Teste para Desenvolvedor(a) Fullstack

1. Objetivo

O objetivo deste teste é montar uma API para gerenciar um cardápio online. E mostrar os produtos de um cardápio no frontend

2. Tecnologias

- Typescript
- Backend
    - NodeJS
    - NestJS (https://nestjs.com)
    - PostgreSQL ou MySQL
    - TypeORM
- Frontend
    - React (create react app)
    - React router dom
    - SASS
- Docker (bonus)

3. Requisitos

3.1 Backend

Você deverá desenvolver as seguintes endpoints:

Autenticação

- **POST /auth/login**: para que os administradores do restaurante acessem o sistema. Este método devolverá um token para ser usado no header Authentication das outras requisições.

Categoria (id: string, parent: Category|null, name: string)

- **GET /category:** para listar todas as categorias de produtos
- Obs.: Adicione as categorias diretamente no banco de dados.

Produto (id: string, categories: Category[], name: string, qty: number, price: number, photo: string)

- **GET /product:** para listar todos os produtos

- **GET /product:id:** para pegar um produto
- **POST /product:** para criar um produto novo
- **PATCH /product/:id:** para alterar um produto
- **DELETE /product:id:** para excluir um produto

3.2. Frontend

- Listagem de Produtos
    - Na versão desktop, deverá ser um layout em 4 colunas, com um component card para cada um dos produtos. Conforme diminui a tela, diminuir a quantidade de colunas, até chegar em 1.
    - Precisa ter um link para os detalhes do produto
- Detalhes do produto
    - Página com visualização do produto selecionado responsiva

Nota: O frontend deve estar integrado a API (backend)

3.3. Requisitos e Dicas

- Seu código deve usar os nomes de classes, métodos e variáveis em inglês
- Você não precisa criar mais campos do que o indicado acima para cada uma das categorias. Esse projeto não será usado em produção, mas será usado para avaliar como você programa.
- Faça commits/pushes frequentemente, precisamos saber se você sabe usar o Git.
- Você pode acessar os sites de delivery como iFood para ver como é uma página de cardápio e inspirar-se.
- Crie a estrutura de pastas, páginas e navegação do seu projeto. Lembre-se que será outra pessoa que vai avaliar o seu código. Quanto mais organizado, mais fácil e melhor será a avaliação.
- A utilização do Docker será contada como um diferencial e o principal critério de desempate.
- Você pode usar bibliotécas ou frameworks da sua escolha, desde que você use as tecnologias da seção 2.
- Ficou com alguma dúvida, envie um e-mail para diogojpina@yahoo.com.br

4. Entregáveis

1. Uma URL do github com o projeto (se o projeto for privado deve dar acesso para o usuário diogojpina). O repositório deve ter duas pastas: backend e frontend.
2. O arquivo README do seu projeto deve conter:
    a. Instruções para rodar o projeto localmente (backend e frontend)
    b. Tecnologias/Frameworks/Bibliotecas usadas e a razão da escolha delas
    c. Qualquer outra informação importante para o projeto.
3. Após terminar o teste, envie um e-mail para diogojpina@yahoo.com.br

5. Avaliação

Os critérios que serão utilizados para avaliar o seu projeto são:

1. Cumprimento dos requisitos
2. Qualidade e legibilidade do código
3. Escolhas feitas em relação a tecnologias, arquitetura, padrões, schemas das entidades, etc.
4. Usabilidade do sistema
5. Foi além dos requisitos (adicionou mais recursos relevantes, fez alguma lógica de cache, testes automatizados, etc)