

Projeto SGHSS - Back-end (Escopo Essencial)

Aluno: Matheus

Enfase: Back-end (Python + Django REST Framework)

Introdução

O presente projeto tem como objetivo desenvolver a modelagem e a implementação de uma API Back-end para o Sistema de Gestão Hospitalar e de Serviços de Saúde (SGHSS), proposto pela instituição VidaPlus. O sistema foi idealizado para centralizar e integrar processos críticos de atendimento de pacientes, gestão de profissionais de saúde e administração hospitalar, incluindo funcionalidades de prontuários, agendamentos, consultas e telemedicina.

Escopo

Escopo essencial: Autenticação (JWT), CRUD de Pacientes e CRUD de Consultas (agendamento, listagem, visualização, cancelamento).

Requisitos Funcionais (seleção)

- RF001: Cadastro de pacientes (nome, CPF, data de nascimento, telefone)
- RF002: Listar pacientes
- RF003: Consultar paciente por id
- RF004: Atualizar paciente
- RF005: Excluir paciente
- RF006: Agendamento de consultas (associar paciente, data, horário)
- RF007: Listar consultas
- RF010: Autenticação com JWT

Requisitos Não Funcionais

- RNF001: Autenticação baseada em JWT
- RNF002: Senhas armazenadas em hash (Django)
- RNF003: Padrão REST para endpoints
- RNF004: Logs de atividades e erros
- RNF006: Integridade relacional entre Pacientes e Consultas
- RNF008: Conformidade básica com LGPD

Modelagem e Arquitetura

- Atores: Usuário autenticado, Paciente, Sistema SGHSS.
- Diagrama de classes e DER incluídos conceitualmente no anexo.
- Estrutura de apps no Django: usuarios, pacientes, consultas.

Endpoints principais

- POST /api/auth/login/ -> Gera tokens (access, refresh)
- GET/POST /api/pacientes/
- GET/PUT/DELETE /api/pacientes/{id}/
- GET/POST /api/consultas/
- POST /api/consultas/{id}/cancelar/

Implementação

A implementação segue o padrão Django REST Framework com ViewSets e Routers. Arquivo de settings com configuração do SimpleJWT e AUTH_USER_MODEL.

Plano de testes (resumo)

- Testes funcionais com Postman (cobertura mínima): criar usuário admin, autenticar, criar paciente, agendar consulta, cancelar.
- Verificar respostas HTTP, códigos de erro, proteção de endpoints (401 sem token).
- Testes básicos de integração: migrar DB e executar scripts de criação.

Conclusão

Este projeto entrega uma API mínima mas consistente do SGHSS, suficiente para demonstrar habilidades em modelagem, segurança (autenticação) e implementação RESTful com Django.

Anexos

- Código fonte do backend (pasta sghss_project)
- Trechos de settings, models, serializers, views e urls.