



Trabalho Prático: Web Services e Containers

1. Objetivo do Trabalho

Neste trabalho prático, você será desafiado a implementar um web service utilizando C# que permita a persistência e manipulação de dados para a Single Page Application (SPA) desenvolvida no trabalho anterior, construída com React. O objetivo é integrar a parte back-end da aplicação, utilizando princípios SOLID e, se possível, aplicando Clean Architecture para criar um sistema robusto e de fácil manutenção.

2. Descrição do Projeto

O objetivo deste trabalho é proporcionar a você a oportunidade de aplicar seus conhecimentos em desenvolvimento web back-end utilizando C# e .NET, implementando um web service que ofereça operações CRUD para manipulação de dados. Além disso, você será desafiado a compilar o seu projeto via Docker e hospedá-lo em um repositório no GitHub.

Você deverá implementar um web service utilizando C# e .NET que ofereça endpoints para realizar as operações CRUD em pelo menos uma entidade vinculada à sua SPA desenvolvida com React. O web service deverá ser capaz de persistir os dados em um banco de dados In Memory do .NET.

Requisitos Técnicos

- C# e .NET: Utilizar C# e .NET para desenvolver o web service.
- Persistência de Dados: Implementar a persistência de dados utilizando um banco de dados In Memory do .NET.
- Operações CRUD: Oferecer endpoints para realizar as operações CRUD na entidade vinculada.

- Princípios SOLID: Aplicar os princípios SOLID durante o desenvolvimento do web service.
- DockerFile: Fornecer um DockerFile para compilar a sua SPA e o seu web service via Docker.

GitHub Repository

Após concluir o desenvolvimento do seu web service, você deve hospedá-lo em um repositório no GitHub.

README.md

No README.md do seu repositório, forneça evidências da sua aplicação funcionando por meio de prints de tela da mesma.

3. Tema da Aplicação

O tema da aplicação deve ser uma extensão do tema utilizado no trabalho anterior. Utilize a apostila do Módulo 3 da disciplina para implementar o seu webservice. Sobre Clean Architecture, você pode se instruir neste curso:

https://www.youtube.com/watch?v=ZWfrl5Bu6so&list=PLJ4k1IC8GhW3GICba2dLmiTZrVPw0St_hC

4. Entrega do Trabalho

Para entrega do trabalho, siga os seguintes passos:

- Desenvolvimento do Web Service: Implemente o web service utilizando C# e .NET, aplicando os princípios SOLID e, se possível, a Clean Architecture.
- DockerFile: Forneça um DockerFile para compilar o projeto via Docker.
- Hospedagem no GitHub: Hospede o seu web service em um repositório no GitHub.
- README.md: No README.md do seu repositório, forneça evidências da sua aplicação funcionando por meio de prints de tela.

5. Avaliação do Trabalho

O trabalho será avaliado com base nos seguintes critérios:

1. Funcionalidade: O web service deve funcionar corretamente e oferecer todas as operações CRUD para as entidades vinculadas.
2. Princípios SOLID: O código do web service deve seguir os princípios SOLID, garantindo uma arquitetura limpa e de fácil manutenção.
3. DockerFile: O projeto deve dispor de um DockerFile e ser compilado via Docker.

4. GitHub Repository: O web service deve estar hospedado em um repositório no GitHub.
5. README.md: O README.md deve conter evidências da aplicação funcionando por meio de prints de tela.

6. Conclusão

Este trabalho prático oferece a você a oportunidade de aplicar seus conhecimentos em desenvolvimento web back-end utilizando C# e .NET. Ao implementar um web service que oferece operações CRUD para a sua SPA desenvolvida com React, você estará integrando as partes back-end e front-end da sua aplicação, proporcionando uma experiência completa aos usuários. Além disso, a aplicação dos princípios SOLID e a compilação via Docker contribuirão para a criação de um sistema robusto e de fácil manutenção.

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