## 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, (valdir.silva@superdigital.com.br)

## Perguntas:

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

## é um padrão de modelagem de software orientado a objetos que reforça conceitos e boas práticas DE DESENVOLVIMENTO.

## É IMPORTANTE POR PERMITIR AGILIDADE NO PROCESSO DE DESENVOLVIMENTO, tais COMO:

## - FAVORECER REUTILIZAÇÃO DE CÓDIGO

## - ACOPLAMENTO

## - iNDEPÊNCIA DA TECNOLOGIA

## - melhora o alinhamento do código com o negócio

## 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.

## ARQUITETURA MICROSERVICES É um modo de projetar softwares EM conjuntos de serviços de implementação independentes.

## uMAS DAS SUAS PRINCIPAIS VANTAGENS É A CENTRALIAZÇÃO EM UM ÚNICO BLOCO.

## Aplicações monolíticas sofrem do problema ponto único de falha (SPOF), caso parte de um sistema falhar, impedirá que todo o sistema funcione. Além disso, aumenta os custos de manutenção, pois a complexidade do código será maior e mais arriscada. Todo o sistema é acoplado, há menor escalabilidade, é necessário fazer deploy da aplicação inteira para alterar somente um componente.

## Com os microservices esses problemas são minimizados, uma vez que a aplicação é dividida em módulos específicos a sua função.

## onde é possível minimizar os “SPOF” devido a aplicação não ser completamente acoplada.

## Caso um módulo apresente falha, o sistema não fica totalmente comprometido.

## O código de cada módulo é bem menor, facilitando a manutenção.

## Como desenvolvimento de testes exigem uma maior complexidade, Um dos desafios seria na implementação no mecanismo de comunicação entre a serviços, por abranger vários serviçoes requer um gerenciamento mais cuidadoso das equipes.

## .

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

## NA COMUNICAÇÃO SINCRONA cada bloco de informação é transmitido e recebido num instante de tempo bem definido e conhecido pelo transmissor e receptor, ou seja, estes têm que estar sincronizados.

## jÁ NA ASSICRONA cada bloco de dados inclui um bloco de informação de controle (flag), para que se saiba exatamente onde começa e acaba o bloco de dados e qual a sua posição na sequência de informação transmitida.

## Transferências assíncronas são geralmente mais rápidas do que transferências síncronas, por não coordenar a transmissão, no entanto tendem a ocorrer mais erros caso for necessario obter a transmissão ordenada.

## 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