

Teste Prático – Vaga: Cientista de Dados

1. Contexto

Desenvolver uma pequena aplicação de chat inteligente, composta por dois “sub-agentes”, que será consumida via API REST. O sistema deverá manter o histórico de conversas em banco SQLite para simular um diálogo contínuo. Um dos agentes vai tratar de assuntos de SAC (políticas de troca, devolução etc) e o outro agente vai tratar de buscar produtos no catálogo da empresa.

2. Objetivos do Exercício

1. Utilizar Framework Open-Source

- Escolher e configurar um framework de agentes.

2. Agente para Base de Conhecimento:

- Responde perguntas sobre a empresa usando Retrieval-Augmented Generation (RAG).
- Deve indexar e buscar em uma base de conhecimento fornecida (anexo a este documento).

3. Agente de Catálogo de Produtos:

- Recebe consultas de busca e retorna informações sobre produtos contidos em um arquivo fornecido (anexo a este documento)

4. API RESTful

- Criar endpoints para iniciar conversa, enviar mensagem do usuário, receber resposta do(s) agente(s) e recuperar histórico.

5. Persistência de Histórico

- Armazenar histórico no SQLite

3. Requisitos Técnicos

- Linguagem: Python
- LLM Local (qwen3:14b) ou API de mercado (OpenAI, Google Gemini)
- Qualquer framework Open Source

- Banco de Dados: SQLite (arquivo local)
- README com instruções de setup e uso da API
- Exemplos de requisições (cURL ou Postman collection)

4. Entregáveis

1. Código-fonte no GitHub (ou repositório ZIP):
2. README.md contendo:
 - Visão geral do projeto
 - Instruções de instalação e execução
 - Descrição dos endpoints e exemplos de uso
3. Arquivo de requisitos requirements.txt.

5. Observações Finais

- Use somente bibliotecas open-source e versões estáveis.
- Você poderá criar as ferramentas (tools) que julgar necessário para atingir o objetivo proposto.
- Certifique-se de que a aplicação rode localmente sem dependências externas complexas (exceto acesso à Internet para baixar pacotes).
- Valorize a clareza e a simplicidade na interface da API e no código.