

Desafio Técnico – Vaga de Desenvolvedor Python | C2S

Objetivo:

- Avaliar sua capacidade de aprendizado em tecnologias novas, e conhecimento em Python.

Etapas do desafio:

1. Modelagem de Dados

Crie um esquema para representar automóveis com, no mínimo, 10 atributos relevantes. Fique à vontade para incluir dados como marca, modelo, ano, motorização, tipo de combustível, cor, quilometragem, número de portas, transmissão, entre outros.

2. Popular o Banco com Dados Fictícios

Desenvolva um script que insira pelo menos 100 registros de veículos no banco de dados, utilizando dados simulados (fakes).

→ Utilize bibliotecas que simplifiquem e agilizem esse processo, como o **Faker**, **SQLAlchemy**, **Pandas**, entre outras de sua escolha.

3. Comunicação Cliente-Servidor via Protocolo MCP

- Implemente um **servidor** que receba requisições no protocolo MCP (Model Context Protocol), interprete os critérios enviados pelo cliente e consulte o banco de dados.
- O **cliente** deve ser capaz de enviar filtros (como marca, ano ou tipo de combustível), receber os resultados e exibi-los de forma organizada.
- Você precisa implementar a estratégia de Server e Client para fazer essa comunicação (não consultar direto pelo agente)
- O server vai receber a informação e fazer a consulta no banco de dados e retornar isso para o Cliente

4. Entrega Final – Aplicação no Terminal com Agente Virtual

Desenvolva uma aplicação que funcione diretamente no terminal, com a seguinte dinâmica:

- Ao ser executada, a aplicação inicia com um **agente virtual** interagindo com o usuário.
- O agente pergunta como pode ajudar.
- O usuário responde que está procurando um veículo.
- O agente faz perguntas para entender os critérios de busca, como:
 - Marca
 - Modelo
 - Ano mínimo ou máximo
 - Tipo de combustível
 - Faixa de preço, etc.
- Após coletar os filtros:
 - O **cliente MCP** envia os dados para o **servidor MCP**
 - O serviço MCP server consulta o banco de dados e retorna os veículos compatíveis.
- O agente recebe a resposta e exibe uma mensagem amigável com a lista de veículos encontrados, incluindo:
 - Marca
 - Modelo
 - Ano
 - Cor
 - Quilometragem
 - Preço

Recomendações Gerais:

- Utilize bibliotecas que funcionem bem e facilitem o desenvolvimento.
- Capriche na organização, legibilidade e clareza do código.
- Envie um repositório com instruções claras de como rodar cada parte da aplicação.

Entrega:

Prazo: 5.

Forma: Enviar o link do repositório (GitHub ou similar) por e-mail.