

Desafio Back-end

Introdução

Faaala Dev,

Nesse desafio você irá desenvolver de forma prática, um servidor utilizando **NodeJs com Typescript** para demonstrar suas habilidades.

Como se trata de um desafio, ele necessita de alguns conhecimentos de como criar uma API que lide com um banco de dados de sua preferência. Valorizamos sua autonomia para conseguir resolver os problemas propostos, e escolher tecnologias que o auxiliem no desenvolvimento. Por isso, lembre-se, **tenha calma e acredite no seu processo.**

Detalhes do desafio.

Nesse desafio, iremos desenvolver uma **API Rest de Pesquisa de Satisfação do Cliente.**

Suponha que você seja do time de desenvolvimento de uma empresa em ascensão de vendas online de roupas, pela grande diversidade de modelos e estampas de roupas, existe uma grande variedade de clientes que vai aumentando cada dia mais, como por exemplo Geeks e Nerds, Minimalistas, Atletas, Esportistas, etc.

Por conta desse cenário, surge a necessidade de pesquisa de satisfação do cliente para cada público específico, e a necessidade de salvar os registros de respostas a essas pesquisas realizadas; É aí que você entrará em ação e criará uma solução para atender essa demanda.

Requisitos técnicos

- Utilizar Typescript
- Aplicação deve expor um servidor rest
- Utilize os padrões que você se sentir mais confortável, mas mantenha o código limpo (Queremos que você faça o teste como você gostaria de trabalhar)
- Testes unitários (Não precisa ser algo complexo, apenas testar as principais funcionalidades da aplicação)

Casos de uso (Endpoint's)

1. Criar uma pesquisa de satisfação
 - Validar os dados enviados para gerar uma nova pesquisa.
 - Salvar as perguntas da pesquisa em um banco de dados.
 - Todas as pesquisas de satisfação devem ter ao menos as seguintes perguntas: **Público-alvo, Quantidade de estrelas e e-mail para contato.**
2. Atualizar uma pesquisa de satisfação

- Validar se a pesquisa que irá ser atualizada, ainda não existe na base.
 - Salvar uma data de atualização.
- 3. Preencher uma pesquisa de satisfação.
 - Validar se os dados enviados correspondem as perguntas cadastradas na pesquisa.
 - Salvar o preenchimento.
- 4. Listar os preenchimentos pelo **Público-alvo**.
 - Poder escolher se quer ordenar pela **Quantidade de estrelas** de forma crescente ou decrescente.

Diferencial

- Configuração de docker-compose para Banco de dados e API
- Clean architecture
- Implementar funcionalidade de exportar os preenchimentos em CSV

Entrega

- O tempo para realização do teste será de 3 dias corridos
- Após concluir o desafio, você deve nos enviar a URL do seu código no GitHub, para analisarmos o seu desenvolvimento.

Dê o seu melhor e dará certo! Boa Sorte!