

Relatório Desafio Técnico

1. Arquitetura da Solução

A solução foi desenvolvida utilizando uma arquitetura moderna e modular, composta por três principais componentes:

- **Backend FastAPI:** Responsável por expor uma API REST para manipulação das agendas (IA, Marketing, RH) e servir como interface para automação.
- **Banco de Dados PostgreSQL:** Armazena todos os registros das agendas de forma estruturada e segura, além do histórico de conversa.
- **n8n:** Plataforma para implementar de forma automatizada a Migração de Dados e a Consulta com IA.

Todos os serviços são executados em containers Docker, o que facilita a configuração, o deploy e a escalabilidade.

2. Como Configurar e Rodar

No terminal, dentro do diretório do projeto, execute:

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

Isso iniciará os containers do PostgreSQL, FastAPI e n8n.

Acesso

- **API FastAPI:** <http://localhost:8000/docs> (swagger)
- **n8n:** <http://localhost:5678>

Guia de Testes

ETL

- Faça uma requisição POST para o link: <http://localhost:5678/webhook/68777c06-72ec-49fd-b0cd-ee87d6b47e51>
- Os dados das planilhas serão salvos no banco de dados.

Agente IA

- Faça uma requisição POST para o link abaixo, com o seguinte JSON no corpo:
<http://localhost:5678/webhook/4d32b57a-33ae-4ec9-aea1-7ba54a6b9496>

```
{  
  
  "prompt": "/SEU PROMPT AQUI/"  
  
}
```

- O agente de IA processará a solicitação e retornará a resposta.

3. Detalhes da Implementação de IA no n8n

a) Workflows

ETL (Extrair, Transformar e Carregar)

- O workflow inicia com um **Webhook** que recebe uma requisição externa.
- Três fluxos paralelos extraem dados de diferentes planilhas do **Google Sheets** (IA, Marketing e RH).
- Os dados são transformados e enviados via **HTTP Request** para os endpoints do backend FastAPI.
- O backend armazena as informações no banco de dados **Postgres**.

Agente IA

- O workflow começa com um **Webhook** que recebe um prompt.
- Um nó de **IA (AI Agent)** utiliza o modelo OpenAI **gpt-4.1-mini** e **memória em Postgres** para interpretar o prompt.
- O agente pode consultar, **criar ou atualizar eventos** nas agendas via ferramentas **HTTP (GET, POST, PATCH)** conectadas à API FastAPI.
- O IA Agent tem um “**System Message**” que indica o seu papel e como ele deve responder o prompt do usuário, além de contar com **guardrails** para evitar perguntas fora do escopo pré-estabelecido.

- A resposta é enviada de volta ao usuário via **Webhook**.

b) Prompt, Ferramenta e Lógica

- **Prompt:** O agente recebe instruções claras para responder apenas sobre eventos, usando dados reais da API, e manter respostas curtas e diretas.
- **Ferramenta:** O nó AI Agent utiliza o modelo gpt-4.1-mini para interpretar comandos e pode acessar endpoints da API para buscar, modificar ou criar dados.
- **Lógica:** O agente decide qual método usar (GET, POST, PATCH) conforme o tipo de solicitação do usuário, garantindo que as ações sejam realizadas sobre dados reais.

c) Workflows

- **Workflow ETL:** Faz a orquestração dos dados das três agendas, desde a entrada do Webhook até o armazenamento no backend.
- **Workflow Agente IA:** Faz a integração entre o Webhook, o agente de IA, o modelo de linguagem, a memória e as ferramentas de consulta e manipulação das agendas.

4. Dificuldades Enfrentadas

- **Validação de dados:** Garantir que todos os campos obrigatórios fossem enviados nas requisições, especialmente após mudanças nos modelos, tendo que alterar constantemente as funções em JavaScript dos nós do n8n.
- **Configuração das credenciais no n8n:** Algumas credenciais não são muito intuitivas de se obter (como a do google sheets), principalmente ao tentar rodar via docker.
- **Configuração do n8n via docker:** Foi necessário ajustar variáveis de ambiente, volumes e permissões para garantir o funcionamento correto, o que demorou mais do que o esperado, principalmente comparado à facilidade para rodar via cloud.
- **Documentação e clareza:** Tornar o projeto compreensível para pessoas não técnicas, através do README e do relatório.