



Desafio Técnico | Back-end | Java Pleno

Seja bem-vindo(a) à nossa etapa de Teste Prático para a posição de Desenvolvimento Java!

Preparamos com muito carinho essa atividade para conhecer melhor você. Aqui, queremos observar suas habilidades de código e arquitetura, através da criação de uma API.

Criação da API

Você deve implementar uma aplicação Java utilizando o framework Spring que exponha dois endpoints para consulta dos dados no banco de dados:

- 1. Endpoint para consultar todos os veículos (uma listagem)**
 - Este endpoint deve retornar todos os veículos disponíveis no banco, sem retornar os pneus aplicados.
- 2. Endpoint para consultar um veículo específico (com pneus)**
 - Este endpoint deve retornar todas as informações de um veículo específico, incluindo os pneus aplicados e suas respectivas posições.
- 3. Endpoint para inserir um veículo específico**
 - Este endpoint deve inserir todas as informações de um veículo específico.
- 4. Endpoint para inserir um pneu específico**
 - Este endpoint deve inserir todas as informações de um pneu específico.
- 5. Endpoint para vincular um pneu em um veículo**
 - Este endpoint deve vincular um pneu com um veículo.
 - Lembre-se: um veículo não pode ter dois pneus na mesma posição.
- 6. Endpoint para desvincular um pneu em um veículo**

Para que a API retorne as informações, se faz necessário que você modele um banco de dados PostgreSQL contendo tabelas, dados e relacionamentos.

Algumas informações *essenciais* estão expostas a baixo:

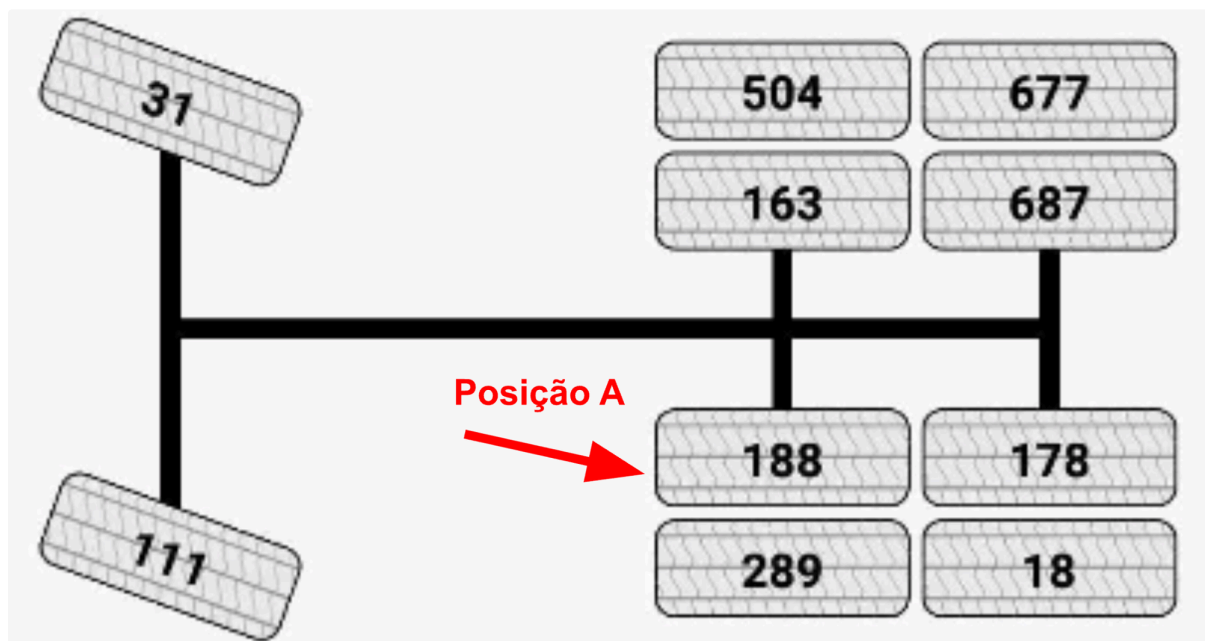
- Veículo
 - PLACA
 - MARCA
 - QUILOMETRAGEM (KM)
 - STATUS (ATIVO / INATIVO)
- Pneu
 - NÚMERO DE FOGO
 - MARCA
 - PRESSÃO ATUAL (EM PSI)
 - STATUS (ATIVO / INATIVO)

💡 Você **pode adicionar** mais informações se julgar necessário.

Além disso, você precisa mapear quais pneus estão aplicados em quais veículos e em qual posição. Cada posição de pneu no veículo deve ser identificada.

Exemplo:

Um veículo do tipo Truck, por exemplo, possui o pneu de número de fogo 188 na posição que chamamos de "A". Precisamos saber a posição específica de cada pneu em um veículo.



► Instruções:

1. Você deve modelar a solução utilizando Banco de Dados PostgreSQL;

2. O banco de dados deve estar populado com dados suficientes para executar o teste.
 3. Você deve fazer a população do banco de dados utilizando Flyway;
 4. Você deve utilizar JAVA e SpringBoot para criar o servidor;
 5. Retorne os dados da API em formato JSON;
 6. Você deve configurar o Docker para rodar, pelo menos, o **banco de dados**;
 7. Implemente testes, e2e ou unitários (os que julgar relevantes), para a aplicação JAVA;
 8. A utilização de JPA é opcional, mas será bem vista;
 9. Documente no README do projeto qualquer informação que achar relevante para facilitar a nossa compreensão sobre o teste e que **deixe simples para rodarmos**;
-

O que será avaliado:

Vamos observar as suas habilidades em:

- Implementar uma API funcional e bem estruturada;
- Conhecimento técnico nas ferramentas obrigatórias propostas no teste (Docker / Java / Spring / PostgreSQL / Flyway);
- Aplicação de conceitos SOLID e Clean Arch na elaboração do código;
- Uso de boas práticas de modelagem de banco de dados;
- Performance dos endpoints criados;
- Escrita de testes automatizados;
- Documentação de decisões técnicas e considerações relevantes;

Entrega das atividades:

1. Você deve hospedar o projeto de maneira privada no GitHub. Ao finalizar, compartilhe o projeto com os seguintes usuários do GitHub e envie o link do repositório:
 - gustavocnp95
 - bernardo2512
 - luizfp

Nossa sugestão é que você também sinalize à pessoa recrutadora que está em contato com você sobre a entrega da atividade. Assim garantimos que não haja ruídos de comunicação.



Prazo

Nosso prazo é de 3 dias corridos para a devolução da solução.

Boa sorte e bom trabalho! Estamos ansiosos para conhecer melhor suas habilidades! 💙