



## DESAFIO BACKEND (NODE.JS)

### Objetivo

Criar uma API RESTful para gerenciar um sistema simples de pedidos e produtos. Nosso objetivo é avaliar a construção de um backend organizado, seguro e funcional.

### Requisitos

- Utilize **Node.js** com **Express**.
- Utilize **MySQL** (Prisma).
- A API deve permitir a criação, leitura, atualização e remoção (CRUD) de **usuários, produtos e pedidos**.
- Implementar autenticação de usuários utilizando **JWT**.
- Aplicar **validação de dados** nas requisições.
- Criar um **arquivo README** com instruções de execução.

### Descrição das Entidades

#### Usuário

- id (UUID ou ObjectId)
- name (string, obrigatório)
- email (string, obrigatório e único)
- password (string, obrigatório, armazenada de forma segura)
- created\_at (timestamp, default: agora)

#### Produto

- id (UUID ou ObjectId)
- name (string, obrigatório)
- decription (string, opcional)

- price (número, obrigatório)
- stock (número, obrigatório)
- created\_at (timestamp, default: agora)

### **Pedido**

- id (UUID ou ObjectId)
- user\_id (referência ao Usuário, obrigatório)
- products (array de objetos contendo product\_id e quantity)
- total (calculado com base nos produtos e quantidades)
- status ("Pendente", "Processando", "Enviado", "Entregue")
- created\_at (timestamp, default: agora)

### **Funcionalidades**

#### **1. Autenticação e Gestão de Usuários**

- Registro de usuário (criptografando a senha)
- Login de usuário (retorna um token JWT)
- Consulta de dados do próprio usuário (autenticado)

#### **2. Gestão de Produtos**

- Criar, listar, atualizar e remover produtos (autenticado)

#### **3. Gestão de Pedidos**

- Criar pedido (apenas usuários autenticados)
- Listar pedidos do usuário
- Atualizar status do pedido (somente administradores, se implementado)

### **Diferenciais (Opcional, mas um plus)**

- Implementar **Swagger/OpenAPI** para documentação.
- Criar **testes automatizados** (Jest ou outra ferramenta) - o quanto você puder fazer - não existe uma cobertura mínima.
- Utilizar **Docker** para facilitar a execução do projeto.

## Entrega

- O candidato deve subir o código para um **repositório público no GitHub ou GitLab**.
- O repositório deve conter um **README.md** com:
  - Passos para executar o projeto localmente.
  - Exemplos de chamadas para a API.
  - Informações sobre tecnologias utilizadas.

Atenciosamente,  
**Rodrigo Rocha**  
Gerente de Tecnologia