Documentação Completa do Projeto - Sistema de Agendamento Doctor Center Med

Gerado em: 30/05/2025 11:28

Este projeto é um sistema completo de agendamento para uma clínica médica, com integração ao WhatsApp, painel visual estilo Google Calendar, e um backend robusto em FastAPI. A aplicação é voltada para operação local em rede interna da clínica.

# Tecnologias Utilizadas

Frontend:  
- React + Vite  
- React Router DOM  
- FullCalendar  
- TailwindCSS (via classes CSS customizadas)

Backend:  
- FastAPI (Python)  
- SQLAlchemy + SQLite  
- Uvicorn

Outros:  
- Integração com WhatsApp (simulada via JSON)  
- Estilo responsivo e moderno baseado em design de sistemas

# Estrutura de Pastas

Projeto/  
|-- backend/  
| |-- main.py  
| |-- database.py  
| |-- models.py  
| |-- whatsapp\_db.py  
| |-- routers/  
| |-- agendamentos.py  
| |-- consultas.py  
| |-- medicos.py  
| |-- pacientes.py  
| |-- whatsapp\_bot.py  
|  
|-- frontend/  
 |-- src/  
 |-- App.jsx  
 |-- main.jsx  
 |-- app.css  
 |-- components/  
 |-- MenuPrincipal.jsx  
 |-- CalendarView.jsx

# Funcionalidades

Agenda (CalendarView):  
- Visualização semanal com slots de 15 em 15 minutos  
- Cores para status da consulta: agendada, confirmada, cancelada  
- Clique em slot abre modal de agendamento  
- Clique em evento mostra alerta com dados e opção de exclusão  
- Exportação da agenda para CSV  
- Filtro por especialidade  
  
Menu Principal:  
- Interface moderna com fundo gradiente  
- Botões verticais grandes com alta legibilidade  
- Direciona para agenda e futuras funcionalidades  
  
Integração WhatsApp:  
- Função simulada: gerar\_resposta(mensagem)  
- Retorna JSON com mensagem e botões de opção para o paciente

# Backend (FastAPI)

- main.py: inicializa a aplicação, inclui rotas e CORS  
- database.py: configura SQLAlchemy com SQLite  
- models.py: define a tabela Consulta  
- whatsapp\_db.py: lógica de resposta do bot  
- Rotas (em routers/): implementam endpoints REST  
  
Endpoints principais:  
- GET /consultas: lista todas as consultas  
- POST /agendamentos/manual: cria nova consulta  
- DELETE /consultas/{id}: exclui uma consulta  
- POST /whatsapp/simular: simula interação via WhatsApp

# Execução

Backend:  
cd backend  
uvicorn main:app --reload  
  
Frontend:  
cd frontend  
npm install  
npm run dev  
  
Aplicação disponível em: http://localhost:5173

# Considerações Finais

- Sistema pensado para uso offline, mas adaptável para nuvem  
- Estrutura modular, facilitando expansão para outras áreas  
- Estilo visual personalizável com foco em usabilidade para recepcionistas  
- Planeja-se futura integração com API do WhatsApp oficial da Meta

# Sugestões de Expansão

- Controle de presença  
- Confirmação automática via WhatsApp  
- Painel administrativo para relatórios  
- Login de usuários com permissões (admin, recepção, médico)  
- Integração com Google Calendar ou sistemas de pagamento