## 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.

## Perguntas:

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

## É uma arquitetura que foca no negócio e ver o negócio como ponto focal “domínio” e tem uma filosofia de ter uma linguagem ambígua, ou seja, desde o usuário final até o programador ter a mesma visão ou até mesmo o nome de um método deve ser compreendido para este usuário. Outro ponto importante que vejo é a separação dos contextos, delimita, organizar, separar as responsabilidades e defini o comportamento dos processos.

## 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.

## Ganho na produtividade, número de deploy maior e menos riscos por esta em pequenos processos, melhor e mais fácil para o time de QA testar e trabalhar e implementação de entrega continua.

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

## Síncrona é quando o emissor precisa da resposta do receptor, um exemplo é quando precisamos fazer um login através de um sistema, o usuário sempre precisa está autenticado para poder continuar o processo.

## Assíncrono não precisa o receptor dar a mensagem no mesmo momento, um exemplo é SMS, WhatsAPP etc..

## 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