



Transformação da SPA em React com C#

1. Trabalho Prático: Transformação da SPA em React com C#

Neste trabalho prático, você será desafiado a transformar a Single Page Application (SPA) desenvolvida com JavaScript puro no trabalho anterior em uma aplicação web utilizando React no front-end e C# no back-end. Este projeto tem como objetivo proporcionar a você uma introdução ao desenvolvimento com React e ao uso do C# para construir um web service simples.

2. Objetivo do Trabalho

O objetivo deste trabalho é permitir que você ganhe experiência na criação de uma aplicação web utilizando React no front-end e C# no back-end. Você irá refatorar a SPA desenvolvida anteriormente para utilizar React e implementar um web service em C# para manipular os dados da aplicação.

3. Descrição do Projeto

Você deverá transformar a aplicação SPA desenvolvida com JavaScript puro no trabalho anterior em uma aplicação web utilizando React no front-end e C# no back-end. A aplicação continuará a realizar operações CRUD em uma entidade escolhida por você.

4. Requisitos Técnicos

1. Front-end com React:

- Refatorar a interface da aplicação para utilizar React.
- Utilizar componentes do React para estruturar a interface de forma modular.
- Manter as funcionalidades de CRUD para a entidade escolhida.

2. Back-end com C#:

- Implementar um web service utilizando C# e .NET.
- Oferecer endpoints para realizar as operações CRUD na entidade escolhida.
- Utilizar um banco de dados In Memory do .NET para persistir os dados.

5. Entrega do Trabalho

Para entrega do trabalho, siga os seguintes passos:

1. Desenvolvimento da Aplicação Web:

- Refatore a aplicação SPA para utilizar React no front-end.
- Implemente um web service em C# para oferecer as operações CRUD no back-end.

2. Hospedagem no GitHub:

- Hospede a sua aplicação web no GitHub.

3. Envio do Link:

- Envie o link da sua aplicação web ao professor até a data de entrega especificada.

6. Avaliação do Trabalho

O trabalho será avaliado com base nos seguintes critérios:

1. Funcionalidade: A aplicação deve funcionar corretamente, oferecendo todas as operações CRUD para a entidade escolhida.
2. Integração React e C#: O front-end em React e o back-end em C# devem estar devidamente integrados, permitindo a comunicação entre as partes.
3. GitHub Repository: O web service deve estar hospedado em um repositório no GitHub.
4. README.md: O README.md deve conter evidências da aplicação funcionando por meio de prints de tela.

PS: Implementações oriundas de plágio ou geradas por Inteligência Artificial podem ser facilmente detectadas. Em um desses casos, a nota do trabalho será zerada.