

# Relatório Técnico - Desafio de Orquestração de IA com n8n para Análise de Dados

---

**Autor:** Joel Medeiros Neto **Data:** 22 de Agosto de 2025

---

## 1. Arquitetura da Solução

---

Este projeto implementa um sistema de Business Intelligence conversacional utilizando o n8n como orquestrador central. A arquitetura combina serviços na nuvem com serviços locais containerizados para garantir portabilidade e facilidade de desenvolvimento.

A interação do usuário ocorre da seguinte forma:

1. Uma pergunta em linguagem natural é enviada para um endpoint de Webhook no **n8n Cloud**.
  2. Para permitir que o n8n na nuvem se comunique com a API rodando localmente, a ferramenta **ngrok** é utilizada para criar um túnel seguro.
  3. O n8n orquestra um fluxo de IA que consulta a API em Python (FastAPI) através da URL pública do ngrok.
  4. A API, rodando em um contêiner Docker, busca os dados solicitados em um banco de dados PostgreSQL, que também roda em seu próprio contêiner.
  5. Os dados retornam pelo mesmo caminho (API -> ngrok -> n8n), onde um segundo processo de IA os utiliza para formular uma resposta em linguagem natural.
  6. O n8n Cloud envia a resposta final de volta ao usuário através do Webhook.
- 

## 2. Como Configurar e Rodar o Projeto

---

O ambiente local foi containerizado com Docker Compose para a API e o banco de dados, garantindo uma experiência de desenvolvimento (DX) simples.

### Pré-requisitos:

- Docker e Docker Compose instalados.
- Uma conta no n8n Cloud.
- A ferramenta ngrok instalada e autenticada.
- Um token de API da OpenAI.
- Git para clonar o repositório.

### Passos para Execução:

1. Clone este repositório: `git clone https://github.com/joelmedeirosn/desafio-ia.git`
2. Navegue até a pasta do projeto: `cd desafio-ia`
3. Execute o Docker Compose para subir a API e o Banco de Dados:

```
docker-compose up --build
```

4. Em um segundo terminal, inicie o ngrok para expor a porta 8000 da API:

```
ngrok http 8000
```

5. Copie a URL pública gerada pelo ngrok.

6. Acesse sua conta no n8n Cloud e importe os dois arquivos de workflow (workflow-etl.json e workflow-ia.json).

7. Nos workflows importados, atualize todos os nós HTTP Request para que apontem para a URL pública do ngrok.

8. No workflow de IA, configure a credencial da OpenAI e use o curl para enviar perguntas à URL de teste do Webhook do n8n.

## Testando o Agente de IA

Para interagir com o agente, envie uma requisição POST para a URL de teste do Webhook do workflow-ia.

### Exemplo 1: Pergunta Válida (Evento Encontrado)

**Comando:** ``bash curl.exe -X POST -H "Content-Type: application/json" -d '{"pergunta": "Qual a descrição do evento Wiki Facilit?"}' [SUA_URL_DE_TESTE_DO_WEBHOOK]`

### Exemplo 2: Pergunta Fora de Escopo (Guardrail)

**Comando:** ``bash curl.exe -X POST -H "Content-Type: application/json" -d '{"pergunta": "Quanto é 2+2?"}' [SUA_URL_DE_TESTE_DO_WEBHOOK]`

---

## 3. Detalhes da Implementação

---

### API (Etapa 1)

- **Framework:** FastAPI foi utilizado para a construção da API RESTful, garantindo alta performance e documentação automática (/docs).
- **Banco de Dados:** PostgreSQL, rodando em um contêiner Docker separado.
- **Funcionalidades:** A API expõe endpoints CRUD (GET, POST, PUT, DELETE) para a entidade eventos, servindo como a única camada de acesso aos dados.

### Workflow de ETL (Etapa 2)

- O primeiro workflow (workflow-etl.json) é responsável por ler os dados das três planilhas do Google Sheets.
- Foi utilizado um nó Code com JavaScript para a transformação e limpeza dos dados, especialmente para tratar os múltiplos formatos de data e remover textos indesejados (ex: (Trimestral), no campo que remete à data).

- Os dados transformados são enviados em lote para a API via HTTP Request, populando o banco de dados.

## Workflow de IA (Etapa 3)

- O segundo workflow (workflow-ia.json) funciona como o agente conversacional.
- **Gatilho:** Um nó Webhook aguarda por requisições POST contendo a pergunta do usuário.
- **Lógica da IA:** A estratégia utilizada foi um pipeline de duas etapas:
  1. **IA Extratora:** O primeiro Basic LLM Chain recebe a pergunta e tem a única tarefa de extrair entidades, como o nome do evento, ou classificar a pergunta como "fora de escopo". O prompt utilizado foi:

Sua tarefa é analisar o texto do usuário e extrair o nome de um evento.

- Se a pergunta for sobre eventos, agendas ou cronogramas, responda com o nome do evento.
- Se a pergunta for sobre qualquer outro assunto (exemplos: "qual o tempo para amanhã?"), responda com "fora de escopo".
- Se a pergunta for sobre um evento mas não for claro qual, responda com "fora de escopo".

Exemplo 1:

Texto: "Qual o período do evento Wiki Facilit?"

Resposta: {"nome\_do\_evento": "Wiki Facilit"}

Exemplo 2:

Texto: "Qual a previsão do tempo para amanhã?"

Resposta: {"nome\_do\_evento": "fora\_de\_escopo"}

Texto do usuário: {{ \$json.body.pergunta }}



2. **IA Respondedora:** Após a busca dos dados na API, o segundo Basic LLM Chain recebe a pergunta original e os dados encontrados, com a tarefa de sintetizar uma resposta final em linguagem natural. O prompt utilizado foi:

Você é um assistente prestativo. Sua tarefa é responder à pergunta:

Pergunta do Usuário:

{{ \$nodes.Webhook.json.body.pergunta }}

Dados Encontrados (em formato JSON):

{{ JSON.stringify(\$json) }}

Instruções:

1. Olhe a "Pergunta do Usuário".
2. Encontre a resposta dentro do JSON em "Dados Encontrados".
3. Formule uma resposta curta e direta em português.



- **Guardrails:** A lógica de "guardrails" foi implementada com um nó IF que verifica se a IA extratora classificou a pergunta como "fora\_de\_escopo", fornecendo uma resposta padrão

e interrompendo o fluxo.

---

## 4. Dificuldades Enfrentadas e Soluções

---

Durante o desenvolvimento, diversas dificuldades foram encontradas, servindo como uma grande oportunidade de aprendizado em depuração de sistemas distribuídos e na utilização de novas ferramentas.

- **Configuração do Ambiente Local e Docker:**

- O desafio inicial foi garantir a comunicação entre os contêineres e o terminal, solucionando um erro comum onde o Docker Desktop não estava em execução, o que impedia os comandos docker-compose de funcionarem.

- **Integração do n8n com a API (Depuração de Erros HTTP):**

- **Erro 404 Not Found:** Inicialmente, o workflow de IA não conseguia se comunicar com a API. A depuração mostrou que a URL no nó HTTP Request estava incompleta. A solução foi adicionar o caminho do endpoint (/eventos/) à URL base.
- **Erro 422 Unprocessable Entity:** Este foi o desafio mais persistente. A API rejeitava os dados enviados pelo n8n. A investigação passou por várias etapas:
  1. Suspeita de formato de data incorreto (DD/MM/AAAA vs AAAA-MM-DD).
  2. A descoberta de que dados "sujos" nas planilhas (textos como (Trimestral)) quebravam a conversão de data.
  3. A implementação de uma lógica de limpeza e transformação robusta utilizando o nó Code do n8n com JavaScript para garantir que os dados estivessem sempre no formato esperado pela API.
- **Falha na Avaliação da Expressão:** Em um estágio final, descobri através da ferramenta de inspeção do ngrok que o n8n não estava processando a expressão dinâmica na URL, mas a enviando como texto puro. A solução foi reconfigurar o nó HTTP Request para usar a seção dedicada de "Query Parameters", garantindo a correta avaliação da expressão.

- **Ponto Extra: Tentativa de Dockerizar o n8n:**

- Foi feita uma tentativa de cumprir o ponto extra de rodar o n8n localmente via Docker Compose. Embora a configuração do docker-compose.yml tenha sido bem-sucedida, o processo de criar credenciais do Google para um ambiente local (localhost) se mostrou excessivamente complexo e burocrático, envolvendo a configuração detalhada de um projeto no Google Cloud Platform, e problemas particulares com minhas credenciais, que levariam um tempo além do prazo do desafio para serem resolvidos.
  - Foi tomada a decisão estratégica de reverter para a arquitetura com o n8n Cloud e ngrok para garantir a funcionalidade completa do projeto dentro do prazo, documentando a tentativa como um aprendizado.
-

## 5. Uso de Inteligência Artificial no Desafio

---

Conforme a política de transparência do desafio, este projeto foi desenvolvido com o auxílio da Inteligência Artificial Gemini (Google). A colaboração com a IA foi utilizada como uma ferramenta de desenvolvimento assistido, para aprendizado durante o processo.

O uso da IA pode ser detalhado nos seguintes pontos:

- **Guia e Estruturação:** A IA foi utilizada para obter um passo a passo em pontos específicos que geraram mais dúvida durante o desafio.
- **Aprendizado de Ferramentas:** A interação serviu como uma ferramenta de aprendizado para aprofundar o conhecimento em tecnologias que não eram de domínio prévio, como o n8n, a criação de credenciais no Google Cloud Platform e as melhores práticas de depuração de workflows.
- **Depuração (Debugging):** A IA foi crucial no processo para diagnosticar e resolver uma série de problemas complexos, como erros de API (404 e 422), particularidades do ambiente n8n e a formatação de dados.
- **Geração e Refinamento de Código:** Foram solicitados snippets de código para a API em FastAPI, scripts SQL e, principalmente, expressões e código JavaScript customizado para os nós do n8n.

A IA foi empregada como uma parceira de desenvolvimento, permitindo um foco maior na lógica da solução e na superação dos desafios técnicos.

---