Documento de Desenho Arquitetural

Descritivo da Solução

[https://github.com/laszloataliba/xpe-desafio-final5](https://github.com/laszloataliba/xpe-pos-projeto-final/tree/develop)

A Solução deverá proporcionar acesso via API Rest para consulta, e manipulação de dados, para clientes internos e parceiros, do domínio “Produto”.

O Elemento “Produto” deverá possuir as seguintes características:

**Id** = Guid = Representando o código identificador único do produto no banco de dados

**Name** = String = Representando o nome do produto no banco de dados

**Price** = Decimal com 2 casas = Representando o preço do produto no repositório

**EAN** = String = Representando o código de barra do produto no repositório.

A API deverá ser do tipo REST Full e deverá conter os seguintes métodos públicos expostos:

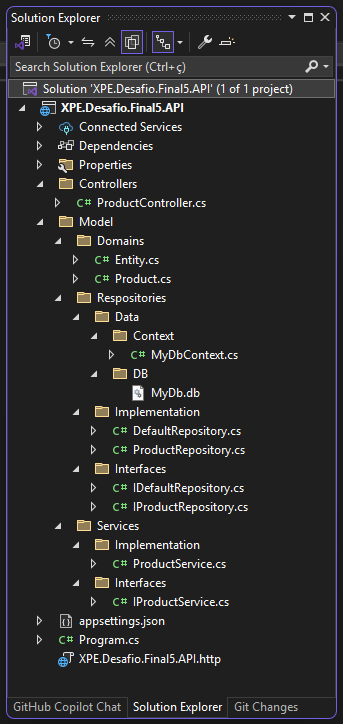
* Path: /api/produto
* **getAll** = Lista todos os Produtos cadastrados = verbo GET (api/products)
* **getByName** = Lista os Produtos com similaridade de nome = verbo GET (api/products/{product-name}
* **getById** = Retorna 1 Produto com o ID indicado = verbo GET (products/{product-id})
* **create** = Cadastro do produto na base de dados = verbo POST (api/products) (requestBody raw – json)
* **update** = Alteração dos dados do produto na base de dados = verbo PUT (api/products) (requestBody raw – json)
* **delete** = Exclui um registro com o ID informado = verbo DELETE (api/products/{product-id})
* **total** = Retorna a quantidade de registros cadastrados = verbo GET (api/products/total)

Persistência

A persistência será realizada por meio de banco de dados simples e local, SQLite(em arquivo) para a manutenção dos dados.

./Model/Repositories/Data/DB/MyDb.db

Estrutura de Pastas



* aplicação (asp.net web api) – repositório raiz da aplicação
  + controllers = controla o fluxo da aplicação
  + model = executa o código de serviços / persistência e domínio
    - domains = armazenar os domínios/entidades da aplicação
    - repositories = armazenar os repositórios/classes de persistência da aplicação além do banco de dados local.
  + services = armazenar as classes de serviço / lógica de negócio da aplicação.

Diagramas

**Diagrama de Casos de Uso**

**Diagrama

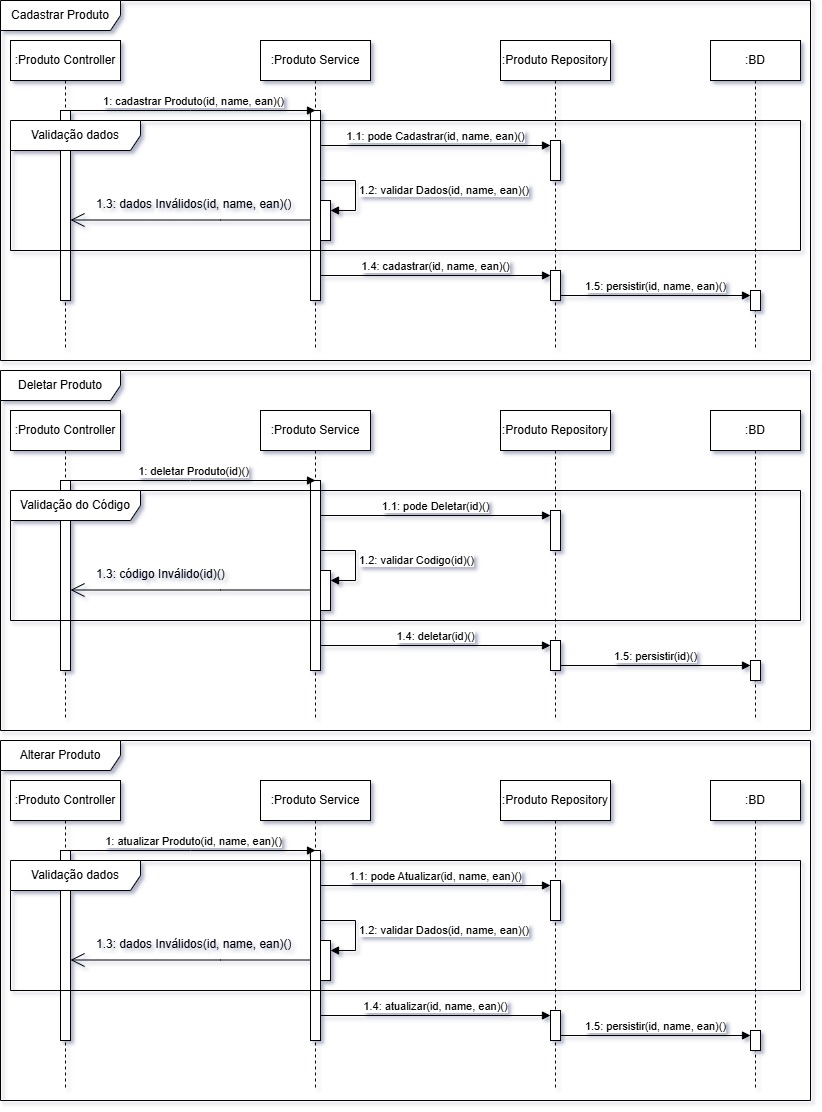
O conteúdo gerado por IA pode estar incorreto.**

**Diagrama de Classes**

**Interface gráfica do usuário, Texto

O conteúdo gerado por IA pode estar incorreto.**

**Diagramas de Sequência**

****

**Diagrama, Desenho técnico

O conteúdo gerado por IA pode estar incorreto.**