Desafio desenvolvedor Full-Stack

O setor de pré-vendas de uma grande companhia do ramo de Telecomunicações viu a necessidade de adquirir um desenvolvedor para criação de um Sistema de Viabilidades Nacional de vendas com seus parceiros.

Este é um sistema Full-Stack onde devemos ter um backend consolidado, performático e de resposta rápida.

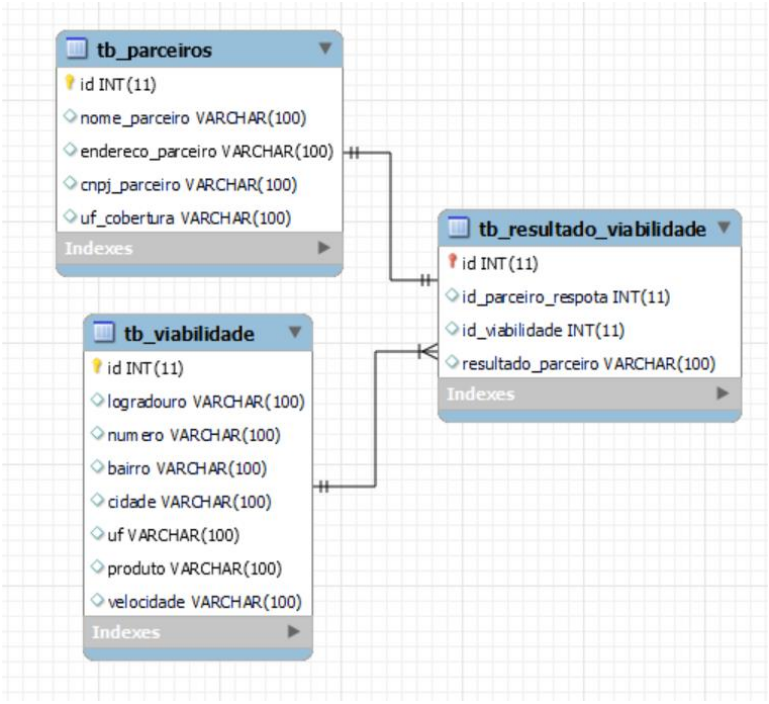
Todo seu código deve ser disponibilizado em um repositório remoto, contendo um arquivo README.md detalhando todas as etapas feitas e instruções de como executar o sistema backend e frontend.

Etapas de desenvolvimento:

1- Estrutura - Banco de dados:

1- Utilizando MySQL ou Postgres, crie a seguinte estrutura:

tb_parceiros – id, nome_parceiro, endereço_parceiro, cnpj_parceiro , uf_cobertura
tb_viabilidade – id, logradouro, numero, bairro, cidade, UF, produto, velocidade
tb_resultado_viabilidade – id, id_parceiro_respota, id_viabilidade , resultado_parceiro



1x1: tb_resultado_viabilidade X tb_parceiros
1xn: tb_viabilidade x tb_resultado_viabilidade
FK: id_tb_parceiros
FK: id_tb_viabiliadade
PK: id_tb_parceiros
PK: id_tb_viabilidade
PK: id_tb_resultado_viabilidade

Desafio desenvolvedor Full-Stack

2- Carga no Banco de dados

Segue em anexo planilha com dados de cada tabela para realizar a carga.

OBS: A carga não pode ser feita “na mão”, deve ser feita usando query SQL. Coloque no arquivo readme.md as querys usadas para fazer a carga no banco de dados.

3- Backend:

1- Crie uma API em Python: Você deve usar uma biblioteca para criação da API. No arquivo readme.md do repositório remoto (GitHub, bitbucket...), você deve explicar o por que da biblioteca escolhida comparado as demais principais de mercado.
Opções de bibliotecas: DJANGO, FLASK e FASTApi.

2- Você deve criar os seguintes métodos HTTPI: GET, POST, PUT e DELETE.
3- Utilizando o Insomnia ou Postman, simule as requisições, tire print da tela e coloque no arquivo readme.md os exemplos de requisições feitas.
    1- GET;
    2- POST;
    3- PUT;
    4- DELETE.

4- Frontend:

1- Crie uma SPA utilizando JS Puro ou algum Framework JS.
2- Sua SPA deve ter as sessões: Header, Main e Footer. Fique à vontade para criar o design. Você pode usar o Bootstrap ou similar caso tenha necessidade.
3- Na sessão Main você deve criar uma tabela que mostra os registros existentes no endpoint: resultado_viabilidade.

5- Documentação:

Você deve ter os seguintes tópicos no arquivo readme.md:

1- **Qual dos dois bancos (MySQL ou Postgres) você utilizou e o motivo;**
R:
2- Quais querys você usou para criar a estrutura do banco?
R:
3- **Quais querys você usou para fazer a carga no banco de dados?**
R:
4- Explique como você construiu o frontend da aplicação.
R:
5- Print do Insomnia ou Postman com o método GET da API;
6- Print do Insomnia ou Postman com o método POST da API;
7- Print do Insomnia ou Postman com o método PUT da API;
8- Print do Insomnia ou Postman com o método DELETE da API;