## 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 compartilhe o link.

## Perguntas:

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

## É um conjunto de técnicas que auxiliam no desenvolvimento de software, pensando em reaproveitamento de código, separação de responsabilidade entre outras técnicas abordadas. Uma técnica interessante é a forma de olhar o processo real de algo físico e colocar dentro do software. No meu ponto de vista é muito importante essa técnica não só pelo reaproveitamento de código mas sim pela facilidade em sua manutenção, já que dessa forma o código seguindo o ddd fica mais fácil de uma possível manutenção.

## 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.

## É uma arquitetura onde os serviços ficam separados por responsabilidade, hoje em dia é muito comum sistemas nesse modelo, já que com a utilização de api você pode trabalhar com sistemas interligados facilitando a integração entre os sistemas. Desde sistemas web até mesmo sistemas mobile fazem a utilização dessa arquitetura. Por se tratar de um serviço de comunicação requer muita atenção na parte de segurança, é um grande desafio a implantação do mesmo buscando sempre tecnologias de ponta para não deixar possíveis falhas. Dessa forma também é sempre importante trabalhar de forma assíncrona, já que não se sabe quanto tempo vai demorar a resposta do sistema que está sendo consumido.

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

## A comunicação assíncrona é muito utilizado quando não se tem controle sobre a resposta do serviço utilizado ou seja, você deve utilizar para não ter problemas com lentidão na resposta de comunicação entre dois serviços. Porém a forma assíncrona é um pouco mais difícil de escrever o código porém em sistemas com micro serviços é muito recomendado.

## 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 e compartilhe o link.