# Avaliação:

# Considerações gerais:

O intuito da prova é avaliar de modo geral a capacidade de resolução de problemas e a qualidade do código (clean code).

Não é necessário que o projeto seja entregue por inteiro. O mesmo será avaliado de acordo com os critérios mencionados.

O projeto tem tempo limite de acordo com a vaga a ser pretendida e será informado no recebimento da prova.

Obrigatoriamente a resolução deverá ser entregue para a avaliação por um link de repositório público (como o GitHub, por exemplo). Para a avaliação, será analisado o commit mais recente.

Caso haja dúvidas sobre o conteúdo da prova ou até mesmo sobre como implementar alguma de suas atividades, entrar em contato com nossos analistas.

# Objetivos:

Parte 1 - Montar uma API usando C# preferencialmente ASP.NET Core versão 3.1 (ou superior), com os seguintes métodos:

#### Insert

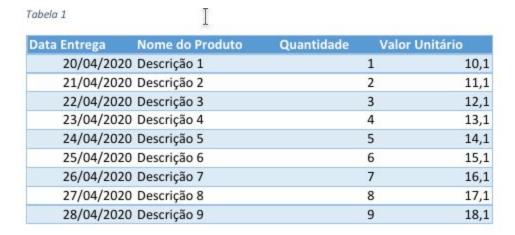
- 1. Deverá receber como entrada um arquivo Excel.
- 2. Durante o método insert, as seguintes regras devem ser contempladas:
  - Todos os campos são obrigatórios;
  - O campo data de entrega n\u00e3o pode ser menor ou igual que o dia atual;
  - O campo descrição precisa ter o tamanho máximo de 50 caracteres:
  - O campo quantidade tem que ser maior do que zero;
  - O campo valor unitário deve ser maior que zero e suas casas decimais devem ser arredondadas matematicamente para duas casas decimais.
- 3. Caso o lote seja válido: Os dados devem ser salvos em um banco de dados relacional (de sua escolha), que respeite o tipo de dados e suas validações, e deverá ser adicionado um identificador único para a importação (ID). O status de retorno deverá ser o 200 e os dados de retorno ficam a critério do desenvolvedor para facilitar a construção das demais partes.
- 4. Em caso de erros de validação: a API deverá retornar o status 400 (bad request) com uma lista de erros, contendo o número da linha do arquivo de Excel e o erro, ou erros, de validação.

## **GetAllImports**

1. Deverá listar todas as importações mostrando o seu identificador criado no método de insert, a data da importação, o número de itens, a menor data de entrega e o valor total da importação (Soma dos valores totais dos itens da mesma).

### GetImportByld

- 1. Deverá ter como parâmetro de entrada o identificador da importação;
- Caso existir o parâmetro solicitado, deverá trazer o dado como está na Tabela 1, com o acréscimo em cada linha da coluna "valor total" (produto de valor unitário multiplicado pela quantidade);
- 3. Caso não existir, retornar o status 404 (NotFound).



Parte 2 – Montar um cliente de Front-end para utilização das APIs com as seguintes rotas e ações:

### Rota: /

- 1. Rota inicial, deverá ocorrer a chamada da API de GetAllImports;
- 2. Deverá conter o botão de importar arquivo;
- 3. Caso sucesso ao importar, deverá redirecionar para a página de visualização;
- 4. Caso ocorra erros ao importar, mostrar de maneira amigável os erros para que o usuário possa corrigir o conteúdo do arquivo.

#### Rota: /import/{id}

- 1. Rota para mostrar os dados da importação de acordo com o método da API: GetByld;
- Deverá ter um botão de voltar para a rota inicial;
- 3. Deverá ter uma tabela com os dados de retorno da API.

# O que será avaliado:

- Conhecimento de banco de dados, olhando a modelagem do banco e como são realizadas as operações;
- Conhecimento de APIs;
- Organização e qualidade do código apresentado;
- Manipulação de arquivo Excel;
- Criação de regras a partir de especificações;
- Conhecimento de HTML;
- Conhecimento de consumo de dados de uma API;
- Performance da solução implementada;
- Atomicidade das operações;
- Teste unitário (Opcional).