**Avaliação Técnica**

1. **Objetivo**

O objetivo principal desta avaliação é determinar o nível de conhecimento do candidato sobre a implementação de soluções com base na demanda estipulada.

Tenha atenção aos detalhes solicitados, garantindo que todos os requisitos solicitados sejam atendidos de forma adequada.

1. **Requisitos**

Criar uma API REST em que tem o objetivo de gerenciar e cadastrar um produto.

1. Características do produto:
   * Id
   * Descrição
   * Categoria
   * Dimensões
   * Código
   * Referência
   * Saldo de Estoque
   * Preço
   * Ativo
2. Deverá ter categoria:
   * Id
   * Descrição
   * Ativo
3. Deverá possuir operações que contemplem os seguintes métodos de HTTP
   * GET
   * POST
   * PUT
   * DELETE
   * PATCH
4. O serviço responsável por listar os produtos e categorias deverá ter paginação implementada
5. Utilize o Swagger para documentar a sua API
6. Deverá salvar os dados em um banco de dados mySQL.
7. O projeto deverá implementar repository pattern
8. Utilizar técnicas para apresentar um código limpo, portanto mostre pelo menos 3 exemplos no seu código que você considera estar escrevendo código de forma limpa e organizada e justifique ao lado por qual motivo tomou esta decisão. Para isso você deve fazer um print da parte do código como o exemplo abaixo:

|  |  |
| --- | --- |
| **Imagem** | **Justificativa** |
|  | Criação de Interfaces para manipulação |
|  | Criação de Repositorios para manipulação dos dados |
|  | Annotations nos controllers |

1. O código deve ter pelo menos 80% de cobertura de testes unitários
2. Faça um breve resumo dos motivos que te levaram a criar o projeto com a arquitetura escolhida, por exemplo, implementações de interfaces, separações de camadas, separações de pastas, criação das tabelas do banco de dados etc…

**Laravel é um framework completo que fornece ferramentas para construção de uma API de forma fácil, rápida e confiável. Com a capacidade de migrations e gerenciamento da base de dados. As pastas gerenciadas com esse framework é intuitiva.**

1. **Bonus 1**

Caso queira entregar um plus, sugiro a implementação de Autenticação e Autorização (**JWT**) utilizando Identity nos endpoints criados, estabelecendo pelo menos uma role específica para cada método HTTP.

1. **Bonus 2**

Implementar a Autenticação e Autorização (**JWT**) a partir do próprio Swagger.

1. **Onde disponibilizar o código fonte?**

Você deverá criar um repositório em um controle de versão de sua preferência, seguindo alguns requisitos:

* Deverá ser privado
* Deverá dar acesso para kleitondev@gmail.com

1. **Dicas**

* Apesar de ser um projeto muito simples, pense em uma arquitetura de um projeto maior, desta forma podemos avaliar a sua capacidade de criar projetos escaláveis e com código reutilizável
* Lembre que em projetos um pouco maiores teremos a necessidade de mais ambientes, como Development, Production etc…
* Caso ache necessário você pode utilizar pacotes nuget de sua preferência para implementar suas funcionalidades
* **Você terá 48 horas para entregar este projeto**