

# Sistema Médico Full-Stack para Gestão de Consultório e Hospital

---

Este sistema é uma solução completa e integrada para a gestão de consultórios e hospitais, cobrindo a administração de médicos e pacientes, bem como funcionalidades avançadas para ambientes hospitalares.

## Status do Projeto

Estamos começando o desenvolvimento deste projeto agora. A estrutura básica está sendo configurada e as principais funcionalidades estão sendo implementadas.

## Tecnologias Utilizadas

### Front-end

- **ReactJS:** Biblioteca para criar a interface do usuário.
- **TypeScript:** Adiciona tipagem estática ao código React.
- **Next.js:** Utilizado para renderização no lado do servidor (Server-Side Rendering), melhorando a performance e SEO.
- **PrimeReact:** Biblioteca de componentes de UI para criar elementos visuais sofisticados e interativos.
- **React Router:** Implementação de rotas protegidas, garantindo que apenas usuários autenticados possam acessar determinadas páginas.
- **AuthGuard:** Protege rotas sensíveis e controla permissões (administradores gerenciam médicos, médicos não podem modificar suas próprias informações).

### Back-end

- **Java + Spring Boot:** Framework robusto para a criação de APIs RESTful e sistemas de grande escala.
- **Spring Data JPA:** Gerenciamento de dados e integração com bancos de dados relacionais.
- **Spring Security + JWT:** Implementação de segurança e controle de acesso com autenticação baseada em tokens JWT.
- **Node.js + Express.js:** Utilizado para criação de serviços rápidos e escaláveis.
- **Sequelize ORM:** ORM utilizado em conjunto com Express.js para interação com bancos de dados relacionais.

### Banco de Dados

- **MySQL:** Armazenamento de dados estruturados como informações de pacientes, médicos e consultas.
- **PostgreSQL:** Funcionalidades complexas e escaláveis, como gestão de consultas, exames e prontuários eletrônicos.

### Docker

- **Docker:** Containeriza a aplicação, garantindo consistência em todos os ambientes de desenvolvimento, testes e produção.
- **Docker Compose:** Orquestração de múltiplos containers para criar ambientes isolados.

## Testes

- **JUnit:** Testes unitários no back-end com Java.
- **Jest:** Testes unitários para serviços implementados com Node.js.
- **Cypress:** Testes end-to-end para garantir que o fluxo completo da aplicação funcione conforme o esperado.

## Ferramentas de Desenvolvimento e Testes

- **Postman:** Teste manual das APIs.
- **Insomnia:** Alternativa ao Postman para testes de APIs.
- **Cypress:** Testes end-to-end.

# Funcionalidades

## Funcionalidades Gerais

1. **Gestão de Médicos (Apenas Administradores)**
  - CRUD completo de médicos.
2. **Gestão de Pacientes**
  - CRUD completo de pacientes.
3. **Gestão de Consultas**
  - Médicos podem criar, editar e visualizar consultas.
4. **Sistema de Autenticação**
  - JWT + AuthGuard para login e controle de acesso.
5. **Dashboard**
  - Exibição de estatísticas gerais e informações importantes.

## Funcionalidades Adicionais

1. **Gestão de Agendamento de Consultas**
  - Agendamento online e confirmação automática.
2. **Gestão de Exames e Resultados**
  - Solicitação, upload e visualização de resultados.
3. **Prontuário Eletrônico Completo (EHR)**
  - Histórico clínico completo e acesso por médicos autorizados.
4. **Telemedicina**
  - Consultas virtuais e anotações digitais.
5. **Gestão de Leitos**
  - Controle de ocupação e alta médica.
6. **Alertas e Notificações**
  - Notificações automáticas sobre consultas e exames.
7. **Sistema de Prescrição Eletrônica**
  - Receitas digitais e histórico de prescrições.

## 8. Painel de Controle para Administradores

- Relatórios de atendimentos e gestão de equipe.

## 9. Sistema de Faturamento e Financeiro

- Controle de cobranças e integração com planos de saúde.

## 10. Gestão de Emergências

- Registro e acompanhamento de atendimentos emergenciais.

# Comandos de Inicialização

## Front-end

### 1. Instalação das dependências:

```
cd front-end  
npm install
```

### 2. Inicializar o servidor de desenvolvimento

```
npm run dev
```