UUID, obrigatório)

Response: (200, OK), (404, Não encontrado)

- Caminho: /sections

Método: POST

Parâmetros: name (obrigatório)

Response: (201, Criado), (400, Dados inválidos)

- Caminho: /sections/{id}

Método: PUT

Parâmetros: id (path, obrigatório), name (body, opcional)

Response: (200, OK), (404, Não encontrado), (400, Dados inválidos)

- Caminho: /sections/{id}

Método: DELETE

Parâmetros: id (path, UUID, obrigatório)

Response: (204, Sem conteúdo), (404, Não encontrado)

- Caminho: /sections/{id}/consumption-average

Método: GET

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



obrigatório)

Response: (200, OK), (400, Dados inválidos), (404, Não encontrado), (500, Erro ao calcular)

STOCK - Gestão de Estoques

- Caminho: /stock/{userId}

Método: GET

Parâmetros: userId (path, UUID, obrigatório)

Response: (200, OK), (404, Não encontrado)

- Caminho: /stock/historico/{merchandiseId}

Método: GET

Parâmetros: merchandiseId (path, obrigatório), tipo (query, opcional), inicio, fim (query, opcionais)

Response: (200, OK)

- Caminho: /stock

Método: POST

Parâmetros: name, location (obrigatórios), active (opcional)

Response: (201, Criado), (400, Dados inválidos), (401, Não autorizado), (403, Acesso negado)

- Caminho: /stock/{stockId}

Método: PUT

Parâmetros: stockId (path, obrigatório), name, location, active (body, opcionais)

Response: (200, OK), (404, Não encontrado), (400, Dados inválidos), (401, Não autorizado), (403, Acesso negado)

- Caminho: /stock/{stockId}

Método: DELETE

Parâmetros: stockId (path, UUID, obrigatório)

Response: (204, Sem conteúdo), (404, Não encontrado), (401, Não autorizado), (403, Acesso negado)



FORNECEDORES - Gestão de Fornecedores

- Caminho: /suppliers

Método: POST

Parâmetros: razaoSocial, cnpj, emailPrimario (obrigatórios)

Response: (201, Criado), (400, Dados inválidos)

- Caminho: /suppliers

Método: GET

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

Response: (200, OK)

- Caminho: /suppliers/{id}

Método: GET

Parâmetros: id (path, UUID, obrigatório)

Response: (200, OK), (404, Não encontrado)

- Caminho: /suppliers/{id}

Método: PUT

Parâmetros: id (path, obrigatório)

Response: (200, OK), (400, Dados inválidos), (404, Não encontrado)

- Caminho: /suppliers/{id}

Método: DELETE

Parâmetros: id (path, UUID, obrigatório)

Response: (200, OK), (404, Não encontrado)