## 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.

## O DDD é um PADRÃO DE ARQUITETURA EM CAMADA ORIENTADA A DOMÍNIO, QUE TEM COMO IMPORTÂNCIA A DIVISÃO DE RESPONSABILIDADE PARA CADA CAMADA, O ISOLAMENTO DO DOMÍNIO E ABSTRAÇÃO.

## 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.

## uMA ARQUITETURA BASEADA EM MICROSERVIÇO É COMPOSTAS DE PEQUENOS COMPONENTES DESENVOLVIDOS INDIVIDUALMENTE PARA A COMPOSIÇÃO DE UMA APLICAÇÃO, COM ISSO OS GANHOS CONSISTEM EM MENOS IMPACTOS QUANDO OCORRE PROBLEMA EM ALGUM DOS SERVIÇOS, NÃO É IMPACTADA TODA A APLICAÇÃO E FACILIATA A CORREÇÃO, ALÉM DA FACILIDADE DE GANHOS EM ESCALABILIDADE. ALGUNS DOS GRADES DESAFIOS PARA A IMPLEMENTAÇÃO DE MICROSERVIÇOS, SÃO OS HABITOS ADQUIRIDOS COMO OS PADRÕES DE ARQUITETURA MONOLÍTICA E QUE DIFICULTA O ENTENDIMENTO NA HORA DE ESCREVER OS RESQUISITOS.

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

## cOMUNICAÇÃO SÍNCLONA, É A COMINICAÇÃO ONDE AS DUAS PONTAS ESTÃO SINCLONIZADAS AO MESMO TEMPO E ASÍNCLONA, NÃO ESTÃO SINCLONIZADAS. OS CENÁRIOS QUE PODE SER UTILIZADO O SÍNCLONO, É A COMUNICAÇÃO POR TELEFONE, ONDE UM PERGUNTA E O OUTRO RESPONDE, JÁ O ASSÍNCLONO, É COMPARADO OS SISTEMA DE CAIXA POSTAL TELEFONE OU WHATSAPP, VOCÊ EMITE UMA MENSAGEM E O REPTOR RESPONDE QUANDO ELE PUDER.

## 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