Nome: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Repositório: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Nota: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Gerenciamento de Produtos**

**Objetivo:** Desenvolver um sistema de gerenciamento de produtos em PHP, utilizando uma arquitetura que separe a camada de visualização da regra de negócio, e fornecer uma API REST para acesso aos dados.

**Requisitos Funcionais:**

* Cadastro de produtos com as seguintes informações: Código do Produto, Nome do Produto, Tipo de Produto (Smartphone, Tablet, etc), Data de Criação e Data da Última Atualização.
* Atualização de produtos existentes.
* Exclusão de produtos.
* API REST para acesso aos dados de produtos.

**Tecnologias Utilizadas:**

* Frontend (opcional, Bootsrap, Tailwind, Materialize, etc);
* Backend (PHP);
* MySQL (Banco de Dados);
* Framework PHP (opcional, Laravel, Codeigniter, etc.)
* API REST (opcional, Laravel API Resource, Slim Framework, etc.)

**Estrutura do Projeto:**

* **Camada de Visualização:** Interface de usuário para interação com o sistema. Pode ser implementada utilizando HTML, CSS e JavaScript.
* **Camada de Serviço:** Lógica de negócio e operações de CRUD (Create, Read, Update, Delete) dos produtos.
* **Camada de Persistência:** Responsável por acessar e manipular os dados no banco de dados MySQL.
* **API REST:** Implementação da API REST para permitir o acesso aos dados dos produtos.

**Funcionalidades Adicionais (Opcionais):**

Autenticação da API.

Validação de entrada de dados.

**Observação:**

Não esqueça de disponibilizar o repositório público contendo a estrutura do projeto, incluindo pastas para frontend, backend, documentação de requisitos e scripts de banco de dados, entre outros.