

1) Requisitos Funcionais

- RF01: Visualizar lista de unidades de saúde e status de lotação (semáforo).
- RF02: Atualizar número de pacientes em espera por unidade.
- RF03: Cadastro e autenticação de pacientes (registro/login).
- RF04: Interface de administração via Django admin para gerenciar unidades e pacientes.

2) Requisitos Não Funcionais

- RNF01: Resposta do servidor em < 500ms (cenário local de apresentação).
- RNF02: Sistema executável em ambiente Windows (PowerShell) com Python 3.12.
- RNF03: Testes automatizados executáveis via `pytest -v`.

3) Requisitos de Dados

- RD01: Banco de produção MySQL 8.0+ (DB_NAME: filazero_db).
- RD02: Arquivo de ambiente `.env` contendo DB credentials e SECRET_KEY.
- RD03: Dados iniciais fornecidos por `python manage.py seed_db`.

4) Requisitos Técnicos / Infra

- RT01: Python 3.12+ instalado e ativo no PATH.
- RT02: Dependências instaladas com `pip install -r requirements.txt`.
- RT03: MySQL rodando localmente e acessível nas credenciais do `.env`.

5) Requisitos de Testes

- RTE01: Rodar `pytest -v` deve retornar todos os testes passando.
 - RTE02: Testes chave: `core_app/test/test_db.py`, `core_app/test/test_cpf_db.py`, `core_app/test/test_rotas.py`.
 - RAP01: Todos os comandos do README devem executar sem erro em ordem (clone → venv → pip → init_db → migrate → seed_db → runserver).
 - RAP02: Fornecer usuário admin via `createsuperuser` para demonstração do admin.
 - RAP03: Garantir que o servidor rode em http://127.0.0.1:8000/ durante a apresentação.
- RAP01: Todos os comandos do README devem executar sem erro em ordem (clone → venv → pip → init_db → migrate → seed_db → runserver).
- RAP02: Fornecer usuario admin via createsuperuser para demonstracao do admin.
- RAP03: Garantir que o servidor rode em http://127.0.0.1:8000/ durante a apresentacao.
