

Especificação do Sistema - AtendeAí 2.0

Visão Geral

- **Projeto:** AtendeAí 2.0
- **Versão:** 2.0.0
- **Última Atualização:** 2024-12-19

Arquitetura

Backend Services

- **whatsapp-service** (porta 3001) - Integração WhatsApp
- **conversation-service** (porta 3002) - Gestão de conversas
- **clinic-service** (porta 3003) - Gestão de clínicas
- **appointment-service** (porta 3004) - Agendamentos
- **auth-service** (porta 3005) - Autenticação
- **user-service** (porta 3006) - Gestão de usuários

Frontend

- **Framework:** React + Vite
- **Calendário:** Próprio (sem integração Google)
- **Agenda:** Tela dedicada com calendário interno

Database

- **Provider:** Supabase
- **Migrações:** Aplicadas

Deployment

- **Platform:** Railway
- **Config Status:** Configurado

Features Principais

1. Sistema de Autenticação

- Login/logout de usuários
- Gestão de sessões
- Controle de acesso

2. Gestão de Clínicas

- Criação/edição de clínicas
- Configuração de WhatsApp
- Contexto personalizado

3. Agendamento de Consultas

- **Calendário Próprio:** Sem integração Google
- **Tela de Agenda:** Interface dedicada
- **Funcionalidades:**
 - Visualização mensal/semanal/diária
 - Criação de agendamentos
 - Edição de horários
 - Cancelamento de consultas
 - Notificações via WhatsApp

4. Integração WhatsApp

- Recebimento de mensagens
- Envio de respostas automáticas
- Notificações de agendamento
- Atribuição manual de conversas

5. Sistema de Conversas

- Interface de chat
- Atribuição automática/manual
- Histórico de mensagens
- Status de conversas

Funcionalidades Pendentes

1. Envio de Mensagens (Conversations)

- **Localização:** src/pages/Conversations.tsx
- **Status:** Interface pronta, API pendente
- **Endpoint:** POST /api/conversations/:id/messages
- **Payload:** { message: string, sender: string }

2. Atribuição Manual (Conversations)

- **Localização:** src/pages/Conversations.tsx
- **Status:** Interface pronta, API pendente
- **Endpoint:** PUT /api/conversations/:id/assign
- **Payload:** { assigned_user_id: string, mode: 'manual'|'auto' }

3. Criação/Edição de Clínicas

- **Localização:** src/pages/Clinics.tsx
- **Status:** Formulários prontos, API pendente
- **Endpoints:**
 - POST /api/clinics (criar)
 - PUT /api/clinics/:id (editar)
- **Payload:** { name, whatsapp_number, context_json, etc. }

4. Autenticação

- **Localização:** src/hooks/useAuth.tsx
- **Status:** Estrutura pronta, integração pendente
- **Endpoint:** POST /api/auth/login
- **Payload:** { login, password }

5. Calendário de Agenda

- **Localização:** src/pages/Agenda.tsx
- **Status:** Interface pronta, funcionalidades pendentes
- **Funcionalidades:**
 - Visualização de calendário
 - Criação de agendamentos
 - Edição de horários
 - Integração com appointment-service

Especificações Técnicas

Calendário Próprio

- **Sem integração Google Calendar**
- **Interface nativa** na tela de Agenda
- **Funcionalidades:**
 - Visualização mensal/semanal/diária
 - Drag & drop para agendamentos
 - Filtros por clínica/profissional
 - Exportação de relatórios

Integrações Removidas

- Google Calendar API
- Google OAuth
- Qualquer serviço Google

Integrações Mantidas

- WhatsApp Business API
- Supabase (banco de dados)
- Railway (deploy)

Próximos Passos

1. Remover serviço Google Calendar do backend
2. Implementar calendário próprio na tela de Agenda
3. Implementar chamadas reais para APIs
4. Testar integração com backend
5. Validar funcionalidades end-to-end