

# API DE PRODUTOS – DTI DIGITAL

---

Em atendimento aos requisitos fornecidos, foi desenvolvido um sistema para controle de estoque de uma empresa contendo uma tela de listagem de produtos, onde é possível acessar as informações ou excluir um produto. Uma tela para cadastro e uma tela para edição dos mesmos.

Autor: Rayne Bruno Gomes – [raynegomes@gmail.com](mailto:raynegomes@gmail.com)

GitHub do projeto: <https://github.com/raynegomes/teste-DTIDigital>

Criado em: 08/12/19

## Especificações Técnicas

---

### Backend:

Tecnologias utilizadas:

- Dotnet Core V.3.1: utilizado devido ser a mais recente versão do framework.
- Modelagem de arquitetura em DDD (Domain Drive Design): Utilizado a modelagem DDD devido ao desacoplamento de camadas do projeto, além de fornecer separação de responsabilidades de cada camada com injeção de dependências, o que vem a facilitar a evolução e modificações no código e estrutura do projeto, também bastante recomendado como Boas Práticas de desenvolvimento.
- Mapeamento DTO com AutoMapper: feito o mapeamento DTO dos objetos para facilitar a visualização de informações nas respostas exibidas na API, evitando assim que sejam expostas informações internet ou desnecessárias, além de fornecer validação inicial de inserção de informações através de Data Annotations, o que evita processamento desnecessário para validações mais básicas.
- Swagger v.5: utilizado a biblioteca Swashbuckle.AspNetCore para fornecer a documentação da API.
- Banco de Dados MySQL v.8: utilizado base de dados MySQL na versão 8, devido ser um banco robusto e gratuito que utiliza a linguagem SQL como interface.
- EFCore v3.1(Entity Framework Core): ORM para banco de dados relacional. Utilizado devido a facilidade para utilização com vários modelos de bancos relacionais (MySQL, MsSql), sendo possível fazer a troca de base de dados com mais agilidade e facilidade.
- xUnitTest: criado a Solucao de teste utilizando a xUnit, devido ser uma das ferramentas de teste unitário mais leves e robustas para projetos dotnet (não implementado testes devido ao tempo proposto para desenvolvimento, projeto pode ser uma evolução do projeto proposto).

### Frontend:

- React 16.12: utilizado a biblioteca React para desenvolvimento do frontend, devido ser de fácil uso, manutenção e com boa performance para aplicações Web.

- React Icons 3.8: biblioteca de ícones para aplicações com React.
- Axios 0.19: biblioteca para consumo de API com javascript.
- Eslint 6.7: ferramenta para análise de códigos Javascript baseado nas melhores práticas.
- Prettier 1.19: ferramenta para padronização e formatação de códigos.
- Styled Component 4.4: ferramenta para estilização de componentes, garantindo cross-browser da aplicação.
- React Tostyfy 5.4: componente para exibição de mensagens de alertas e avisos.