## 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.

## DDD – Nao é uma arquitetura, nao e um design pattern. E uma metodologia de desenvolvimento onde o foco principal se concentra no dominio do negocio onde sera diretamente atuado na resolucao do problema.

## 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.

## E uma arquitetura baseada em pequenos servicos responsaveis por fragmentos de um sistema complexo. esses pequenos servicos, conhecidos como micro-services, sao construidos pensando em sua alta escalabilidade. a principal vantagem e tornar o sistema sempre operante. ou seja, caso algum micro-service fique indisponivel, apenas uma pequena parte do sistema sera afetado.

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

## Podems dizer que comunicacao sincrona, equipara-se ao termo “one way!” ou seja, existe apenas um caminho a ser seguido na comunicacao e o mesmo nao retorna uma resposta. seria tambem uma especie de ping, sem pong!

## ja a comunicacao assincrona, enviamos o ping a espera de um pong. claro, com um tempo de espera onde caso seja atingido, sera emitido pelo protocolo de comunicacao um “time out!”

## Teste prático:

## regra de negócio:

## 

## Precisamos de um serviço que , dado duas palavras (alfanuméricas) , calcule o total de movimentações para se transformar a primeira palavra na segunda palavra.

## exemplos:

## dado as palavras “gato”, “rato” a saída deve ser “1” pois:

## substituindo o “G” do gato pelo “R” , transformamos “gato” em “rato”

## exemplo 2:

## dado as palavras “cavalo”, “pato” a saída deve ser “4” pois:

## removendo o “c”, removendo o “a”, substituindo “v” por “p” e substituindo “l” por “t” transformamos “cavalo” em “pato”

## informações adicionais:

## as strings podem ter qualquer tamanho.

## qualquer palavra pode ser informada.

## faça uma api rest com um método “post” recebendo as palavras no body da requisição

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

## documente a api com swagger

## tecnologias que você pode utilizar .net core 2.1, c#, xunits (testes)

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