

O escopo descrito neste teste serve de base para que você demonstre seus conhecimentos. Os requisitos deste sistema são muito simples, a evolução e práticas de arquiteturas é com você!

Iremos citar alguns pontos chaves que gostaríamos de ver em prática, mas fique a vontade para utilizar algo novo, estamos sempre em aberto para melhorar nosso pool de tecnologias.

#### **Requisitos Básicos:**

- Crie uma solução que contenha um projeto de apresentação de dados e uma Web Api construída com c#, que realize o acesso ao banco de dados e sirva de apoio para a aplicação de apresentação;
- Esta solução precisa gerenciar usuários, produtos e ter a possibilidade de criar pedidos de venda;
- Ao realizar um pedido de venda, um e-mail deve ser enviado ao comprador usando algum sistema de mensageria;
- Os usuários precisam estar autenticados para efetuar as ações do sistema.

#### **Requisitos de Banco de Dados:**

- Criar um banco de dados para armazenar os usuários, produtos e pedidos de venda (pode ser usado um local DB);
- O acesso ao banco de dados deve ser feito sem o uso de Wizards (ex: entity framework) e Store Procedures. Use queries diretamente pelo código;
- Utilizar transações nas inserções dos objetos.

#### **Requisitos da camada de visualização:**

- Utilizar html com jQuery, React ou Angular;
- As telas devem conter ações de cadastro, edição, alteração e exclusão/inativação;
- Os campos devem ser validados no cliente e no servidor.

#### **Requisitos de BackEnd:**

- Utilizar .NET Core 6;
- Dividir o código em camadas, preferencialmente utilizando DDD;
- Utilizar injeção de dependência para utilização dos serviços criados;
- Documentação da Web Api (swagger por exemplo).

#### **Diferenciais desejáveis (não obrigatórios):**

- Rabbit ou Kafka como plataforma de mensageria para simular a ação do envio de e-mail de um pedido efetuado com sucesso (não precisa disparar o email, precisa apenas ter a estrutura assíncrona via fila que possibilitará este envio no futuro);
- Aplicar algum sistema de log (log4net, serilog, etc);
- Aplicar testes unitários no projeto;
- Utilizar docker para executar a solução.

#### **O que será avaliado:**

- Lógica;
- Organização do código/projeto.

#### **Material para entrega:**

- Script/comandos para rodar/inicializar o projeto/banco de dados;
- Link do github.

Boa sorte!