

# Instruções

- O teste abaixo terá até **7 dias corridos** para ser entregue à contar da **data de recebimento deste manual**
- O código deverá ser **versionado** em um repositório **público do Github** e um link deverá ser enviado para avaliação assim que concluído
- Leia as instruções com cuidado e certifique-se que todos requisitos estão sendo endereçados
- O repositório deve fornecer instruções de como configurar, executar e testar o projeto
- Documentação e organização no geral também serão levados em consideração

## Caso de Uso

O Sistema da empresa 123Vendas está dividido em domínios. Existem domínios específicos para gestão de Estoque, CRM (cliente) e Vendas. Você é um desenvolvedor do time de Vendas. Agora nós precisamos implementar um protótipo de API de vendas. Como trabalhamos com DDD, para referenciar entidades de outros domínios, nós fazemos uso do padrão de **External Identities** com a desnormalização do descritivo das entidades.

Assim, você vai escrever uma API (CRUD completo) que manipule os registros de vendas. A API precisa ser capaz de informar:

Número da venda; data em que a venda foi efetuada; cliente; valor total da venda; filial em que a venda foi efetuada; produtos; quantidades; valores unitário; descontos; valor total de cada item; Cancelado/Não Cancelado;

Não será obrigatório, mas seria um diferencial a construção de um código para publicação de eventos de CompraCriada, CompraAlterada, CompraCancelada; ItemCancelado. Se fizer o código, é dispensável publicar em algum Message Broker (Rabbit ou Service bus, por exemplo) de fato. Pode logar uma mensagem no log da aplicação ou como você achar mais conveniente.

## **Você precisa utilizar na sua API:**

- Serilog para logs;
- Divisão em camadas (API, Domain, Data);
- Aplicar Git Flow workflow;
- Aplicar Commit semântico;
- APIs REST;
- Clean Code;
- SOLID;
- DRY;
- YAGNI;
- Object Calisthenics;
- Testes de unidade
  - Xunit;
  - FluentAssertions;
  - Bogus;
  - NSubstitute;
  - Boas práticas de escrita de código;
- Teste de integração (desejável)
  - Test Container

O objetivo final é analisarmos sua capacidade criativa e de solução de problemas, a aplicação de boas práticas de codificação, clareza na escrita e gestão do histórico de alterações (GIT).