**CRUD com Entity Framework Core, Front-end em React.js e TDD**

**Objetivo:**

Este exercício tem como finalidade avaliar o seu nível de conhecimento em diferentes temas relacionados ao desenvolvimento de software.

Caso não seja possível completar algum item, justifique se a dificuldade foi por falta de tempo ou por problemas com o tema específico.

**Descrição do Exercício:**

Você deve criar uma aplicação que permita exibir uma lista de produtos, incluindo nome, preço e descrição. Além disso, a aplicação deve permitir as operações de cadastrar, editar, atualizar e excluir produtos dessa lista.

O desafio está dividido em duas etapas: Back-end e Front-end. Você pode iniciar pela etapa de sua escolha.

**Requisitos Técnicos:**

1. *Back-end:*

* Implementar as operações CRUD (Create, Read, Update, Delete) utilizando o Entity Framework Core.
* Escrever testes automatizados para o back-end seguindo a prática de Desenvolvimento Orientado por Testes (TDD).

1. *Front-end:*

* Criar uma interface de usuário utilizando React.js.

**Entrega:**

O código desenvolvido deve ser colocado em um repositório público no GitHub. O link para o repositório deve ser compartilhado através do e-mail: [msenra@cygnussoftware.com.br](mailto:msenra@cygnussoftware.com.br).

Boa sorte e bom trabalho! 😊

**1° Etapa - Back-end:**

1. **Modelo de Dados:**

* Crie um modelo “*Product*” com as propriedades “*Id*”, “*Name*”, “*Price*” e “*Description*”.

1. **Contexto de Dados:**

* Configure um contexto de dados usando o Entity Framework Core.

1. **Controlador API:**

* Adicione um controlador API chamado “*ProductsController*” que implemente as ações CRUD para o modelo “*Product*”.

1. **Configurações:**

* Utilize migrations para criar o banco de dados.
* Adicione as dependências necessárias para o Entity Framework Core no “*Startup.cs*”.

1. **Testes Automatizados:**

* Crie um projeto de testes para o back-end.
* Escreva testes unitários para o controlador “*ProductsController*” utilizando um framework de testes como xUnit.
* Utilize o Moq para criar mocks do contexto de dados.
* Implemente testes para verificar as operações CRUD.

**2° Etapa - Front-end:**

1. **Configuração do Projeto React:**

* Crie um projeto React.js

1. **Componentes:**

* Crie os seguintes componentes React:
  + “*ProductList*”: para listar todos os produtos.
  + “*ProductForm*”: para adicionar ou editar um produto.
  + “*ProductDetail*”: para exibir detalhes de um produto específico.

1. **Serviços de API:**

* Crie um serviço para consumir os endpoints da API criada no back-end usando “*axios*” ou “*fetch*”.

1. **Operações CRUD:**

* Implemente as operações CRUD na interface React:
  + Listar produtos: obtenha os dados do endpoint “*GET /api/products*” e exiba os produtos na interface.
  + Adicionar produto: utilize o endpoint “*POST /api/products*”.
  + Editar produto: utilize o endpoint “*PUT /api/products/{id}*”.
  + Deletar produto: utilize o endpoint “*DELETE /api/products/{id}*”.

1. **Estilo:**

* Adicione estilos básicos para uma interface amigável, utilizando CSS ou uma biblioteca como Material-UI.