**Lista exercício 3 – Desenvolvimento Web**

**Regras:**

**- Individual a entrega via github é obrigatório.**

**- Data entrega no portal**

**Parte 1 – Teórica:**

**Questão 1:**

Em sala no início da disciplina criamos e comentamos que nossa API estava com muitas responsabilidades( regras de negócio , validações, persistência) , ferindo assim qual princípio? Descreva o princípio e o que a partir de então começamos a fazer para corrigir isso?

**R:** O princípio da "Separação de Responsabilidades" ou o princípio da "Única Responsabilidade".

Para corrigir, deve reorganizar o código, dividindo as responsabilidades em componentes separados e garantindo que cada componente tenha uma única responsabilidade bem definida.

**Questão 2:**

Começamos a dividir nosso projeto em camadas , são elas:

Domain, Data, Application e API. Descreva qual o papel de cada uma delas.

**R:**

**Domain**: O domínio se refere a área de conhecimento em que um software está inserido. Ele descreve as regras, conceitos e elementos relacionados a esse contexto específico. O domínio ajuda a definir o escopo e os requisitos do sistema, garantindo que ele atenda às necessidades do ambiente em que será usado.

**Data**: Os dados são informações que são armazenadas, processadas e utilizadas pelo sistema de informação. Eles representam fatos, números, textos e outros tipos de informações relevantes para o domínio. Os dados podem ser estruturados (como em um banco de dados relacional) ou não estruturados (como arquivos de texto, imagens, vídeos, etc.). O papel dos dados é fornecer a matéria-prima para a aplicação, permitindo que ela realize suas operações e forneça resultados.

**Application**: A aplicação é o software que é desenvolvido para resolver problemas ou automatizar processos dentro de um domínio específico. Ela utiliza os dados do domínio para executar funções, processar informações e apresentar resultados aos usuários ou outros sistemas. A aplicação é responsável por implementar a lógica de negócios e as funcionalidades necessárias para atender às necessidades do domínio. Pode ser uma aplicação web, desktop, mobile ou qualquer outra forma de software.

**API**: Uma API é uma interface que permite a comunicação e interação entre diferentes sistemas de software. Ela define um conjunto de regras e protocolos que permitem que aplicativos solicitem e compartilhem dados ou funcionalidades uns com os outros. As APIs podem ser usadas para integrar sistemas, permitir a expansão de funcionalidades, acessar serviços de terceiros e muito mais.

**Questão 3:**

Na camada de Domain criamos classes cujas propriedades são com set privado. Descreva, vantagem de usar dessa forma destacando como fizemos em sala com o produto.

**R:** São elas algumas vantagens: Controle sobre os dados, maior encapsulamento, validações e regras de negócio, segurança, etc. Em resumo, usar setters privados em classes do domínio é uma prática recomendada na programação orientada a objetos, pois promove um encapsulamento mais forte, permite a aplicação de validações e regras de negócio, e ajuda a manter a flexibilidade e a integridade do código.

**Questão 4:**

Na camada de applicattion na classe service (de serviço) fizemos o que chamamos de injeção de dependência, descreva por que utilizamos essa técnica e como isso pode ser uma vantagem?

**R:** é uma técnica usada para gerenciar as dependências externas que essas classes precisam. Vantagens: Desacoplamento, testabilidade, reutilização do código, legibilidade e manutenção, etc.

**Parte 2- Prática**

**Questão 1**

Desenvolver na API de produtos feita em aula duas classes a seguir:

Classe categoria: Código, Descrição

Classe fornecedor: Código, Razão social, CNPJ, Ativo, Data Cadastro, Email Contato.

Criar o Domain das duas classes, Repositório, Application, e Controller na API. Nas ações faça as operações de crud (Criar, Atualizar, Remover, Buscar por código e buscar todos).