



# RomaneiosApp - Plataforma de Gerenciamento e automação de fluxos de Suprimentos

License Apache 2.0

Bem-vindo ao **RomaneiosApp**, um sistema completo para gerenciamento de romaneios de entrega com funcionalidades avançadas de controle e comunicação.

Criado por **Diego Bernardes**, este projeto foi idealizado para trazer eficiência, segurança e usabilidade ao processo de suprimentos.

## ✨ Funcionalidades Principais

### Romaneios

- Criação e edição de romaneios
- Assinatura digital (nome + matrícula) via canvas
- Visualização de assinaturas
- Filtros dinâmicos por data, status e caixa

### Caixas

- Gerenciamento de caixas
- Visualização dos romaneios por caixa
- Envio em lote para assinaturas

### Notificações

- Sistema de notificações em tempo real
- Marcação de leitura (individual ou em massa)
- Contador de notificações não lidas

### Controle de Acesso

- 3 níveis: Usuário | Almoxarifado | Admin

## Delegação de Assinaturas

Autorize outros usuários a assinarem romaneios por você

## Relatórios Avançados

Geração de relatórios com filtros flexíveis

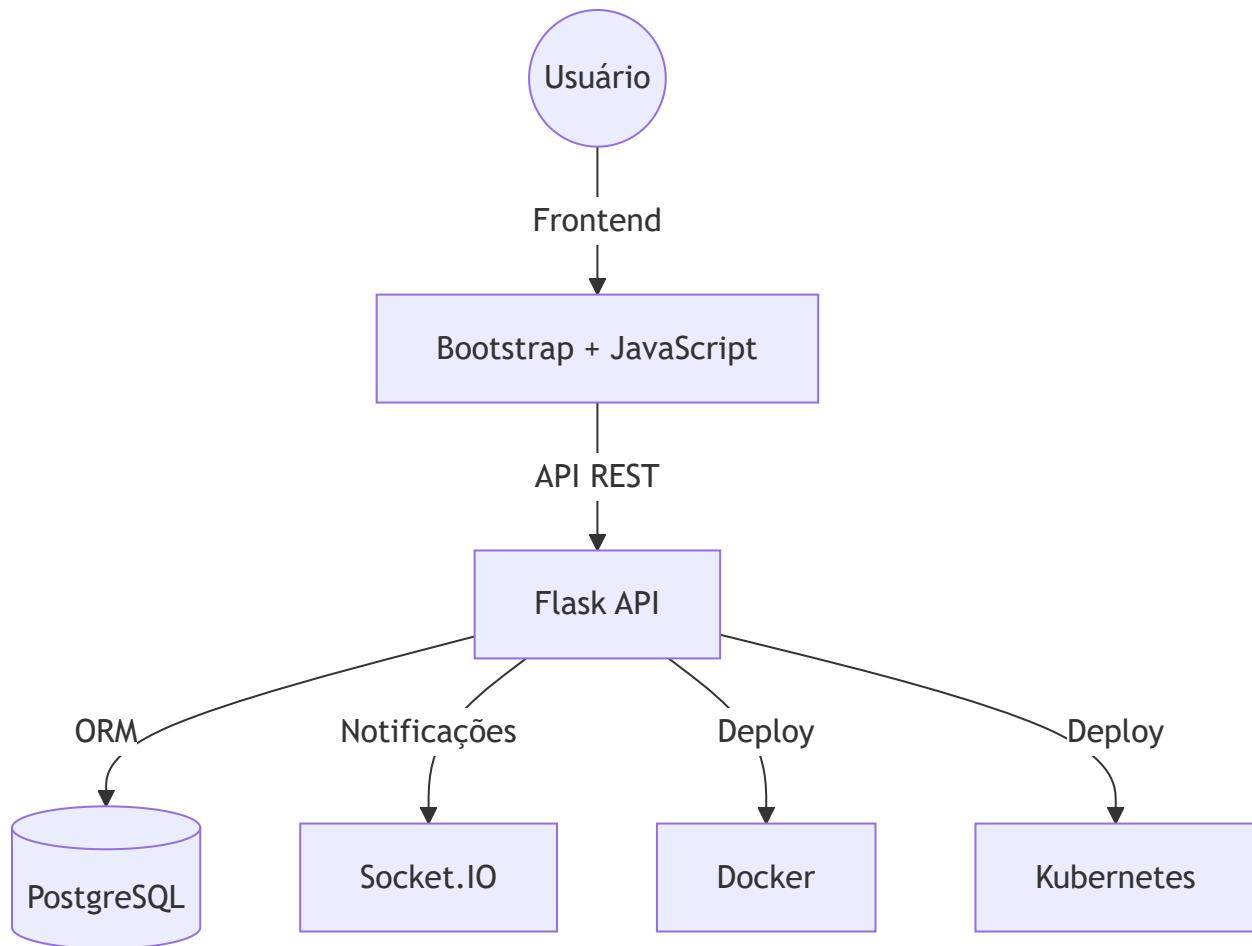
## Tecnologias Utilizadas

| Categoria   | Tecnologias                                    |
|-------------|------------------------------------------------|
| Backend     | Python, Flask, Flask-SQLAlchemy, Flask-Migrate |
| Frontend    | Bootstrap 5, JavaScript, CSS                   |
| Banco Dados | PostgreSQL, SQLAlchemy ORM, Alembic            |
| DevOps      | Docker, Kubernetes, GitLab CI/CD               |

## Pré-requisitos

- Python 3.11+
- PostgreSQL
- Navegador moderno (Chrome, Firefox, Edge)

# Arquitetura do Sistema



## Instalação

### 1. Clone o repositório

```
git clone https://github.com/diegobernardessv/RomaneiosApp.git  
cd RomaneiosApp
```

### 2. Instale as dependências

```
pip install -r requirements.txt
```

### 3. Configure as variáveis de ambiente

Crie um arquivo `.env` na raiz do projeto com:

```
DATABASE_URL=postgresql://usuario:senha@host:porta/banco  
SESSION_SECRET=sua_chave_secreta
```

#### 4. Initialize o banco de dados

```
flask db-init  
flask db migrate -m "Initial migration"  
flask db upgrade
```

#### 5. Execute o servidor

```
python main.py
```

Acesse: <http://localhost:5000>

## Estrutura do Projeto

```
RomaneiosApp/  
├── app/                      # Código fonte da aplicação  
│   ├── static/                # Arquivos estáticos (CSS, JS, imagens)  
│   ├── templates/             # Templates HTML  
│   ├── __init__.py            # Inicialização do app  
│   ├── *.py                   # Módulos principais  
├── kubernetes/              # Configurações Kubernetes  
├── migrations/               # Migrações do banco de dados  
├── tests/                    # Testes automatizados  
├── *.yml                     # Configurações CI/CD  
└── Dockerfile                # Configuração Docker  
└── requirements.txt          # Dependências Python
```

## Executando com Docker

```
docker network create net_romaneiosapp  
docker container run -ti -v$(pwd):/app --network net_romaneiosapp -w /app -p5000:5000 python:3.9  
docker container run -dti --rm --network net_romaneiosapp --name scr_db \  
-e POSTGRES_USER=romaneiosapp \  
-e POSTGRES_PASSWORD=wEaZ27t9HruMBT6c4jbVSd \  
-e POSTGRES_DB=romaneiosapp \  
-p5432:5432 postgres:16
```

# Licença

Este projeto está licenciado sob a licença [Apache License 2.0](#).

## Autor

-  **Diego Bernardes Silva**
-  GitHub: [diegobernardessv](https://github.com/diegobernardessv)
-  GitLab: [diegobernardessv](https://gitlab.com/diegobernardessv)
-  LinkedIn: [diegobernardessv](https://www.linkedin.com/in/diegobernardessv/)

## Melhorias Futuras

- Implementar testes automatizados
- Busca full-text otimizada
- Dashboard com métricas
- Melhorar responsividade mobile
- Sistema de auditoria
- Confirmações para ações críticas