

# Documentação da API - Relatório

Este relatório visa proporcionar uma visão abrangente dos principais aspectos e funcionalidades da API desenvolvida para o gerenciamento de pacientes, médicos e consultas.

## Informações do Desenvolvedor

- **Nome:** Jorge Bruno Costa Alves
- **Matrícula:** 509718
- **Disciplina:** Sistemas Distribuídos
- **Universidade:** Universidade Federal de Quixadá

## Introdução

A API foi concebida para oferecer funcionalidades eficientes no contexto do gerenciamento de informações de pacientes, médicos e consultas. Utilizando Flask como framework, a aplicação segue um modelo RESTful e se comunica utilizando o formato JSON.

## Endpoints e Funcionalidades

### Pacientes

#### 1. Listar Pacientes

- **Endpoint:** `/pacientes/`
- **Métodos:** `GET`
- **Descrição:** Retorna a lista de todos os pacientes.

#### 2. Adicionar Paciente

- **Endpoint:** `/pacientes/`
- **Métodos:** `POST`
- **Descrição:** Adiciona um novo paciente.
- **Parâmetros:**
  - `nome` (string): Nome do paciente.
  - `cpf` (string): CPF do paciente.

#### 3. Consultar Paciente

- **Endpoint:** `/pacientes/<cpf>/`
- **Métodos:** `GET`
- **Descrição:** Retorna os detalhes de um paciente específico.
- **Parâmetros:**
  - `cpf` (string): CPF do paciente.

#### 4. Deletar Paciente

- **Endpoint:** `/pacientes/<cpf>/`
- **Métodos:** `DELETE`
- **Descrição:** Remove um paciente com base no CPF.
- **Parâmetros:**
  - `cpf` (string): CPF do paciente.

### Médicos

#### 1. Listar Médicos

- **Endpoint:** `/medicos/`
- **Métodos:** `GET`
- **Descrição:** Retorna a lista de todos os médicos.
- **Parâmetros:**
  - `especialidade` (opcional): Filtra os médicos por especialidade.

#### 2. Adicionar Médico

- **Endpoint:** `/medicos/`
- **Métodos:** `POST`
- **Descrição:** Adiciona um novo médico.
- **Parâmetros:**
  - `nome` (string): Nome do médico.
  - `cpf` (string): CPF do médico.
  - `especialidade` (string): Especialidade do médico (0 - Oftalmologista, 1 - Patologista, 2 - Clínico Geral).

#### 3. Consultar Médico

- **Endpoint:** `/medicos/<cpf>/`
- **Métodos:** `GET`
- **Descrição:** Retorna os detalhes de um médico específico.
- **Parâmetros:**
  - `cpf` (string): CPF do médico.

#### 4. Deletar Médico

- **Endpoint:** `/medicos/<cpf>/`
- **Métodos:** `DELETE`
- **Descrição:** Remove um médico com base no CPF.
- **Parâmetros:**
  - `cpf` (string): CPF do médico.

## Consultas

### 1. Listar Consultas

- **Endpoint:** `/consultas/`
- **Métodos:** `GET`
- **Descrição:** Retorna a lista de todas as consultas.
- **Parâmetros:**
  - `medico_cpf` (opcional): Filtra as consultas por CPF do médico.
  - `paciente_cpf` (opcional): Filtra as consultas por CPF do paciente.

### 2. Marcar Consulta

- **Endpoint:** `/consultas/`
- **Métodos:** `POST`
- **Descrição:** Marca uma nova consulta.
- **Parâmetros:**
  - `paciente_cpf` (string): CPF do paciente.
  - `medico_cpf` (string): CPF do médico.
  - `data` (string): Data da consulta (formato: DD-MM-YYYY).

### 3. Atualizar Consulta

- **Endpoint:** `/consultas/`
- **Métodos:** `PUT`
- **Descrição:** Atualiza uma consulta existente.
- **Parâmetros:**
  - `paciente_cpf` (string): CPF do paciente.
  - `medico_cpf` (string): CPF do médico.
  - `data` (string): Data da consulta (formato: DD-MM-YYYY).

### 4. Deletar Consulta

- **Endpoint:** `/consultas/`
- **Métodos:** `DELETE`
- **Descrição:** Remove uma consulta.
- **Parâmetros:**
  - `paciente_cpf` (string): CPF do paciente.
  - `medico_cpf` (string): CPF do médico.
  - `data` (string): Data da consulta (formato: DD-MM-YYYY).

## Especialidades

### 1. Listar Especialidades

- **Endpoint:** `/especialidades/`
- **Métodos:** `GET`
- **Descrição:** Retorna a lista de especialidades disponíveis.

## Considerações Finais

Esta documentação fornece uma visão detalhada dos endpoints disponíveis na API, destacando as operações permitidas e os parâmetros necessários para cada requisição.