

## API Sistema de Agendamento de Consultas Médicas

### Objetivo:

Construir uma API para gerenciar agendamentos de consultas em uma clínica médica. A API permitirá que pacientes agendem consultas, que médicos visualizem suas agendas e que a clínica controle a disponibilidade de horários.

### Estrutura:

- **Introdução:**
  - Explicação do Spring Boot e como ele pode ser usado para criar sistemas de agendamento de consultas médicas.
  - Visão geral sobre o sistemas de agendamento de consultas médicas.
- **Estrutura do Banco de Dados:**
  - Entidades: Paciente, Médico, Consulta, Especialidade e Prontuário.
  - Relacionamento entre as entidades (exemplo: Paciente agendando uma Consulta com um Médico de uma Especialidade específica).
- **Desenvolvimento da API:**
  - **Controladores:** Endpoints para pacientes, médicos, consultas e prontuários.
  - **Serviços:** Regras de negócios, como verificar disponibilidade de horários e validar especialidades.
  - **Repositórios:** Acesso aos dados utilizando Spring Data JPA.
- **Regra de Negócio:**
  - Validação de Agendamento:
  - Impedir agendamentos fora do horário comercial ou em datas passadas.
  - Verificar se o médico está disponível no horário solicitado.
  - Pacientes podem cancelar até 24h antes da consulta.

### Exemplo de Endpoints:

- POST /pacientes: Cadastrar um novo paciente.
- POST /medicos: Registrar um novo médico.
- POST /consultas: Agendar uma nova consulta.
- GET /consultas/{id}: Consultar os detalhes de uma consulta.

## Descrição Técnica:

### Entidades:

#### 1. Paciente

- id (Long): Identificador único.
- nome (String): Nome completo.
- cpf (String): Documento (único).
- telefone (String): Contato.
- email (String): Para notificações.

#### 2. Médico

- id (Long): Identificador único.
- nome (String): Nome completo.
- crm (String): Registro profissional (único).
- especialidadeId (Long): Chave estrangeira para Especialidade.

#### 3. Especialidade

- id (Long): Identificador único.
- nome (String): Ex: "Cardiologia", "Pediatria".

#### 4. Consulta

- id (Long): Identificador único.
- pacienteId (Long): Chave estrangeira para Paciente.
- medicoId (Long): Chave estrangeira para Médico.
- dataHora (LocalDateTime): Data e hora agendadas.
- status (String): "Agendada", "Cancelada", "Concluída".
- observacoes (String): Detalhes opcionais.

#### 5. Disponibilidade

- id (Long): Identificador único.
- medicoId (Long): Chave estrangeira para Médico.
- diaSemana (String): Ex: "Segunda-feira".
- horarioInicio (LocalTime): Ex: "09:00".
- horarioFim (LocalTime): Ex: "17:00".

## **Endpoints:**

### **Paciente**

#### **1 - Registrar Paciente:**

- Método: POST
- Endpoint: /pacientes
- Descrição: Cadastra um novo paciente.
- Campos no Corpo da Requisição: nome, cpf, telefone, email.

#### **2 - Listar Pacientes:**

- Método: GET
- Endpoint: /pacientes
- Descrição: Lista todos os pacientes cadastrados.
- Parâmetros opcionais: page, size.

#### **3 - Buscar Paciente:**

- Método: GET
- Endpoint: /pacientes/{id}
- Descrição: Retorna os dados de um paciente específico.
- Parâmetro obrigatório: id.

#### **4 - Atualizar Paciente:**

- Método: PUT
- Endpoint: /pacientes/{id}
- Descrição: Atualiza os dados de um paciente.
- Campos atualizáveis: nome, telefone, email.

#### **5 - Remover Paciente:**

- Método: DELETE
- Endpoint: /pacientes/{id}
- Descrição: Exclui um paciente.
- Restrição: Só é permitido se não houver consultas agendadas.

### **Médico**

#### **1 - Registrar Médico:**

- Método: POST

- Endpoint: /medicos
- Descrição: Cadastra um novo médico.
- Campos no Corpo da Requisição: nome, crm, especialidadeId.

## **2 - Listar Médicos:**

- Método: GET
- Endpoint: /medicos
- Descrição: Lista todos os médicos cadastrados.
- Filtros opcionais: especialidade, disponível.

## **3 - Buscar Médico:**

- Método: GET
- Endpoint: /medicos/{id}
- Descrição: Retorna os dados de um médico específico.

## **4 - Atualizar Médico:**

- Método: PUT
- Endpoint: /medicos/{id}
- Descrição: Atualiza os dados de um médico.

## **5 - Remover Médico:**

- Método: DELETE
- Endpoint: /medicos/{id}
- Descrição: Exclui um médico.
- Restrição: Só é permitido se não houver consultas futuras.

## **Consulta**

### **1 - Agendar Consulta:**

- Método: POST
- Endpoint: /consultas
- Descrição: Agenda uma nova consulta.
- Campos obrigatórios: pacienteId, medicoId, dataHora.
- Validações: Conflito de horários; Horário dentro da disponibilidade do médico.

### **2 - Listar Consultas:**

- Método: GET

- Endpoint: /consultas
- Descrição: Lista todas as consultas agendadas.
- Filtros opcionais: medico, paciente, status, dataInicio, dataFim.

### **3 - Buscar Consulta:**

- Método: GET
- Endpoint: /consultas/{id}
- Descrição: Retorna os dados de uma consulta específica.

### **4 - Atualizar Status de Consulta:**

- Método: PATCH
- Endpoint: /consultas/{id}/status
- Descrição: Atualiza o status de uma consulta.
- Valores aceitos: Agendada, Cancelada, Concluída.

### **5 - Cancelar Consulta:**

- Método: DELETE
- Endpoint: /consultas/{id}
- Descrição: Cancela uma consulta.
- Restrição: Só é permitido até 24h antes.

## **Especialidade**

### **1 - Registrar Especialidade:**

- Método: POST
- Endpoint: /especialidades
- Descrição: Cadastra uma nova especialidade.
- Campo obrigatório: nome.

### **2 - Listar Especialidades:**

- Método: GET
- Endpoint: /especialidades
- Descrição: Lista todas as especialidades cadastradas.

### **3 - Buscar Especialidade:**

- Método: GET
- Endpoint: /especialidades/{id}

- Descrição: Retorna os dados de uma especialidade específica.

#### **4 - Atualizar Especialidade:**

- Método: PUT
- Endpoint: /especialidades/{id}
- Descrição: Atualiza o nome de uma especialidade.

#### **5 - Remover Especialidade:**

- Método: DELETE
- Endpoint: /especialidades/{id}
- Descrição: Exclui uma especialidade.
- Restrição: Só é permitido se não houver médicos vinculados.

### **Disponibilidade**

#### **1 - Registrar Disponibilidade:**

- Método: POST
- Endpoint: /medicos/{medicoid}/disponibilidades
- Descrição: Registra horários disponíveis para um médico.
- Campos obrigatórios: diaSemana, horarioInicio, horarioFim.

#### **2 - Listar Disponibilidades:**

- Método: GET
- Endpoint: /medicos/{medicoid}/disponibilidades
- Descrição: Lista as disponibilidades de um médico.

#### **3 - Remover Disponibilidade:**

- Método: DELETE
- Endpoint: /disponibilidades/{id}
- Descrição: Exclui um horário de disponibilidade.