## **Agente Nutricionista com n8n**

### **Qual é o objetivo desse projeto?**

Hoje a gente vai criar **um agente de inteligência artificial que atua como nutricionista**.

O fluxo começa com uma simples mensagem via **WhatsApp**. Pode ser um **texto**, pode ser uma **foto de um prato de comida**. E o que esse agente vai fazer?

* Se for **texto**, o agente interpreta e responde normalmente.
* Mas se for uma **imagem de comida**, o agente analisa visualmente o prato, identifica alimentos como **arroz, feijão, carne, salada** e estima quantas **calorias** esse prato tem.

### **O que vamos construir?**

Você vai aprender a montar um fluxo completo no **n8n**, que envolve:

* Receber mensagens via **webhook do WhatsApp**
* Identificar o tipo da mensagem (texto ou imagem)
* Se for imagem: fazer **download**, converter para **Base64**, passar por uma **Vision API**
* Usar um **IA Agent** para interpretar a imagem e retornar a quantidade aproximada de calorias do prato
* E por fim, **responder automaticamente no WhatsApp**

### **O que você vai aprender na prática?**

Essa aula é uma aula prática e avançada de automação com IA. Você vai entender como:

1. **Integrar o WhatsApp com o n8n**
2. Fazer controle condicional com If e dividir fluxos por tipo de mídia
3. Usar **Visão Computacional (Vision API)** para interpretar imagens
4. Usar um **Agente de IA (com GPT)** para gerar uma resposta contextual com base na imagem
5. Retornar essa resposta automaticamente pelo WhatsApp

### **Por que esse projeto é importante?**

Esse é um exemplo poderoso de como **agentes de IA** e automações podem ser aplicados em **casos do mundo real**: nutrição, saúde, atendimento automatizado, fitness e muito mais.

E mais: você estará aprendendo a **combinar visão computacional com linguagem natural**, que é uma das **fronteiras mais quentes da IA hoje**.

## **Atualizando a versão da Evolution API**

Antes de começarmos a montar o fluxo do agente nutricionista, precisamos garantir que a **versão da API do WhatsApp** está atualizada para funcionar corretamente com a Evolution API no n8n.

### **Passo 1: Acesse o site**

Abra o navegador e vá para:

https://web.whatsapp.com

### **Passo 2: Abra o Console do Navegador**

1. Clique com o botão direito na página e selecione **"Inspecionar"**.
2. Vá até a aba **"Console"**.

### **Passo 3: Pegue a versão do WhatsApp Web**

Digite o seguinte comando no console:

window.Debug.VERSION

Ele vai retornar algo como:

"2.3000.1025110696"

Essa é a versão atual do WhatsApp Web que está em execução no seu navegador.

**Passo 4: Atualize a variável no servidor (EasyPanel)**

No seu painel de gerenciamento do n8n (como o EasyPanel), edite as **variáveis de ambiente** da sua aplicação e atualize a linha:

CONFIG\_SESSION\_PHONE\_VERSION=2.3000.1025110696

**Dica**: Essa variável garante que a sessão WhatsApp seja compatível com a versão usada no navegador. Se não estiver atualizada, pode causar falhas na autenticação ou na leitura de mensagens.

### **Finalize salvando e reiniciando o container.**

## **Instalar o node Evolution API**

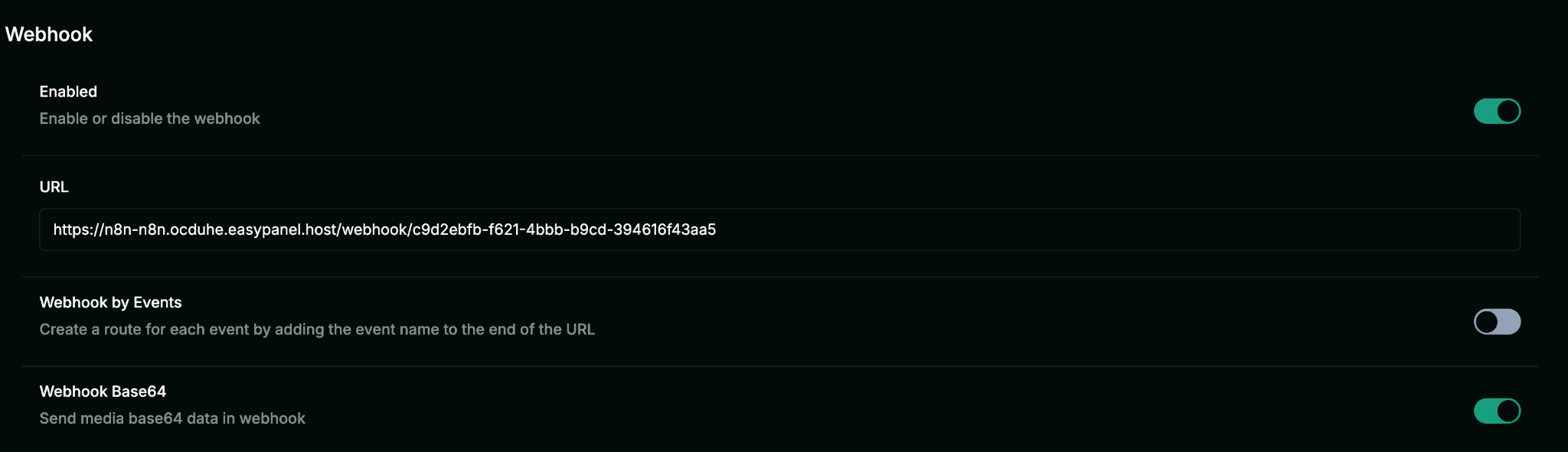
* Instalar Node e Pegar as variáveis

## **1. GERENCIADOR DE MENSAGEM**

### **📍 Webhook1**

* **Entrada de dados do WhatsApp (via Evolution API)**.

Importante lembrar de atualizar na Evolution API



* Gatilho para todas as mensagens que chegam, sejam **texto ou imagem**.

### **TypeMessage (Switch)**

* **Identifica o tipo de mensagem**:  
  + Se existe campo message.conversation, é **texto** → ativa o fluxo de consulta.
  + Se existe campo imageMessage.mimetype, é **imagem** → ativa o fluxo de processamento de imagem.

### **WelcomeMessage**

* Envia uma **mensagem padrão inicial** se for a primeira interação.
* Exemplo:  
    
   "Seja bem-vindo! 😄 O que você comeu hoje? Me manda uma foto do seu prato e eu te conto quantas calorias tem aí!"

### **formatToLLM**

* **Prepara o texto da mensagem recebida** (texto puro do WhatsApp) para ser enviado ao agente consultaNutricional.

### **📍 consultaNutricional**

* Agente de IA responsável por **consultar registros anteriores**.
* Usa um system message com uma **tool conectada ao Google Sheets** (consultaNutricional1) que busca os dados da data atual.

System message

Você é um agente nutricionista que recebe perguntas em texto pelo WhatsApp. Sua função é interpretar essas mensagens e consultar os dados registrados anteriormente em uma planilha de alimentação.

Você conta com uma ferramenta chamada `ConsultaNutricional`, que retorna os seguintes campos por data:

- data (formato YYYY-MM-DD)

- refeicao (ex: "almoço", "jantar", etc.)

- calorias (valor numérico)

- proteina (valor numérico)

- hora (ex: "12:45")

### Comportamento esperado:

1. Quando o usuário enviar perguntas como:

- "Quantas calorias eu comi hoje?"

- "Quanto de proteína já consumi?"

- "Qual foi minha última refeição?"

2. Use a ferramenta `ConsultaNutricional` para buscar os registros da data de hoje (`{{ $now }}`).

3. Some todas as calorias e proteínas do dia e responda com algo como:

👉 "Consultei aqui pra você! Hoje você já consumiu \*\*[X] kcal\*\* e \*\*[Y]g de proteína\*\*."

4. Se houver múltiplas refeições, você pode mencionar a última refeição registrada também:

👉 "Sua última refeição foi às 13:40, com 520 kcal e 38g de proteína."

5. Se \*\*não houver nenhum dado hoje\*\*, diga:

👉 "Ainda não encontrei nenhuma refeição registrada hoje. Que tal me mandar uma foto do seu prato agora?"

⚠️ Importante:

- \*\*Não use nenhuma ferramenta de gravação.\*\*

- Apenas use a `ConsultaNutricional` para buscar dados.

Todas as respostas devem ser em \*\*português\*\*, com linguagem amigável, simples e motivadora.

Datra de hoje: {{ $now }}  
**RespondToUser1**

* Envia a resposta gerada pelo agente consultaNutricional **de volta ao WhatsApp**.

## **2. PROCESSAMENTO DE IMAGEM**

### **📍 Wait5s1**

* Pausa para garantir que a imagem tenha sido processada pelo WhatsApp antes do download.

### **📍 DownloadImage**

* Usa a **Evolution API** para fazer o **download da imagem em Base64** usando o ID da mídia recebida.

### **Convert to File**

Converte o conteúdo Base64 em um **arquivo JPEG** com nome personalizado (usuário\_dataHora.jpeg), usando:

### **📍 Vision**

* Envia a imagem para o **modelo de visão do OpenAI (gpt-4o-mini)**.  
  System prompt

# Instruções para Análise Nutricional de Imagens

Sua tarefa é analisar a imagem enviada e retornar uma resposta estruturada com base no conteúdo visual. Siga \*\*rigorosamente as regras abaixo, na ordem de prioridade\*\*:

---

### 1. Se a imagem contiver \*\*comida\*\*, identifique:

\* Os \*\*alimentos visíveis\*\* (nomes comuns como arroz, feijão, chocolate, hambúrguer etc.).

\* Uma \*\*estimativa da porção\*\* de cada item (pequena, média ou grande).

\* A \*\*estimativa de calorias totais\*\* da refeição ou item.

\* A \*\*estimativa de proteína total\*\* (em gramas).

⚠️ A análise \*\*deve ser feita mesmo que a comida não esteja em um prato\*\*, como:

\* Lanches industrializados (ex: barra de cereal, chocolate).

\* Fast food (ex: hambúrguer, pizza).

\* Bebidas alimentares (ex: milkshake, vitamina).

---

### 2. Se a imagem \*\*não mostrar comida ou for inconclusiva\*\*, retorne:

📍 \*\*"Imagem fora do contexto nutricional"\*\*

📍 Inclua uma breve explicação visual, por exemplo:

\*"A imagem mostra um animal de estimação com um objeto ao fundo, sem presença clara de alimentos."\*

---

### 📌 Formato obrigatório da resposta:

```

Prato identificado

Alimentos: [lista com nome e porção de cada item]

Calorias estimadas: [valor aproximado em kcal]

Proteína estimada: [valor aproximado em gramas]

```

#### \*\*Exemplo 1\*\* (prato tradicional):

Prato identificado

Alimentos: arroz (médio), feijão (pequeno), frango grelhado (médio), salada (grande)

Calorias estimadas: 620 kcal

Proteína estimada: 40 gramas

#### \*\*Exemplo 2\*\* (snack):

Prato identificado

Alimentos: barra de chocolate ao leite (média)

Calorias estimadas: 210 kcal

Proteína estimada: 2 gramas

---

Todas as respostas devem ser escritas em \*\*português\*\* e seguir \*\*exatamente o formato acima\*\*.

## **3. SUPER AGENT – registrarRefeicao1**

### **📍 registrarRefeicao1**

* Agente de IA que **interpreta o texto da imagem** e gera um **output JSON estruturado** com:  
  + data
  + hora
  + refeicao
  + calorias
  + proteina
  + resumo (mensagem final para o usuário)

### **⚠️ Importante:**

Este agente **não usa nenhuma tool**, apenas o **Structured Output Parser** para garantir a saída correta.

System Message

Você é um agente nutricionista que recebe uma mensagem de texto contendo o resultado da análise de uma imagem de um prato de comida. Sua tarefa é interpretar esse conteúdo e gerar uma resposta no formato JSON estruturado.

A resposta \*\*deve seguir exatamente\*\* o seguinte formato:

{

"data": "YYYY-MM-DD",

"hora": "HH:mm",

"refeicao": "[descrição completa da refeição com alimentos e porções]",

"calorias": [número inteiro],

"proteina": [número inteiro],

"source": "whatsapp",

"resumo": "✅ Refeição registrada com sucesso! Foram aproximadamente [calorias] kcal e [proteina]g de proteína. Continue assim!"

}

### Regras:

\* Use a \*\*data e hora atual\*\* no formato correto (ex: `"2025-07-24"` e `"12:45"`).

\* O campo `refeicao` deve conter a descrição dos alimentos com porções, exatamente como foi extraído da análise (ex: "arroz (médio), feijão (médio), carne moída (média), salada (pequena)").

\* O campo `source` \*\*deve sempre ser\*\* `"whatsapp"`.

\* O campo `resumo` deve ser uma mensagem amigável que será enviada ao usuário confirmando o registro da refeição, com os valores de calorias e proteínas inseridos dinamicamente.

### Se não for possível identificar calorias, proteínas ou alimentos com clareza:

\* Retorne \*\*apenas\*\*:

{

"resumo": "Não consegui interpretar corretamente essa imagem. Pode enviar uma foto mais nítida, por favor?"

}

```

Todas as respostas devem estar em \*\*português\*\*. Seja objetivo, acolhedor e motivador.

Data de hoje: {{ $now }}

### **📍 Structured Output Parser**

* Valida e garante que o JSON gerado pelo agente esteja no formato correto.

{

"data": "2025-07-24",

"hora": "12:45",

"refeicao": "Almoço",

"calorias": 680,

"proteina": 45,

"resumo": "✅ Refeição registrada com sucesso! Foram aproximadamente 680 kcal e 45g de proteína no almoço de hoje, às 12:45. Continue firme!"

}

### **📍 Append row in sheet**

* Usa os dados do output para **registrar a refeição no Google Sheets** com as colunas:  
  + data, hora, refeicao, calorias, proteina

### **📍 RespondToUser**

* Envia a **mensagem final de confirmação da refeição registrada** com base no campo resumo do JSON estruturado.