Sistema de Administração de Condomínio

Descrição

Sistema completo para gestão de condomínios desenvolvido com Flask (Python) e frontend em HTML/CSS/JavaScript. O sistema possui dois níveis de acesso:

- 1. Área Pública (Visitantes): Visualização de informações gerais do condomínio
- 2. Área Administrativa: Painel completo para gerenciamento (requer login)

Funcionalidades

Área Pública

- · Dashboard com estatísticas gerais
- · Lista de condomínios cadastrados
- Boletos recentes
- Comunicados importantes

Área Administrativa

- · Login seguro com autenticação
- Dashboard administrativo
- Gerenciamento de condomínios
- · Gerenciamento de moradores
- · Controle de boletos
- Gestão de ocorrências
- · Publicação de comunicados

Tecnologias Utilizadas

- Backend: Python 3.11, Flask, SQLAlchemy
- · Banco de Dados: SQLite
- Frontend: HTML5, CSS3, JavaScript (ES6+)
- Autenticação: Flask Sessions com hash de senhas
- Estilização: CSS Grid, Flexbox, Font Awesome Icons

Instalação e Execução

Pré-requisitos

- Python 3.11 ou superior
- pip (gerenciador de pacotes Python)

Passos para instalação

1. Clone o repositório:

git clone <url-do-repositorio> cd condominium-admin

1. Crie e ative o ambiente virtual:

python -m venv venv source venv/bin/activate # Linux/Mac # ou venv\Scripts\activate # Windows

1. Instale as dependências:

pip install -r requirements.txt

1. Execute a aplicação:

python src/main.py

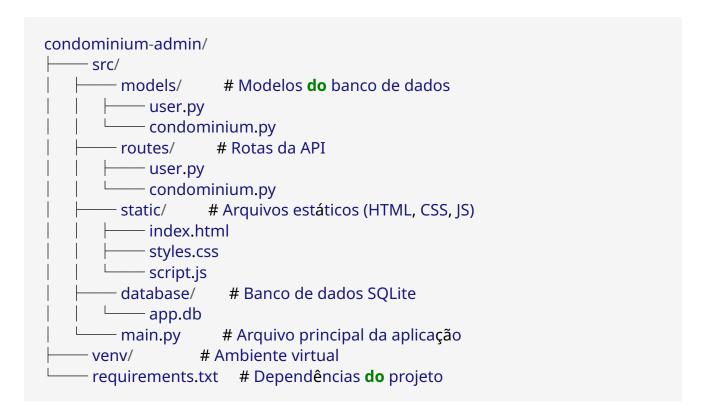
1. Acesse no navegador:

http://localhost:5000

Credenciais Padrão

Usuário: adminSenha: admin123

Estrutura do Projeto



API Endpoints

Públicos

- GET /api/dashboard Estatísticas gerais
- GET /api/condominiums Lista de condomínios
- GET /api/boletos/recent Boletos recentes
- GET /api/announcements Comunicados

Autenticação

- POST /api/login Login do administrador
- POST /api/logout Logout
- GET /api/check-auth Verificar autenticação

Administrativos (requerem autenticação)

- POST /api/admin/condominiums Criar condomínio
- POST /api/admin/residents Criar morador
- POST /api/admin/boletos Criar boleto
- POST /api/admin/announcements Criar comunicado
- POST /api/admin/occurrences Criar ocorrência

Modelo de Dados

Condomínio

• ID, Nome, Endereço

Morador

· ID, Nome, Unidade, Email, Telefone, ID do Condomínio

Boleto

 ID, Descrição, Valor, Data de Vencimento, Data de Pagamento, Status, ID do Morador

Ocorrência

• ID, Título, Descrição, Data de Registro, Status, ID do Condomínio, ID do Morador

Comunicado

· ID, Título, Conteúdo, Data de Publicação

Usuário (Administrador)

• ID, Username, Email, Hash da Senha

Recursos de Segurança

- Senhas armazenadas com hash seguro (Werkzeug)
- · Autenticação baseada em sessões
- Proteção de rotas administrativas
- Validação de dados de entrada
- · CORS configurado para desenvolvimento

Design Responsivo

O sistema foi desenvolvido com design responsivo, adaptando-se a diferentes tamanhos de tela:

- Desktop (1200px+)

- Tablet (768px 1199px)
- Mobile (até 767px)

Funcionalidades Futuras

- [] Geração automática de boletos recorrentes
- [] Sistema de notificações por email
- [] Relatórios em PDF
- [] Upload de arquivos/documentos
- [] Sistema de reserva de áreas comuns
- [] Integração com APIs de pagamento
- [] Aplicativo mobile

Contribuição

- 1. Faça um fork do projeto
- 2. Crie uma branch para sua feature (git checkout -b feature/AmazingFeature)
- 3. Commit suas mudanças (git commit -m 'Add some AmazingFeature')
- 4. Push para a branch (git push origin feature/AmazingFeature)
- 5. Abra um Pull Request

Licença

Este projeto está sob a licença MIT. Veja o arquivo LICENSE para mais detalhes.

Contato

Vander Baptista - vanderks300@gmail.com

Link do Projeto: https://github.com/vanderbaptista/condominium-admin