

Documentação — Aquecimento de Chips WhatsApp para SDR (n8n + Evolution API)

Público-alvo: time de vendas/SDR, operações e tecnologia. Escrito em linguagem simples para quem não é técnico.

1) O que é “aquecimento de chip” (no contexto de WhatsApp) e por que fazer

Aquecimento de chip é o processo de colocar novos números/contas de WhatsApp para **conversarem de forma legítima, humana e gradual** antes de usá-los em operações do time de SDR. Objetivo:

- Validar que os números funcionam bem (envio/recebimento, áudio, mídia, grupos).
- Reduzir bloqueios por comportamento anormal.
- Ter **redundância** quando a conta principal cair/ficar instável ("quando o WPP cai").
- Garantir **conformidade** com regras do WhatsApp e leis (LGPD): comunicação com **contatos que deram consentimento e sem spam**.

Importante: Não é um processo para “burlar” regras. É um **piloto operacional** com contatos de teste/ambiente controlado e conversas reais e consentidas. Nada de disparo em massa/sem opt-in.

2) O que vamos construir (visão geral)

Um ambiente onde ≥ 10 **chips (números)** de WhatsApp conversam em **privados e em grupos**, orquestrados pelo **n8n** (automação) e conectados via **Evolution API** (provedor de WhatsApp escolhido). O n8n:

- **Recebe mensagens** dos chips (webhooks) e registra tudo.
- **Responde/conduz conversas** com roteiros simples e/ou com apoio de IA, quando apropriado.
- **Monitora saúde** dos números e faz **rotação/falha** se um cair.
- **Gera relatórios** para o time de SDR/Operações.

Não haverá **front-end** no início; operaremos pelo painel do n8n e relatórios (planilha/DB/dashboard). Front pode vir depois.

3) Escopo e premissas

Escopo IN:

- Provisionar n8n e integrar com Evolution API.
- Cadastrar ≥ 10 chips e conectá-los.
- Configurar **webhooks**, filas, logs e relatórios básicos.
- Rodar **conversas reais** (internas/consentidas), em **privado** e **grupos** de teste.
- Mecanismos de **monitoramento** e **fallback** quando um chip cair.

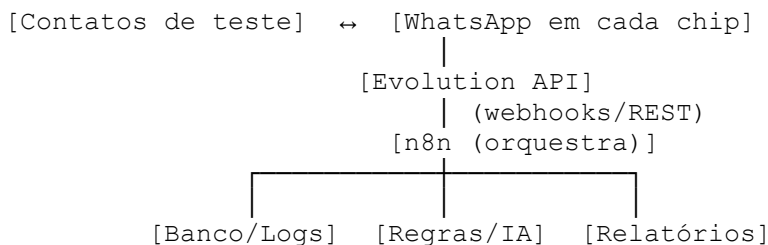
Escopo OUT (não entra agora):

- Disparo em massa/sem opt-in (proibido).
- Front-end próprio (opcional no futuro).
- Integrações com CRM (pode ser fase 2).

Premissas:

- Contatos/grupos usados no aquecimento têm **consentimento claro**.
 - Conteúdo é **útil, humano e não repetitivo**.
 - Seguiremos LGPD e políticas do WhatsApp.
-

4) Arquitetura (alto nível)



Componentes:

- **Chips (≥ 10)**: números/contas WhatsApp em aparelhos ou instâncias multi-device.
 - **Evolution API**: ponte de integração (envio/recebimento, QR code, webhooks).
 - **n8n**: fluxos de automação (rotas de mensagem, respostas, monitoramento, relatórios).
 - **Armazenamento**: Postgres/SQLite/Planilha (logs de mensagens, métricas).
-

5) Requisitos

5.1 Pessoas

- **Owner do projeto** (operações): aprova políticas, contatos de teste e grupos.
- **Técnico(a) n8n**: cria e mantém fluxos.
- **Responsável LGPD/compliance**: revisa termos e consentimentos.

5.2 Técnicos (mínimo viável)

- Servidor para **n8n** (Docker recomendado) com HTTPS.
- Acesso ao **Evolution API** (instância/tenant, token e webhooks).
- Banco de dados (Postgres recomendado) **ou** planilha compartilhada.
- Caixa de e-mail para alertas (opcional) e monitoramento.

5.3 Contas e números

- **≥ 10 chips** ativos, com **perfil completo** (nome/foto/bio/descrição).
- Grupos de teste (ex.: "Piloto-SDR-Grupo-01/02...") com **participantes que aceitaram**.

6) Padrões de conversa e limites operacionais

Para parecer humano e saudável:

- **Variação natural**: não repetir texto idêntico; alternar mídia (texto/áudio/imagem), tamanho das mensagens e tempos de resposta.
- **Horários**: janelas de 09:00–18:00 úteis, com pausas. Evitar madrugada.
- **Conteúdos úteis**: dúvidas reais, confirmações, follow-ups, validações. Nada de link frio publicitário.
- **Grupos**: respeitar tema, regras e etiqueta. Sem flood.
- **Crescimento gradual**: começar com **poucas conversas por dia**, aumentando somente quando métricas (respostas/denúncias/bloqueios) estiverem saudáveis. *Sem números exatos aqui de propósito: foco é qualidade, não volume.*

Se uma conta receber bloqueio/advertência: **pausar imediatamente**, investigar conteúdo/ritmo e só retomar após ajuste.

7) Fluxos no n8n (MVP)

Fluxo A — Onboarding de chip

1. Registrar chip no banco (número, apelido, status, tags).
2. Parelhar QR code via Evolution API.
3. Salvar **token/instanceId** e habilitar **webhook inbound**.
4. Rodar teste rápido (mensagem ping/pong para um contato interno).

Fluxo B — Recebimento de mensagens (webhook → roteador)

- Recebe webhook do Evolution.
- **Valida** assinatura/origem.
- Normaliza payload (contato, grupo, tipo de mídia).
- **Classifica** (privado × grupo) e envia para subfluxos.
- **Loga** no banco (mensagem, timestamp, chip, contexto).

Fluxo C — Resposta/Condução de conversa

- Regras simples (ex.: palavras-chave, intenção) e/ou IA com limites.
- Seleciona **chip ativo** (ou responde pelo mesmo chip que recebeu).
- Envia resposta via Evolution (`sendMessage`), com **backoff** se falhar.

Fluxo D — Agendador de conversas de teste

- Agenda conversas internas **com consentimento**: pares de contatos simulando cenários de SDR (ex.: confirmação de horário, envio de material, follow-up).
- Evita padrões repetitivos (usa variações de templates).

Fluxo E — Health-check e fallback

- A cada X minutos: checa status dos chips.
- Se um chip cair: marca como **indisponível**, **notifica** e **rota** mensagens por outro chip.

Fluxo F — Relatórios

- Gera CSV/planilha diária com: mensagens enviadas/recebidas, % respostas, grupos ativos, alertas.

8) Modelo de dados (sugestão simples)

chips

- id, numero, apelido, status(`ativo`|`indisponivel`|`pausado`), criadoEm, atualizadoEm

contatos

- id, nome, whatsapp, consentimento(`sim`|`nao`), origem, tags

grupos

- id, nome, waGroupId, participantes (lista), regras, tags

mensagens

- id, chipId, contatoId/grupoId, direcao(in|out), tipo(texto|audio|imagem|doc), conteudo, ts, statusEntrega

eventos

- id, tipo(erro|alerta|info), descricao, referenciaId, ts
-

9) Conteúdo de conversa (exemplos seguros)

Somente com pessoas do time/participantes que aceitaram participar dos testes.

- **Confirmações simples:** “Oi, tudo bem? Você consegue falar às 15h?”
- **Compartilhamento de agenda:** “Te enviei o convite do Google Agenda, recebeu?”
- **Follow-up curto:** “Boa tarde! Conseguiu ver o material?”
- **Grupos:** tópicos combinados (ex.: agenda da semana, estudo de caso), evitando mensagens repetitivas.

Evitar: correntes, promoções frias, links encurtados, anexos grandes, listas de transmissão não consentidas.

10) Operação e segurança

- **LGPD:** manter base de contatos de teste com origem/consentimento e opção de sair.
 - **Políticas WhatsApp:** sem automação abusiva, sem coleta indevida de dados, sem spam.
 - **Segredos (.env):** tokens/keys só no servidor seguro. Acesso com 2FA.
 - **Backups:** banco/planilhas diárias. Logs rotacionados.
 - **Observabilidade:** alertas por e-mail/Slack quando chip cair ou houver pico de erros.
-

11) Passo a passo de implantação (resumo)

1. **Infra:** subir servidor com Docker e HTTPS.
2. **n8n:** instalar/rodar; criar credenciais (HTTP Request, Webhook).
3. **Evolution API:** criar instância, configurar webhooks apontando para o n8n.
4. **Banco/Planilha:** criar tabelas/abas conforme modelo de dados.
5. **Onboarding dos 10 chips:** perfis completos, pareamento, teste ping/pong.
6. **Grupos de teste:** criar/descrever regras; adicionar participantes consentidos.
7. **Fluxos:** importar/ajustar Fluxos A–F e validar em sandbox.
8. **Rodada piloto (1 semana):** conversar pouco, conteúdo variado, medir saúde.

9. **Revisão:** métricas, feedback do time, ajustes de regras/temas.
-

12) Métricas/KPIs que importam

- **Taxa de resposta** (respostas ÷ mensagens saindo): qualidade da conversa.
 - **Alertas/bloqueios** por chip: saúde da conta.
 - **Mensagens por chip por dia:** manter dentro de zona segura e natural.
 - **Tempo médio de resposta:** parecer humano.
 - **Erros de entrega/API:** estabilidade técnica.
-

13) Plano de contingência (quando “o WPP cai”)

- **Deteção:** health-check identifica quedas.
 - **Ação:** pausar automações do chip afetado, registrar evento, notificar time.
 - **Reroteamento:** mensagens ativas passam para outro chip **apenas se for conversa em andamento e consentida**.
 - **Pós-incidente:** revisar conteúdo/ritmo; ajustar limites antes de retomar.
-

14) Roadmap (opcional)

- **Fase 2:** integração com CRM, dashboards (Grafana/Looker), front leve para o SDR.
 - **Fase 3:** modelos de IA mais contextuais (com red team/conformidade), avaliação semântica de qualidade.
-

15) FAQ rápido

Isso substitui o WhatsApp Business oficial? Não. É um piloto operacional. Siga sempre políticas e LGPD.

Podemos usar listas de transmissão? Somente com opt-in claro. Evitar em aquecimento.

Quantas mensagens por dia? Comece **pequeno**, só aumente quando saúde estiver boa. Foco em qualidade.

Posso copiar/colar a mesma mensagem para vários contatos? Não. Evite padrões; varie conteúdo e timing.

16) Anexos (referências técnicas úteis)

- **n8n**: nós usados (Webhook, HTTP Request, IF, Switch, Function, Wait, Schedule, Postgres/Sheets, Error Trigger).
- **Evolution API** (genérico):
 - **Webhook inbound**: recebe JSON de mensagens.
 - **Envio**: endpoint REST `sendMessage` com campos básicos (to, type, body/mediaUrl, messageId).
 - **Status**: endpoint para health-check da instância.

Os nomes de endpoints variam por provedor/versão. Validar na sua documentação do Evolution API.

17) Checklists imprimíveis

Antes de rodar o piloto:

- ☐ 10 chips com perfil completo
- ☐ Instância Evolution ativa e webhook → n8n
- ☐ Fluxos A–F publicados
- ☐ Tabelas/planilha criada
- ☐ Grupos de teste com regras e consentimento

Diário (operador):

- ☐ Health-check sem alertas
- ☐ Conversas variadas (sem padrões)
- ☐ Sem reclamações/bloqueios
- ☐ Relatório exportado

Incidente:

- ☐ Pausar chip afetado
 - ☐ Notificar time
 - ☐ Rerrotear apenas conversas consentidas
 - ☐ Post-mortem simples
-

Mensagem final

Construiremos um **ambiente robusto e responsável** de conversas reais para o time de SDR, com foco em qualidade, segurança e conformidade. Qualquer dúvida, ajuste ou necessidade de exemplos de fluxos no n8n, é só pedir que incluimos na próxima versão deste documento.