## Teste para desenvolvedor .net

O teste abaixo visa avaliar o perfil técnico dos candidatos a desenvolvedor backend. 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 nos envie.

## Perguntas:

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

## DDD é uma metodologia de desenvolvimento onde as regras de negocio são definidas em dominios. Sendo que cada dominio deve representar as necessidades de negócio em comportamentos. No dominio temos classes que tem suas propriedades e métodos que são responsaveis por tomar as suas proprias ações, e devem ter suas reponsabilidades bem definidas e separadas dentro dos dominios. O DDD visa uma abordagem de desenvolvimento onde as classes ficam mais proximas as regras de negocio do mundo real.

## **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.**

## Microserviços segue uma idéia de implementação onde, o sistema pode ser dividido em vários modulos, e cada modulo pode ser criado e implementado de forma que funcione independente dos outros modulos. Uma das grandes vantagens dessa forma de desenvolvimento é que é possível efetuar a manutenção de cada modulo de forma independente e tambem é possivel adicionar novos modulos sem impactar no modulos existentes. Tambem é possivel pensar no escalonamento modular de um sistema de acordo com as necessidades de negocio.

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

## Comunicação síncrona é quando é disparada uma ação e quem disparou esta ação fica aguardando o retorno da mesma. Comunicação assíncrona é quando é disparada uma ação e quem disparou esta ação fica livre para executar outras ações sem ficar preso esperando a resposta desta ação.

## 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