**Challenge - Developer Backend**

Não é esperado que todos consigam realizar esse desafio por completo, já que é destinado a todos os níveis de carreira.

Você será avaliado pela sua capacidade de escrever um código simples, de fácil manutenção, e pela quantidade de funcionalidades que você entregar.

**Instruções**

* **Nome do Projeto:** Cantina do Ovelha
* **Objetivo do Projeto:** Criar uma API RESTful capaz de gerenciar os restaurantes e os produtos do seu cardápio.
* **Tecnologia:** Node.js com banco de dados relacional ou NoSQL.
* **Entregáveis:** Crie um repositório pessoal para esse projeto.

**Desafio**

* A sua API deverá ser capaz de:
  + Listar todos os restaurantes
  + Cadastrar novos restaurantes
  + Listar os dados de um restaurante
  + Alterar os dados um restaurante
  + Excluir um restaurante
  + Listar todos os produtos de um restautante
  + Criar um produto de um restaurante
  + Alterar um produto de um restaurante
  + Exlcuir um produto de um restaurante
* O cadastro do restaurante precisa ter os seguintes campos:
  + Foto do restaurante
  + Nome do restaurante
  + Endereço do restaurante
  + Horários de funcionamento do restaurante (ex.: De Segunda à Sexta das 09h as 18h e de Sabado à Domingo das 11h as 20h).
* O cadastro de produtos do restaurante precisa ter os seguintes campos:
  + Foto do produto
  + Nome do produto
  + Preço do produto
  + Categoria do produto (ex.: Doce, Salgados, Sucos...)
  + Quando o Produto for colocado em promoção, precisa ter os seguintes campos:
    - Descrição para a promoção do produto (ex.: Chopp pela metade do preço)
    - Preço promocional
    - Dias da semana e o horário em que o produto deve estar em promoção

**Formato de horários**

* É necessário tratar os campos que indicam horários de funcionamento e horário para as promoções dos produtos.
* Os campos devem possuir o formato HH:mm.
* Os horários devem possuir intervalo mínimo de 15 minutos.

**O que nós vamos avaliar**

* Você será avaliado pela qualidade do código, legibilidade e pela quantidade de funcionalidades implementadas.
* Você é livre para tomar as decisões técnicas com as quais você se sente mais confortável. Apenas esteja pronto para explicar as razões que fundamentaram suas escolhas =]
* Inclua um arquivo *README* que possua:
  + desafios/problemas com os quais você se deparou durante a execução do projeto.
  + maneiras através das quais você pode melhorar a aplicação, seja em performance, estrutura ou padrões.
  + todas as intruções necessárias para que qualquer pessoa consiga rodar sua aplicação sem maiores problemas.

**Dicas**

* Documente seu projeto em arquivos markdown explicando a estrutura, processo de setup e requisitos.
* Tenha sempre um mindset de usabilidade, escalabilidade e colaboração.
* A organização das branches e os commits no repositório falam muito sobre como você organiza seu trabalho.
* Os testes unitários são mais do que desejados.
* O design/estrutura do código da aplicação deve ser *production ready*.

Use boas práticas de programação.

**FAQ**

**Posso utilizar frameworks/bibliotecas?**

Sim.

**Quanto tempo eu tenho ?**

Quanto mais tempo você demorar, mais críticos seremos na sua avaliação =]

**Banco de Dados Relacional ou NoSQL?**

Você pode escolher qualquer uma delas. Não queremos te influenciar, mas optar por MySQL ou NoSQL seria uma boa :)

**Happy coding**