

Sistema de Controle de Almoxarifado

By Anderson Landim

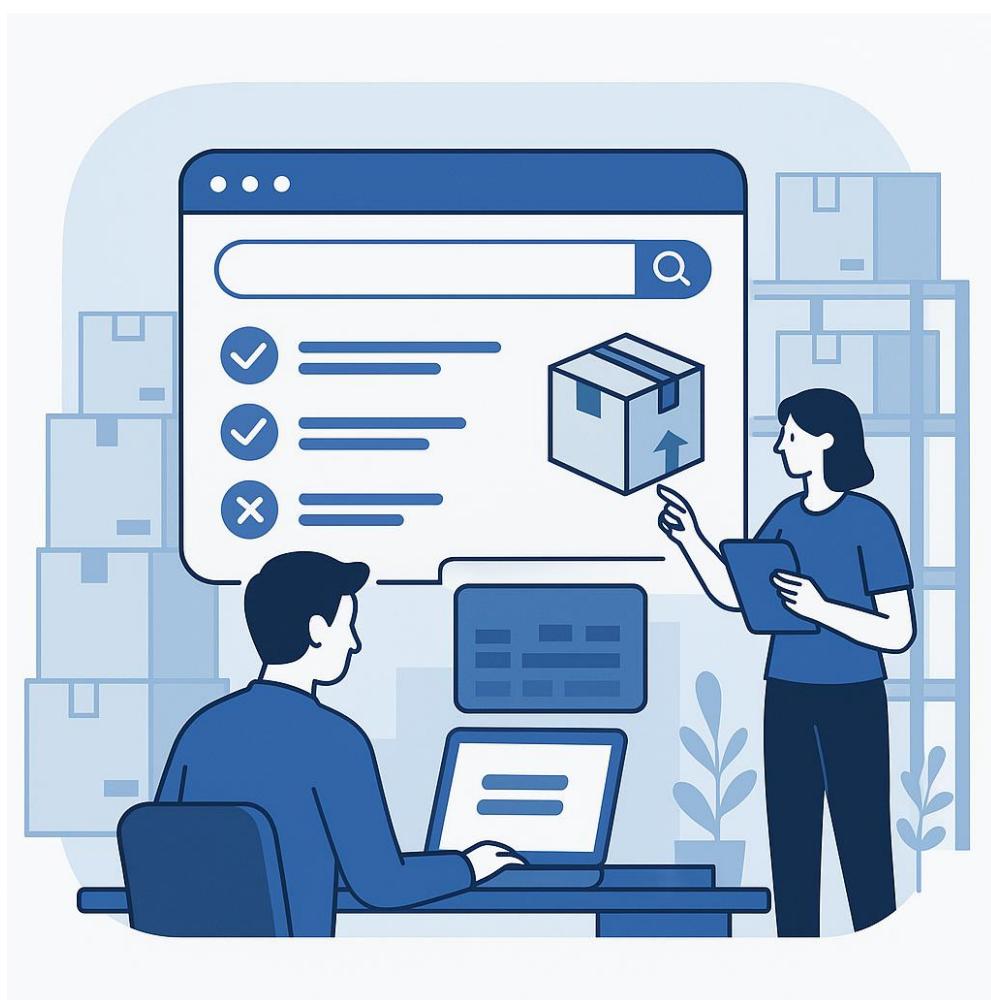
Índice

1. Introdução
2. Objetivo do Projeto
3. Tecnologias Utilizadas
4. Especificação do Sistema
 - Especificação Funcionais
 - Especificação Não Funcionais

1. Introdução

Este documento tem como finalidade apresentar o projeto de desenvolvimento de um sistema de controle de almoxarifado, desenvolvido com tecnologia atuais e modernas. O sistema foi projetado com foco em segurança, organização e usabilidade, visando atender às necessidades de gestão de estoque em ambientes organizacionais.

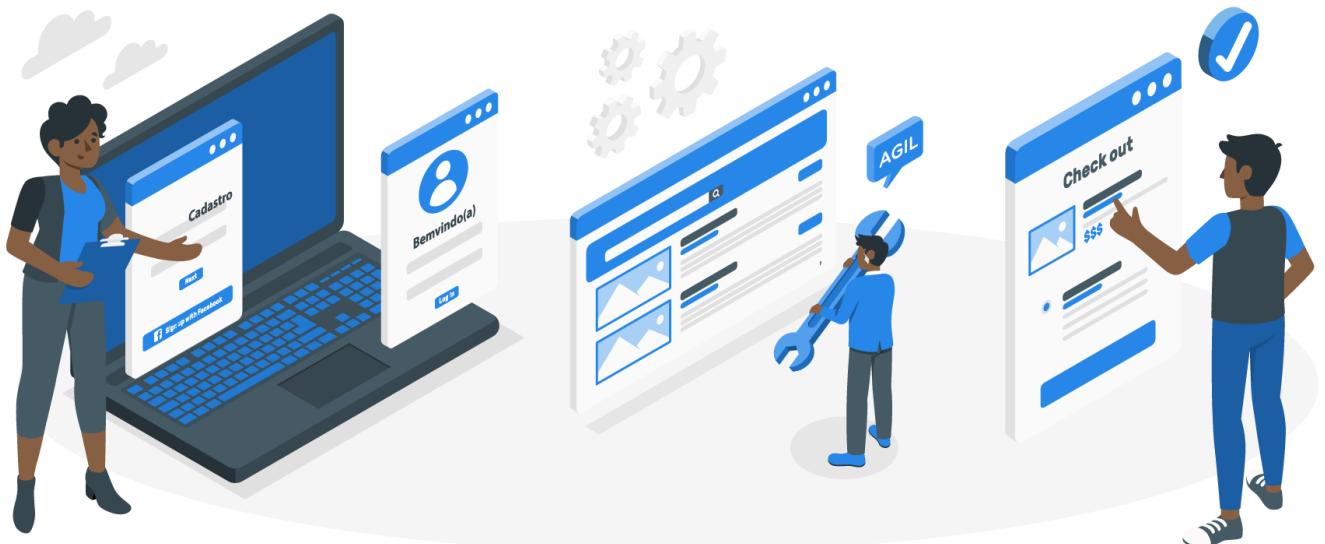
O Sistema de Almoxarifado é uma aplicação web projetada para facilitar o gerenciamento de itens de estoque em uma organização. O sistema será desenvolvido utilizando o framework Django, em Python, e contará com bibliotecas adicionais e uma interface interativa em HTML e CSS além de bootsTrap, que será compatível tanto para dispositivos desktop quanto para celulares, garantindo uma melhor visualização e localização de itens.



2. Objetivo do Projeto

Ser um sistema web capaz de:

- Gerenciar o almoxarifado de uma instituição por meio de funcionalidades de cadastro, edição e exclusão de itens;
- Permitir busca pública com visualização de itens com scroll infinit.;
- Melhorar a produtividade dos usuários no gerenciamento do estoque;
- Aumentar o acerto no preenchimento dos códigos dos itens, reduzindo erros de digitação e inconsistências;
- Facilitar a localização rápida dos itens, por meio de uma busca avançada e eficiente;
- Otimizar o gerenciamento logístico, proporcionando controle mais preciso sobre entradas, saídas e movimentações;
- Servir como um diferencial estratégico para o almoxarifado, promovendo mais controle, agilidade e confiabilidade nos processos internos.

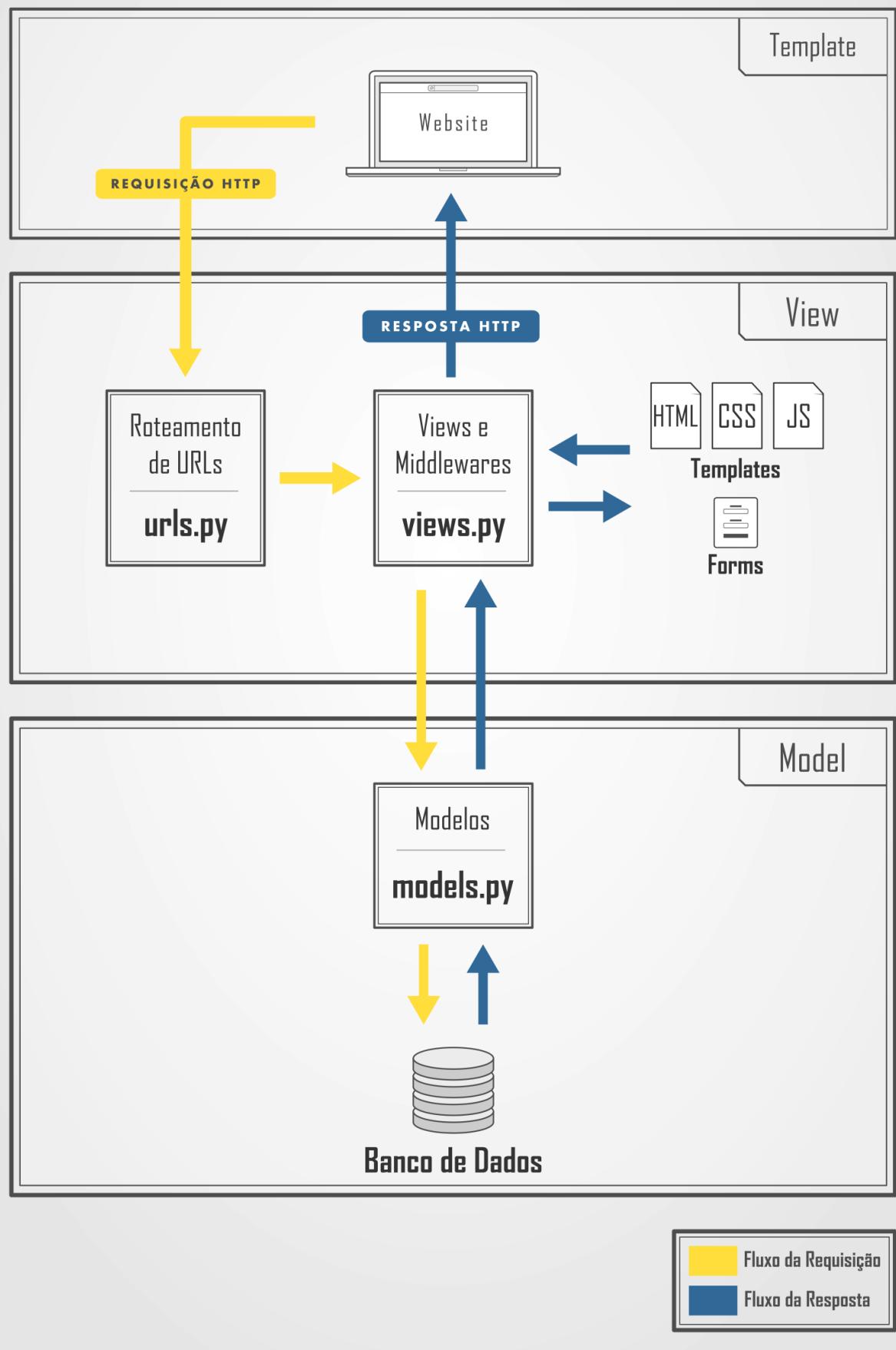


3. Tecnologias Utilizadas

- Linguagem Python 3
- Framework Django 5.2
- HTML, CSS e JavaScript
- SQLite3 (banco de dados)
- Bootstrap (interface)
- Bibliotecas acessórias:

```
asgiref==3.8.1 blinker==1.9.0 Brotli==1.1.0
certifi==2025.1.31 cffi==1.17.1 charset-normalizer==3.4.1
click==8.1.8 colorama==0.4.6 ConfigArgParse==1.7
Django==5.2 django-widget-tweaks==1.5.0
et_xmlfile==2.0.0 Flask==3.1.0 flask-cors==5.0.1 Flask-
Login==0.6.3 gevent==24.11.1 geventhttpclient==2.3.3
greenlet==3.1.1 idna==3.10 itsdangerous==2.2.0
Jinja2==3.1.6 locust==2.34.1 MarkupSafe==3.0.2
msgpack==1.1.0 numpy==2.2.4 openpyxl==3.1.5
pandas==2.2.3 pillow==11.1.0 ping3==4.0.8 psutil==7.0.0
pycparser==2.22 python-dateutil==2.9.0.post0
pytz==2025.2 pywin32==310 pyzmq==26.4.0
requests==2.32.3 setuptools==78.1.0 six==1.17.0
sqlparse==0.5.3 tzdata==2025.2 urllib3==2.4.0
uuid==1.30 Werkzeug==3.1.3 zope.event==5.0
zope.interface==7.2
```

ARQUITETURA DO django



4. Especificação do Sistema

1 Especificação Funcionais

O gerenciamento de itens será uma das funcionalidades principais do sistema, permitindo a interação direta com o estoque. Abaixo estão os requisitos principais para o gerenciamento:

O sistema deve permitir o cadastro de itens com os seguintes campos obrigatórios:

- Código único: Identificador único para cada item no estoque. (PK)
- Nome: Nome do item.
- Descrição: Descrição detalhada do item.
- Quantidade disponível: Quantidade atual do item no estoque.
- Foto: Imagem do item.

Os usuários poderão realizar uma busca rápida de itens pelo código, nome ou descrição, facilitando a localização de itens específicos no estoque.

1.2 Autenticação e Controle de Acesso

A segurança e a gestão de usuários serão realizadas por meio de um sistema de autenticação robusto, que permitirá o acesso ao sistema com diferentes níveis de permissão:

O sistema deve possuir um mecanismo de login e cadastro de usuários. Os usuários serão categorizados em três níveis hierárquicos com permissões diferentes:

- **Super_admin:** Este usuário tem permissões completas, podendo cadastrar, editar, excluir itens, além de gerenciar outros usuários com o perfil Almoxarife_cadastrados. O super_admin pode visualizar todos os registros de itens no sistema e acessar página de estatísticas e performance.
- **Almoxarife_cadastrados:** Este perfil tem permissão para cadastrar e editar itens, colocar descrições e imagens, excluir itens e atualizar base de dados.
- **Usuário_sem_cadastro:** Este usuário tem apenas permissão para visualizar os itens cadastrados, sem possibilidade de edição ou exclusão. E acesso ao suporte.

Resumo:

- Cadastro e edição de itens com múltiplas imagens
 - Cadastro de usuários com diferentes permissões
 - Busca dinâmica por itens (código, nome, descrição, quantidade)
 - Importação de dados via planilha Excel
 - Controle de estoque
 - Painel de estatísticas para super_admin
 - Visualização de catálogo público sem login
-

2 Especificação Não Funcionais

2.1 Tecnologias Utilizadas

A arquitetura do sistema será baseada em tecnologias amplamente utilizadas para desenvolvimento web, garantindo alta performance, segurança e escalabilidade:

- Backend: Django, um framework robusto e seguro, escrito em Python, que será responsável por gerenciar a lógica do servidor e a interação com o banco de dados.
- Frontend: HTML, CSS e JavaScript, responsáveis pela construção de uma interface amigável e responsiva, capaz de se adaptar a diferentes dispositivos (desktop e mobile).
- Banco de Dados: Para o desenvolvimento inicial, será utilizado o SQLite, um banco de dados leve e eficiente.

2.2 Desempenho

- A busca por código, nome e descrição dos itens será otimizada para garantir respostas rápidas, mesmo com um grande volume de dados.
- O sistema deve ser completamente responsivo, adaptável a diferentes dispositivos.

2.3 Segurança

Principais Proteções do Django

- Proteção contra SQL Injection – ORM do Django impede injeção de SQL.
- Proteção contra Cross-Site Scripting (XSS) – Escape automático de templates.
- Proteção contra Cross-Site Request Forgery (CSRF) – Tokens CSRF para evitar ações forçadas.
- Proteção contra Clickjacking – Cabeçalho X-Frame-Options para bloquear iframes maliciosos.
- Proteção contra Cross-Origin Resource Sharing (CORS) – Controle de acessos externos.
- Criptografia de Senhas – Uso de hash seguro (PBKDF2, Argon2, bcrypt).
- Proteção contra Directory Traversal – Impede acesso indevido a arquivos/diretórios.

- Segurança nos Cookies – HttpOnly, Secure e SameSite para evitar roubo de sessão.
- Proteção contra Host Header Attack – ALLOWED_HOSTS impede falsificação de cabeçalhos.
- Middleware de Segurança – HSTS e outras proteções para reforçar a segurança da aplicação.
- Controle de acesso baseado nos níveis hierárquicos.

BACKEND



FRONTEND



SEGURANÇA



DESEMPENHO



Considerações Finais

Este projeto foi mais do que um sistema de almoxarifado; foi uma experiência prática de desenvolvimento completo, da concepção à execução. Ele mostra como é possível aliar organização, segurança e eficiência em um sistema acessível, escalável e fácil de usar. O sistema está pronto para evoluir ainda mais e se adaptar a outros contextos.