

Guia de Implementação: Sistema de Multiagentes para Clínicas

Bem-vindo(a)!

Este documento é o seu guia completo para configurar e colocar em produção o seu sistema de multiagentes para WhatsApp voltado para clínicas (odontológicas, médicas, estéticas, etc.). Siga atentamente cada passo para garantir que todas as integrações funcionem corretamente.

Visão Geral do Projeto

O sistema utiliza Inteligência Artificial para oferecer um atendimento humanizado e eficiente 24/7, capaz de qualificar pacientes, tirar dúvidas, apresentar procedimentos com mídias (imagens/vídeos), agendar consultas em múltiplas agendas e enviar lembretes automáticos. Ele é construído na plataforma n8n e dividido em 3 workflows principais:

1. **CONSOLE DA CLÍNICA (CLINICA CONSOLE - TEMPLATE)**: O cérebro da operação, onde residem os múltiplos agentes de IA (recepcionista, agendador, especialista em procedimentos, etc.) e suas ferramentas.
2. **ENTRADA E SAÍDA (ENTRADA E SAIDA MODELO)**: A ponte de comunicação que recebe mensagens do WhatsApp (incluindo textos, áudios, imagens), trata esses dados e envia as respostas da IA de volta para o paciente.
3. **LEMBRETES DE CONSULTAS (LEMBRETES DE CONSULTAS MULTI AGENDAS)**: O robô que verifica as agendas e envia lembretes automáticos aos pacientes um dia antes e no dia da consulta.

1. Pré-requisitos: Contas e Serviços

Antes de importar e configurar os workflows, prepare todas as contas e serviços externos necessários:

Serviço	Função no Fluxo	O que você vai precisar
n8n	Plataforma central que orquestra toda a automação.	Recomenda-se a instalação Self-Hosted robusta, Link com Desconto: https://hostinger.com.br/?REFERRALCODE=NSOTHALENTF2)

Evolution API	Gateway de comunicação com o WhatsApp para receber e enviar mensagens.	Uma instância ativa conectada a um número de WhatsApp. Você precisará do Domínio/URL, Nome da Instância e API Key .
Supabase	Banco de dados principal (Postgres) para armazenamento persistente de longo prazo.	Uma conta no Supabase (plano gratuito é suficiente). Usado para armazenar: chats (sessões de conversa), chat_messages (histórico detalhado). Você precisará da URL e API Key .
OpenAI	Fornecer os modelos de inteligência artificial (cérebro dos agentes).	Uma conta na OpenAI com método de pagamento ativo para uso da API. Você precisará da sua API Key . Usaremos modelos como GPT-4.1-mini ou similar.
Redis	Memória cache de alta velocidade para armazenamento de curto prazo da conversa e reconstrução de mensagens.	Uma instância do Redis (geralmente disponível na sua hospedagem). Você precisará do Host, Porta, Usuário (se houver) e Senha .
Google Calendar	Ferramenta de agendamento de consultas.	Uma conta Google com as agendas dos profissionais criadas. Você precisará criar uma Credencial OAuth2 no n8n para permitir o acesso às agendas.
n8n Data Tables	Banco de dados interno do n8n para armazenar informações dos procedimentos (links de mídias, legendas). Atualize o N8N para versões a partir da 1.114.3.	Funcionalidade nativa do n8n (a partir da 1.114.3). Não requer credenciais externas, apenas configuração interna.

Nota: O Supabase utiliza um banco de dados Postgres. Você pode usar o mesmo banco de dados para a memória de longo prazo da IA (**n8n_chat_histories**), centralizando os dados.

1.2. Conhecimento Técnico Recomendado

Embora o template seja detalhado, um conhecimento básico em APIs, JSON e na interface do n8n facilitará a configuração.

2. Passo 1: Preparando a Tabela de Procedimentos (n8n Data Tables)

Esta etapa configura a base de dados interna do n8n que o agente `agente_atende_qualifica` usará para buscar mídias (imagens/vídeos) dos procedimentos.


1. **Acesse Data Tables no n8n:** No menu lateral do n8n, clique em "Data Tables".
2. **Crie a Tabela:**
 - Clique em "Create table".
 - Dê um nome à tabela, por exemplo: `Procedimentos`.
3. **Defina as Colunas:**
 - Adicione as seguintes colunas (os nomes devem ser **exatos**):
 1. `nomeProcedimento` (Tipo: String) - *Este nome deve corresponder exatamente aos nomes listados no prompt do agente_atende_qualifica.*
 2. `urlMidia` (Tipo: String) - *Link direto para a imagem ou vídeo. Veja como gerar o link abaixo.*
 3. `legenda` (Tipo: String) - *Texto descritivo que acompanhará a mídia.*
4. **Popule a Tabela:**
 - Adicione uma linha para cada procedimento que você deseja que a IA possa mostrar. Preencha o `nomeProcedimento` (ex: `Clareamento Dental`), a `legenda` atrativa e a `urlMidia` correspondente.
 - **Como gerar o link `urlMidia` a partir do Google Drive:**
 1. **Faça o upload** da sua imagem ou vídeo para uma pasta no Google Drive.
 2. Clique com o botão direito no arquivo e selecione "**Compartilhar**" > "**Compartilhar**".
 3. Em "Acesso geral", mude para "**Qualquer pessoa com o link**" e certifique-se de que a permissão seja "**Leitor**".
 4. Clique em "Copiar link". Você terá um link parecido com este:
`https://drive.google.com/file/d/1rmjiO-njtMOAtwNagq2K5AMFT2InmoJO/view?usp=drive_link`
 5. **Identifique o ID do arquivo:** É a longa sequência de caracteres entre `/d/` e `/view?`. No exemplo acima, o ID é `1rmjiO-njtMOAtwNagq2K5AMFT2InmoJO`.
 6. Construa o link direto: Use o seguinte formato, substituindo `SEU_ID_AQUI` pelo ID que você copiou:
`https://drive.usercontent.google.com/download?id=SEU_ID_AQUI&export=view`
 7. **Cole este link final** na coluna `urlMidia` da sua Data Table no n8n. Este formato permite que a Evolution API (e o WhatsApp) acessem e exibam a mídia diretamente.

3. Passo 2: Preparando o Banco de Dados (Execução Única)

Agora, criaremos as tabelas essenciais no Supabase/Postgres para armazenar os dados das conversas.

- **Workflow a ser utilizado:** CLINICA CONSOLE - TEMPLATE

Dentro deste workflow, localize a seção com os nós de setup inicial:

1. **Conecte sua Credencial Postgres/Supabase:**
 - Nos nós 1 - **criaTabelaChatMessages** e 2 - **criaTabelaChats**, clique sobre eles e selecione sua credencial de acesso ao banco de dados no painel à direita.
2. **Execute na Ordem Correta:**
 - Execute os seguintes nós manualmente (clcando no ícone  sobre cada um) e nesta ordem exata:
 1. 1 - **criaTabelaChatMessages**: Cria a tabela para o histórico detalhado das mensagens.
 2. 2 - **criaTabelaChats**: Cria a tabela para gerenciar as sessões de conversa e status da IA (pausada/ativa).

4. Passo 3: Configurando o Console da Clínica (A Inteligência)

Este é o workflow principal onde a IA toma decisões e utiliza suas ferramentas.

- **Workflow a ser utilizado:** CLINICA CONSOLE - TEMPLATE.json
1. **Conecte os Componentes Centrais:**
 - Nó **openai4.1Mini**: **CRÍTICO!** Selecione sua credencial da OpenAI.
 - Nó **Postgres Chat Memory**: **CRÍTICO!** Selecione sua credencial do Postgres (onde a tabela **n8n_chat_histories** está) e confirme o nome da tabela.
 2. **Configure as Ferramentas dos Agentes:**
 - Nó **buscar_midia_procedimento** (Data Table Tool): **CRÍTICO!** Selecione a tabela de dados (**Procedimentos**) que você criou no Passo 2. Verifique se o **keyName** (**nomeProcedimento**) e a **condition** (**like**) estão corretos.
 - Nós do **Google Calendar** (**verificar_disponibilidade_agenda**, **criar_consulta**, **atualizar_consulta**, **cancelar_consulta**): **CRÍTICO!** Selecione sua credencial do Google Calendar. *Não é necessário configurar o ID da agenda diretamente nestes nós*, pois eles recebem essa informação dinamicamente do agente agendador.
 - Nó **encaminhar_para_atendente_humano** (HTTP Request Tool): **CRÍTICO!**

- Copie a URL de produção (Production URL) do nó `webhookRecebeTransferencia` (que está *neste mesmo* workflow).
 - Cole essa URL no campo `URL` deste nó `encaminhar_para_atendente_humano`. Isso cria o link para a transferência interna.
3. **Configure os IDs das Agendas dos Profissionais:**
- **CRÍTICO!** A IA precisa saber qual agenda do Google Calendar pertence a qual profissional/especialidade. Essa configuração é feita dentro do prompt do agente agendador.
 - Localize e clique no nó `agente_agendador` (AgentTool).
 - No painel de configurações à direita, vá para a aba "Options" e localize o campo `System Message`.
 - Dentro do texto do System Message, encontre a seção chamada `## Mapeamento de Agendas (Fonte da Verdade)`.
 - Você verá uma lista de especialidades/profissionais com seus respectivos `id_agenda`. Exemplo:
 - ****Especialidade: Dermatologia (Dr. Ricardo Mendes)****
 - ``id_agenda`: 'bb4417...google.com'`
 - ****Especialidade: Odontologia (Dra. Ana Clara)****
 - ``id_agenda`: '0d475...google.com'...`
 - **Substitua** os valores de exemplo dentro das aspas simples ('...') pelos **IDs reais** das agendas correspondentes no seu Google Calendar.
 - **Como encontrar o ID da Agenda:** Vá ao Google Calendar, encontre a agenda desejada, clique nos três pontos > "Configurações e compart.", procure por "Integrar agenda" e copie o "ID da agenda".
 - Certifique-se de manter a estrutura e a formatação exatas do prompt.
4. **Configure o Manipulador de Transferência Humana (dentro do Console):**
- Nó `trataDados1`: **CRÍTICO!** Preencha os campos `numeroEquipe` (WhatsApp que receberá a notificação), `evoDomain`, `evoInstance` e `evoAPIKey`.
 - Nó `pausalaSupabase`: **CRÍTICO!** Selecione sua credencial do Supabase e verifique os nomes da tabela (`chats`) e colunas (`phone`, `ai_service`).
5. **Personalize as Personalidades (Opcional):**
- Você pode clicar em cada nó de agente (`agente_rotador`, `agente_atende_qualifica`, `agente_agendador`, `agente_faq`, `agente_especialista_procedimentos`) e editar o `systemMessage` para ajustar a personalidade, regras e fluxos de conversa, deixando o bot com a cara da sua clínica.
6. **Ative o Workflow:** Clique no botão "Active".
-

5. Passo 4: Configurando a Entrada e Saída (WhatsApp)

Este workflow conecta o WhatsApp à sua IA.

- **Workflow a ser utilizado:** **ENTRADA E SAIDA MODELO**

1. **Conecte sua Evolution API:**
 - Nó **webhookEvo**:
 - Ative o workflow.
 - Copie a URL de produção (**Production URL**).
 - Cole esta URL no campo de Webhook da sua instância na Evolution API.
 - Nó **trataDados**: **CRÍTICO!** Preencha os campos **evoDomain**, **evoInstance** e **evoAPIKey** com as informações da sua instância da Evolution API.
 2. **Conecte seus Bancos de Dados (Supabase e Redis):**
 - **Nós Supabase** (**procuraTelefoneSupaBase**, **criaUsuarioSupaBase**, **pausaAtendimentoIA**, **reativaAtendimentoIA**, **verificaStatusIA**, **buscaChatPorTelefone**, **criaNovoChatSupabase**, **atualizaChatExistente**, **gravaHistoricoMensagem**): Selecione sua credencial do Supabase em todos eles e verifique se os nomes das tabelas (**chats**, **chat_messages**) e colunas (**phone**, **conversation_id**, **ai_service**, etc.) correspondem à sua estrutura.
 - **Nós Redis** (Ex: **salvaTextoUsuario2**, **salvaMemoriaTranscricao1**, **recuperaMemoriaInicial**, **recuperaMemoriaAtual**, **limpaMemoriaChatRedis**): Selecione sua credencial do Redis em todos eles.
 3. **Conecte a Inteligência Artificial (OpenAI):**
 - Nos nós **transcreveAudioOpenAI**, **analisaImagemOpenAI**, **analisaFigurinhaOpenAI**, selecione sua credencial da OpenAI.
 4. **Conecte ao Console da IA:**
 - Nó **acionaMultiagentes**: No campo **Workflow ID**, certifique-se de que o workflow **CLINICA CONSOLE - TEMPLATE** está selecionado.
 5. **Configure o Filtro de Teste (Opcional, mas recomendado):**
 - Nó **filtroNumero**: Durante os testes, para que o bot responda apenas ao seu número, edite este nó. No campo **Right Value** de uma das condições, substitua o número existente pelo seu número de WhatsApp no formato **55DDDXXXXXXXX@s.whatsapp.net**. **LEMBRE-SE DE DESATIVAR OU REMOVER ESTE NÓ ANTES DE IR PARA PRODUÇÃO!**
 6. **Ative o Workflow**: Clique no botão "Active". O workflow está pronto para receber e enviar mensagens.
-

6. Passo 5: Configurando os Lembretes de Consultas

Este workflow enviará mensagens automáticas aos pacientes lembrando-os de suas consultas.

- **Workflow a ser utilizado:** LEMBRETES DE CONSULTAS MULTI AGENDAS.json
1. **Defina os Horários dos Gatilhos:**
 - Nó **disparaLembretes24hAntes**: Configura o horário para enviar lembretes com 1 dia de antecedência (ex: meio-dia). Clique no nó e ajuste o **Trigger At Hour**.
 - Nó **disparaLembretesMesmoDia**: Configura o horário para enviar lembretes no dia da consulta (ex: 7 da manhã). Clique no nó e ajuste o **Trigger At Hour**.
 2. **Configure os IDs das Agendas dos Profissionais:**

- **CRÍTICO!** Este workflow precisa saber quais agendas verificar para encontrar as consultas. Essa configuração centraliza a lista de agendas.
- Localize e clique nos nós `defineAgendas` (há um para cada gatilho, mas a configuração é a mesma).
- No painel de configurações à direita, localize a atribuição (assignment) chamada `agendas`.
- Clique no campo `Value` para editar a lista (Array) de agendas. Você verá uma estrutura similar a esta:

```
[
  {
    "medico": "Dr. Ricardo Mendes",
    "especialidade": "Dermatologia",
    "id_agenda":
    "bb4417b38dff3bb5b9d739abdf2faa2d4f9330379912b91b776c494164507cc9@group.calen
    dar.google.com"
  },
  {
    "medico": "Dra. Ana Clara",
    "especialidade": "Odontologia",
    "id_agenda":
    "0d475beb43a70cbe7603ef907e07db6fd0c3a9d428974d358a816adfe0c8e3a1@group.cale
    ndar.google.com"
  },
  // ... mais profissionais
]
```

- **Substitua** os valores de exemplo dentro das aspas ("`...`") no campo `id_agenda` pelos **IDs reais** das agendas correspondentes no seu Google Calendar para **cada profissional**.
 - **Como encontrar o ID da Agenda:** Vá ao Google Calendar, encontre a agenda desejada, clique nos três pontos > "Configurações e compart.", procure por "Integrar agenda" e copie o "ID da agenda".
- Adicione ou remova objetos `{...}` conforme necessário para listar todos os seus profissionais. Certifique-se de manter a estrutura JSON válida (vírgulas entre os objetos, chaves e valores entre aspas).

3. Conecte as Credenciais:

- **Google Calendar:** Nos nós `buscaConsultasAmanha` e `buscaConsultasDeHoje`, selecione sua credencial do Google Calendar. O campo `Calendar` já está configurado para usar o `id_agenda` do loop, então não precisa mexer nele.
 - **Supabase:** Nos nós `buscaDadosPaciente` e `buscaDadosPaciente1`, selecione sua credencial do Supabase. Verifique se os nomes da tabela (`chats`) e colunas (`email`, `phone`, `conversation_id`) estão corretos.
 - **OpenAI:** Nos nós `OpenAI Chat Model1` (lembrete 24h) e `OpenAI Chat Model` (lembrete hoje), selecione sua credencial da OpenAI.
 - **Postgres (Memória):** Nos nós `Postgres Chat Memory` e `Postgres Chat Memory1`, selecione sua credencial do Postgres onde a tabela `n8n_chat_histories` está.
 - **Evolution API:** Nos nós `trataDados` e `trataDados1`, **CRÍTICO!** Preencha os campos `evoDomain`, `evoInstance` e `evoAPIKey` com os dados **da sua** instância da Evolution API.
4. **Ative o Workflow:** Clique no botão "Active" no canto superior direito. Os lembretes agora serão disparados automaticamente nos horários definidos, consultando as agendas corretas.
-

7. Conclusão

Parabéns! Se você seguiu todos os passos, seu sistema de multiagentes para clínicas está configurado e pronto para operar.

*** Lembre-se de remover ou desativar o nó `filtroNumero` no workflow **ENTRADA E SAIDA MODELO** antes de divulgar o número para produção.