## Teste para desenvolvedor .net

O teste abaixo visa avaliar o perfil técnico dos candidatos a desenvolvedor backend na Superdigital. Leia atentamente as instruções abaixo e responDa as questões da melhor forma possível.

* O teste será composto de três perguntas técnicas e uma avaliação prática.
* Responda as perguntas com suas palavras, com seu entendimento sobre o tema.
* Publique o teste prático com as respostas em um repositório no git-hub e envie o endereço, (valdir.silva@superdigital.com.br)

## Perguntas:

## Explique com suas palavras o que é domain driven design e sua importância na estratégia de desenvolvimento de software.

## R.: É um príncipio de modelagem de software que tem com objetivo focar o desenvolvimento no negócio da empresa, chamado de domínio, usando uma linguagem ubíqua, ou seja, uma linguagem comum a desenvolvedores e key users.

## Explique com suas palavras o que é e como funciona uma arquitetura baseada em microservices. explique ganhos com este modelo e desafios em sua implementação.

## R.: São serviços independentes com o objetivo de gerenciar uma única funcionalidade no sistema, por exemplo: Cadastro cliente, cadastro produto, consulta pedidos, cada um desses é um serviço único e responsável por seu contexto e domínio, os microserviço se integram entre si, mas não conhecem o contexto e o domínio do outro.

## **Prós:** independencia, se um microserviço estiver fora o resto do sistema continua funcionando, equipes indididuais em cada microserviço, entender e entregar pequenas atividades permite mais qualidade e acertividade.

## **Contras:** o nível de complexidade geral da solução como um todo aumenta, testar dependências de outros serviços, lantência entre serviços, relacionamentos dos dados.

## explique qual a diferença entre comunicação sincrona e assincrona e qual o melhor cenário para utilizar uma ou outra.

## **sincrona:** após fazer uma requisição o cliente aguarda o retorno do servidor.

## **assincrona:** o cliente faz a requisição e fica disponível para executar outras tarefas enquanto é executado no servidor, após execução é feito o callback para quem fez a requisição.

## **cenário sincrono:** Usado quando é necessário dar continuidade em uma atividade: ex.: Consulta de cliente para alteração, cadastro de cliente em que será criado um pedido em seguida.

## **cenário Assincrono:** pode ser usado quando não é necessário retorno para continuar com as atividades, exemplo: envio de e-mail, integrações em back-ground, geração de relatórios.

## Teste prático:

## regra de negócio:

## 

## criar um microservices que, através de um http post efetue uma operação de debito (origem) e credito (destino) nas contas correntes.

## entidades: contacorrente, lancamentos (voce pode incrementar com outras entidades se achar necessário)

## Parâmetros de entrada:

## conta origem

## conta destino

## valor

## Parâmetros de saída:

## http status code

## 

## informações adicionais:

## o método “post” devera receber os parametros no body da requisição em formato json

## UTILIZE Domain Driven Design

## serão avaliados critérios de arquitetura como separação de responsabilidade, clean code, segurança e testes

## tecnologias que você pode utilizar .net core 2.X, c#, xunits (testes)

## no término do projeto, publique o código em um repositório no git-hub