VMind

Passo a passo completo

O ciclo que *fecha sozinho*

Do pedido no WhatsApp até o post no ar

O que você vai montar

Um ciclo de conteúdo onde você só manda um texto dizendo o que quer (criar ou ajustar um post), uma IA gera a peça sozinha, te manda a prévia de volta no WhatsApp, e só publica depois que você responder um comando de aprovação. Você nunca abre editor de imagem nem o Instagram, só decide com um "sim ou não" no fim.

Antes de começar

Você precisa de:

1. **Uma instância de n8n** (pode ser na nuvem paga, ou auto-hospedada num servidor pequeno), é a ferramenta que vai orquestrar os passos e ligar tudo.
2. **Acesso à API do WhatsApp Business** (via Meta, ou um provedor intermediário tipo Twilio/Z-API/Evolution API), pra receber e mandar mensagem no grupo.
3. **Um agente de IA com acesso a arquivos e capacidade de gerar imagem/HTML** (no seu caso, o Claude rodando com uma automação que lê o pedido e gera o carrossel).
4. **Um lugar pra publicar as imagens geradas** (um servidor com pasta pública, ou serviço de storage com URL, tipo um bucket S3 com acesso público de leitura).
5. **Acesso à API do Instagram** (Graph API, conta business) pra publicar de fato quando aprovado.

Não precisa saber programar em nível avançado, o n8n é visual (conecta caixinhas), mas você vai mexer em webhooks e variáveis, então já ajuda ter mexido com automação antes.

1. Montando o webhook de entrada (onde a mensagem chega)

1.1 Criando o fluxo no n8n

Abra o n8n e crie um novo Workflow. Adicione um nó "Webhook" como gatilho. Esse webhook vai ser o endereço que a API do WhatsApp chama toda vez que uma mensagem nova chegar no seu número.

1.2 Conectando o WhatsApp ao webhook

No painel do seu provedor de WhatsApp Business, configure o "Webhook URL" apontando pra URL gerada pelo nó do passo 1.1. Isso normalmente fica em Configurações > Webhooks, no painel do provedor escolhido.

1.3 Testando o recebimento

Mande uma mensagem de teste no grupo (ex.: "teste 123") e veja se o n8n mostra a execução na aba de execuções, com o texto da mensagem dentro do JSON recebido. Se não aparecer nada, confira se a URL do webhook foi salva certa no provedor.

2. Filtrando o gatilho certo

2.1 Adicionando o nó de condição

Depois do webhook, adicione um nó "IF" (condição). Configure pra checar se o texto da mensagem começa com a palavra que você escolheu como gatilho (no caso do Marabo, é "marabo").

2.2 Por que filtrar

Sem esse filtro, qualquer mensagem do grupo (até conversa que não é pra IA) dispararia o fluxo. O filtro garante que só o que começa com a palavra-gatilho vira uma tarefa de verdade.

3. Mandando o pedido pra IA gerar o conteúdo

3.1 Chamando o agente

Depois do filtro, adicione um nó de HTTP Request (ou o nó nativo, se seu provedor de IA tiver um) que manda o texto da mensagem pra sessão do agente de IA responsável por gerar o carrossel.

3.2 O que o agente faz do lado dele

O agente recebe o pedido, decide se é criação ou ajuste, monta ou edita o arquivo de conteúdo (JSON, no caso do Marabo) e roda o script que renderiza as imagens e publica no storage público.

3.3 Conferindo a geração

No fim dessa etapa, você deve conseguir abrir a URL pública das imagens geradas num navegador e ver o carrossel pronto, com o texto e o visual da marca.

4. Mandando a prévia de volta pro grupo

4.1 Nó de resposta

Adicione outro nó de HTTP Request (ou o nó nativo do provedor de WhatsApp) que manda as imagens geradas de volta pro grupo, junto com uma mensagem tipo "Prévia pronta. Responda POSTAR pra publicar."

4.2 Conferindo o recebimento

Confirme que as imagens chegam no grupo do WhatsApp com boa qualidade e que a legenda de instrução ("responda POSTAR") está clara.

5. Montando o gatilho de aprovação

5.1 Segundo filtro no mesmo webhook

Volte no fluxo do passo 1 e adicione um segundo caminho a partir do nó "IF": se o texto da mensagem for exatamente "POSTAR" (ou variações, tipo "postar", em caixa baixa), segue pro fluxo de publicação.

5.2 Guardando qual conteúdo está pendente

Antes de aprovar, o sistema precisa saber qual é o conteúdo "atual" aguardando aprovação. A forma mais simples é sempre ter um único arquivo de conteúdo ativo (uma espécie de "rascunho atual"), e o POSTAR sempre publica esse rascunho.

6. Publicando no Instagram

6.1 Chamando a API do Instagram

No caminho de aprovação (passo 5.1), adicione um nó de HTTP Request que chama o endpoint de publicação da Graph API do Instagram (`media` e depois `media_publish`), usando as imagens já publicadas no storage do passo 3.3.

6.2 Confirmando no grupo

Depois da publicação, adicione um último nó que manda uma mensagem de confirmação no grupo com o link do post publicado.

6.3 Conferindo o resultado

Abra o Instagram e confira se o post saiu com a legenda certa e as imagens na ordem certa. Compare com a mensagem de confirmação que chegou no grupo, os dois devem bater.

Erros comuns

- **O webhook não recebe nada quando mando mensagem.** Causa: a URL configurada no provedor de WhatsApp está desatualizada (o n8n gera uma URL nova se você recriar o nó). Correção: sempre confira a URL atual do nó Webhook antes de configurar no provedor.
- **A IA responde no grupo errado, ou não responde.** Causa: o ID do grupo/número usado na etapa de resposta (passo 4.1) está trocado. Correção: confira o ID do grupo no payload recebido no passo 1.3 e use o mesmo ID pra responder.
- **O POSTAR publica o conteúdo errado.** Causa: existe mais de um "rascunho atual" ao mesmo tempo (por exemplo, dois pedidos seguidos sem esperar a prévia). Correção: trate cada pedido novo como substituição do rascunho anterior, nunca acumule mais de um pendente.
- **A publicação no Instagram falha com erro de token.** Causa: token de acesso da Graph API expirou (tokens de curta duração expiram em horas, os de longa duração em semanas). Correção: configure um token de longa duração e um lembrete pra renovar antes de expirar.

Próximo nível

Depois que o ciclo pedido → geração → aprovação → publicação estiver estável, o próximo passo é adicionar um comando de agendamento (tipo "agenda pra quinta 9h"), guardando a aprovação junto com uma data/hora futura em vez de publicar na hora, e uma rotina separada que roda de tempos em tempos checando o que já pode ser publicado.