

**ULBRA - UNIVERSIDADE LUTERANA DO BRASIL**  
**CURSO DE ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**KAILAN DE SCENA PACHECO**

**APLICAÇÃO WEB API UTILIZANDO .NET 8**

**TORRES, RIO GRANDE DO SUL**

2024

## SUMÁRIO

RESUMO

INTRODUÇÃO

DESENVOLVIMENTO.....

3.1 CONCEITOS FUNDAMENTAIS EM APIs RESTful .....

3.2 ORGANIZAÇÃO DO CÓDIGO COM O PADRÃO REPOSITORY.....

3.3 EFICIÊNCIA COM O ENTITY FRAMEWORK CORE.....

3.4 IMPLEMENTAÇÃO PRÁTICA NO PROJETO.....

CONCLUSÃO

REFERÊNCIAS

## **1. RESUMO**

Este trabalho apresenta as etapas de desenvolvimento de uma API RESTful utilizando o .NET 8. O foco foi aplicar conceitos fundamentais de desenvolvimento de APIs, incluindo organização com o padrão Repository e o uso do Entity Framework Core para persistência de dados. A API foi projetada para gerenciar pedidos e fornecedores, seguindo boas práticas que garantem modularidade, eficiência e facilidade de manutenção. Também foi adotado o SQLite como banco de dados para simplificar o desenvolvimento e garantir uma implementação leve e funcional.

## **2. INTRODUÇÃO**

A criação de APIs RESTful é uma prática essencial para o desenvolvimento de sistemas modernos. Elas permitem a comunicação eficiente entre clientes e servidores, promovendo integração e escalabilidade. Neste contexto, a adoção de boas práticas é fundamental para garantir a robustez e a longevidade das aplicações. Este projeto utilizou tecnologias como o .NET 8 e o Entity Framework Core para desenvolver uma API RESTful que gerencia pedidos e fornecedores. O trabalho explora a aplicação de conceitos teóricos e práticos, como o uso de padrões de projeto e ferramentas modernas para persistência de dados.

### **3. DESENVOLVIMENTO**

#### **3.1 CONCEITOS FUNDAMENTAIS EM APIs RESTful**

APIs RESTful seguem princípios como a utilização de verbos HTTP de forma semântica (GET, POST, PUT e DELETE), estruturas claras de URL e o retorno de códigos de status apropriados. A API desenvolvida aplicou essas diretrizes, garantindo que cada endpoint cumprisse sua função de forma clara e eficiente.

#### **3.2 ORGANIZAÇÃO DO CÓDIGO COM O PADRÃO REPOSITORY**

O padrão Repository foi implementado para centralizar o acesso a dados e desacoplar a lógica de negócios das operações de persistência. Isso permitiu que as entidades Pedido e Fornecedor fossem gerenciadas de forma modular, facilitando a manutenção e testes do sistema.

#### **3.3 EFICIÊNCIA COM O ENTITY FRAMEWORK CORE**

O Entity Framework Core foi utilizado para mapear as entidades da aplicação para o banco de dados SQLite. A utilização de consultas assíncronas garantiu um desempenho superior e evitou bloqueios durante operações de leitura e escrita. O ApplicationDbContext foi configurado para gerenciar todas as interações com o banco, promovendo uma integração transparente.

#### **3.4 IMPLEMENTAÇÃO PRÁTICA NO PROJETO**

A implementação seguiu as diretrizes definidas, com endpoints organizados para manipular pedidos e fornecedores. O uso do Swagger foi essencial para documentar os endpoints e permitir testes interativos, facilitando a integração com outros sistemas. O SQLite garantiu um ambiente de banco de dados leve e eficiente para o desenvolvimento local.

#### **4. CONCLUSÃO**

O projeto demonstrou como boas práticas e ferramentas modernas podem ser aplicadas para desenvolver uma API RESTful funcional e escalável. A separação de responsabilidades com o padrão Repository e o uso do Entity Framework Core garantiram uma arquitetura limpa e eficiente. A API criada não apenas atende aos requisitos funcionais, mas também proporciona uma base sólida para futuras expansões e melhorias.

## 5. REFERÊNCIAS

1. RestfulAPI.net. Práticas recomendadas para APIs RESTful. Disponível em: <https://restfulapi.net/>
2. Microsoft Docs. Boas práticas de desenvolvimento no .NET. Disponível em: <https://learn.microsoft.com/pt-br/>
3. Microsoft Docs. Entity Framework Core - Documentação oficial. Disponível em: <https://learn.microsoft.com/pt-br/ef/core/>
4. Fielding, R. T. (2000). Architectural Styles and the Design of Network-based Software Architectures. University of California, Irvine. Disponível em: [https://www.ics.uci.edu/~fielding/pubs/dissertation/rest\\_arch\\_style.htm](https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm)
5. Microsoft Docs. Swagger e OpenAPI no ASP.NET Core. Disponível em: <https://learn.microsoft.com/pt-br/aspnet/core/tutorials/getting-started-with-swashbuckle>