

MX DIGITAL

Nada sai sem você

O passo a passo de um SDR no WhatsApp com n8n, Claude e OpenWA — inclusive as armadilhas que só aparecem com ele rodando.

Escrito por Maxwell, da MX Digital. É a mesma máquina que eu mostrei no vídeo, contada de verdade: o que ela faz, como se monta, e o que quebra no caminho. Se você comentou "n8n", é isto aqui.

[n8n](#)

[Claude API](#)

[OpenWA](#)

[Postgres](#)

[Telegram](#)

- 00 O que isto é, e o que não é
- 01 As peças
- 02 O desenho em uma tela
- 03 Montando, na ordem
- 04 As armadilhas
- 05 O que a máquina não resolve

O que isto é, e o que não é

O que você vai montar é um SDR que **abre conversa no WhatsApp, faz uma pergunta por vez, sobe uma escada de qualificação e para** — porque nada sai sem alguém aprovar.

Não é disparador em massa. Se você quer isso, o caminho é outro e eu não vou te ajudar com ele.

A diferença não é moral, é econômica. Máquina que dispara sem ninguém olhar não economiza o seu tempo: ela **queima o seu número**, e leva o seu nome junto. Você tem um número só. Perde ele e perde a operação inteira, não uma campanha.

Por isso a arquitetura inteira gira em torno de um detalhe que parece burocracia e é o que segura tudo: **toda mensagem passa por um cartão de aprovação** no seu Telegram antes de sair. Você lê em oito segundos e aperta um botão. É lento de propósito, e é o que faz o resto funcionar.

O QUE VOCÊ PRECISA SABER ANTES

- Ler JSON sem se assustar. Não precisa programar.
- Saber o que é uma chamada HTTP. Vai fazer várias.
- Um pouco de SQL ajuda muito, mas dá pra começar sem.
- Paciência com erro que mente. O capítulo 04 é só sobre isso.

Aviso honesto sobre o OpenWA. Ele é WhatsApp Web por baixo, não a API oficial. Serve pro *seu* número, onde o risco é seu e você conserta quando cai. Não serve pra rodar em nome de vinte clientes: são vinte sessões não oficiais, vinte QR pra reescanear e vinte banimentos esperando. Quando virar produto pra terceiros, o caminho é a API oficial da Meta, com verificação de empresa e template aprovado. Começar pelo OpenWA é certo; ficar nele pra sempre, não.

As peças

n8n — o orquestrador

É onde tudo se encontra. Fluxos com gatilho de horário, chamadas HTTP, nós de código em JavaScript e integração com Telegram. Rode **auto-hospedado** (uma VPS de 10 dólares dá conta) — na nuvem deles você paga por execução, e esta máquina executa muito.

Claude — o que escreve e o que decide

Duas funções distintas, e é importante que sejam duas. Uma escreve a abordagem a partir dos dados reais do negócio. A outra lê a conversa inteira e decide o próximo passo. Chamadas diretas à API, sem framework: é um POST com um `system` e uma pergunta.

OpenWA — a mão que digita

Servidor local que expõe o seu WhatsApp como API HTTP. Roda em Docker, você lê um QR uma vez, e depois manda mensagem com `POST /sendText`. Ele também guarda o histórico, o que vai importar no capítulo 03.

Um banco de dados — a memória

Qualquer Postgres serve; eu uso Supabase pela API pronta. Aqui mora o que sobrevive ao fluxo: os leads, o estado de cada conversa, cada mensagem enviada e por qual versão do script. **Sem banco você não tem um SDR, tem um disparador com passos.**

Telegram — a sua mão no volante

Um bot simples, criado no @BotFather. É onde o cartão de aprovação chega e de onde a sua decisão volta. Telegram e não e-mail porque você precisa decidir em oito segundos, na fila do mercado.

O desenho em uma tela

Três fluxos no n8n, e cada um tem um trabalho só.

1 O relógio

De tempos em tempos, pergunta ao banco se pode abordar (horário, dia, teto do dia), pega o próximo lead da fila, pede pro Claude escrever a abordagem, grava e manda o cartão pro Telegram.

Não envia nada.

2 O ouvido

A cada poucos minutos, procura conversa onde a última mensagem é da pessoa e ainda não foi respondida. Manda a conversa inteira pro Claude, recebe a decisão, grava, e manda outro cartão.

3 A mão

Recebe o clique do botão do Telegram. Só ele fala com o WhatsApp. Divide a resposta em partes, manda uma de cada vez com alguns segundos entre elas, e marca no banco o que saiu.

Repare na separação: **quem pensa não envia, e quem envia não pensa**. Foi a decisão mais barata e mais valiosa do projeto inteiro. Quando o Claude escreve besteira, a besteira fica parada num cartão. Quando o envio falha, é um fluxo só pra investigar. E o botão do Telegram é a única porta entre os dois.

As três tabelas que importam

leads	quem é, telefone, nicho, cidade, avaliações, de onde veio, e uma coluna dizendo se ainda está na fila
estado	uma linha por conversa: em que degrau está, com quem está falando (robô, porteiro, decisor), quando ela falou pela última vez
envios	uma linha por mensagem que a máquina escreveu: as partes, quando foi decidida, qual decisão, quando saiu, e qual versão do script escreveu

A coluna de versão do script na tabela de envios parece detalhe e é o que permite responder, no fim da semana, **qual abordagem funcionou melhor**. Sem ela você troca o texto o tempo todo e nunca sabe o que aconteceu.

Montando, na ordem

Passo 1 · Suba o OpenWA e mande uma mensagem pra você mesmo

Antes de qualquer fluxo. Docker, um contêiner, ler o QR com o celular, e um `curl` mandando "oi" pro seu próprio número. Se isso não funciona, nada depois vai funcionar, e você vai passar horas achando que o problema é no n8n.

Ponha o OpenWA atrás de um túnel (Cloudflare Tunnel resolve de graça) em vez de abrir porta. O n8n precisa alcançá-lo, e você não quer esse endereço exposto.

Passo 2 · O banco, e a fila

Crie as três tabelas. Depois crie uma função no banco que responda **uma** pergunta: *posso abordar agora?* Ela olha horário, dia da semana, quantas conversas já foram abertas hoje e a taxa de quem pediu pra parar. Devolve sim ou não, com o motivo escrito.

Escreva essa regra **no banco, não no fluxo**. Ela decide dinheiro, e regra que decide dinheiro não mora em lugar que se edita arrastando caixinha.

Passo 3 · O relógio

Gatilho de horário → pergunta "posso abordar?" → se sim, pega o próximo da fila → monta o pedido pro Claude com os dados reais daquele negócio → recebe as mensagens prontas → grava → manda o cartão.

O cartão traz quem é, por que entrou na fila, e o **texto exato** que vai sair. Dois botões: aprovar e pular.

Passo 4 · O contrato com o Claude

Este é o passo que separa quem monta algo estável de quem passa o mês consertando. Peça JSON, sempre, com campos fixos:

```
{
  "com_quem_falo": "robo | porteiro | decisor | ninguem",
  "estado": "<o degrau em que a conversa ficou>",
  "categoria": "andando | interessado | pergunta | preco |
               sem_interesse | opt_out | numero_errado",
  "deve_responder": true,
  "resposta": "o texto, OU uma lista de textos",
  "notas": { },
  "notificar_humano": false,
  "motivo": "no máximo 15 palavras"
}
```

`resposta` aceitar lista é o que permite mandar três balões curtos em vez de um texto longo — que é como gente escreve no WhatsApp. `notas` é onde ele devolve só o que a pessoa realmente disse, pra virar coluna no banco.

Passo 5 · O ouvido

Uma consulta no banco resolve: conversas não encerradas em que a última mensagem é *de entrada* e é mais nova que a última resposta sua. Devolva junto o **histórico inteiro** da conversa, não só a última fala.

Ver a conversa inteira é a diferença entre responder à mensagem e responder à pessoa. Custa mais token e paga.

Passo 6 · A mão

Webhook recebe o clique → confirma no banco que o cartão ainda vale → divide a resposta em partes → **manda uma, espera uns segundos, manda a próxima** → marca o que saiu.

A pausa entre as partes não é enfeite. Três mensagens no mesmo segundo é assinatura de robô, e mais importante: **o cartão precisa conferir, antes de enviar, se a pessoa escreveu de novo desde que ele foi escrito**. Se escreveu, o cartão está velho — feche como pulado e deixe o próximo ciclo escrever outro, com a fala nova na frente.

As armadilhas

Nenhuma destas está em tutorial. Todas custaram horas aqui.

O n8n mente sobre o cron

Ao ativar um fluxo com gatilho de horário, ele pode recusar com `Invalid cron expression`. Você vai reescrever a expressão seis vezes, e ela está certa.

O erro real é o fuso. Falta `settings.timezone` no fluxo. Defina o fuso e a mesma expressão passa.

Renomear um nó desliga o fluxo em silêncio

As conexões entre nós são indexadas **pelo nome**. Renomeie um nó pela API e as conexões apontam pra um nome que não existe mais. O fluxo continua ativo, e não faz nada.

Renomeou, mova a chave. Troque o nome no objeto de conexões também, e nas referências que outros nós fazem a ele.

O Telegram quebra com underline

No modo padrão, `_` é itálico. Um `callback_data` com underscore no texto derruba a execução com `can't parse entities` — e derruba **depois** que as mensagens já saíram, então você acha que falhou o envio.

Force HTML e escape. `parse_mode: HTML`, e troque `&`, `<` e `>` antes de mandar.

A IA escreve conversa antes do JSON

De vez em quando vem "Aqui estão as mensagens:" antes do objeto. O `JSON.parse` estoura, a execução morre, e num gatilho de horário isso queima o horário inteiro.

Leia tolerante. Tente o texto inteiro; se falhar, recorte do primeiro `{` ao último `}` e tente de novo.

Português sem acento entrega a máquina

Junto com o travessão e o emoji, é a marca que mais rápido revela que do outro lado tem um robô. E pedir acento no prompt **não basta**: dois cartões gerados no mesmo minuto, mesmo prompt, um sai acentuado e o outro inteiro sem.

Vire conserto, não pedido. Uma lista de troca no código, aplicada antes de gravar, com as palavras que não têm outra leitura sem acento. E deixe de fora as ambíguas — “*sao*” pode ser São Paulo, “*esta*” pode ser esta casa.

Nunca peça ao modelo pra copiar um link

Pedi, e um dia ele escreveu `[link do portfólio]` — o rótulo, em vez do endereço. A mensagem saiu pro cliente sem link nenhum.

O que é fixo, o código põe. Endereço, telefone, nome de arquivo: nada disso se pede a um modelo. Ele escreve o texto; você insere o dado.

O mesmo contato tem dois endereços

No WhatsApp moderno, uma conversa pode existir como `@lid` ou como `@c.us`. Consultar a forma errada devolve “conversa não existe” — e você conclui que nada foi enviado quando tudo foi.

Guarde o identificador que o OpenWA devolveu no envio, e consulte por ele. Não remonte o endereço a partir do telefone.

A coluna com valor padrão mente calada

Uma coluna com *default* não reclama quando você esquece de preenchê-la: ela inventa o valor. Aqui, toda resposta de uma frente ficou arquivada como sendo da outra, por dias, sem erro nenhum. As mensagens estavam certas; só a medição estava errada.

Confira o que você mede, não só o que você envia. Uma consulta simples agrupando por frente teria mostrado no primeiro dia.

O que a máquina não resolve

Se você montou tudo até aqui, você tem um motor funcionando. E vai descobrir uma coisa incômoda: **a parte difícil não era essa.**

O motor é genérico. As mesmas peças servem pra clínica, pra corretor de imóveis, pra escritório de contabilidade. O que muda de um pro outro — e o que faz a diferença entre conversa que anda e conversa que morre — é o que não tem no n8n:

- **A escada.** Quais perguntas, em que ordem, e qual delas decide. Clínica pergunta volume e se dá conta. Corretor pergunta faixa de preço, região e financiamento aprovado. Não existe "os mesmos parâmetros".
- **A régua.** Quem pode comprar, antes de qualquer pergunta sobre dor. Uma conta simples que elimina metade da lista e economiza reunião.
- **O veredito.** A regra que decide se vale propor reunião — e ela mora em código, nunca no prompt. Um modelo que conhece a regra ainda escolhe ignorá-la quando acha que o lead "demonstrou interesse real". O meu fez isso, com a conversa na frente.
- **Saber o que fazer com a resposta.** Que é onde a maioria trava, e nenhuma automação cobre.

A máquina é vinte por cento. A escada é o resto.

Escrevo isso porque é verdade e porque é o que eu mais gostaria de ter lido antes de começar. Tem gente montando automação há meses achando que falta uma integração, quando o que falta é decidir qual é a segunda pergunta.

Se quiser continuar

Eu estou montando isso em público, inclusive as partes que quebram. Se você for montar o seu e travar em alguma dessas armadilhas, me chama e eu te falo como saí — não cobro por pergunta.

E se em algum momento você preferir que alguém escreva a escada do seu nicho e implante junto, é exatamente isso que eu faço na MX. Mas monta o seu primeiro. Você vai entender muito mais do próprio negócio no caminho.

