## 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.

## dOMAIN DRIVEN DESIGN É O DESENVOLVIMENTO ORIENTADO AO DOMINIO, TODA MODELAGEM DE CLASSES REGRAS DE NEGÓCIO VÃO ESTAR CONCENTRADAS NO DOMINIO DA APLICAÇÃO.

## E A DIRETRIZ MAIS IMPORTANTE DO DDD É A LINGUAGEM OBLIQUOA ONDE TODOS OS (USUÁRIOS, DESENVOLVEDORES) DA ORGANIZAÇÃO “FALAM A MESMA LINGUA” O QUE FACILITA A COMUNICAÇÃO MESMO PARA UM NOVO DESENVOLVEDOR.

## Por exemplo se um funcionario fala que está com erro no cadastro de conta corrente o desenvolvedor novo já sabe onde deve debugar para sanar o erro.

## O ddd também trouxe o conceito de separar as camadas de acesso a banco de DOS IOC, crosscutting usando um pouco de programação orientada a aspectos. no ddd se aplica muito a orientação a objeto, solid e design patterns, mudou MUITO A forma de desenvolver e para melhor.

## 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.

## MICROSERVICES, É UMA ARQUITETURA QUE TEM COMO OBJETIVO SEPARAR MÓDULOS DO SISTEMA EM SERVIÇOS. POR EXEMPLO UM ECOMMERCE ANTIGAMENTE ERA FEITO UM SERVIÇO PARA ATENDER TODAS AS REQUISIÇÕES SEJA

## - CADASTRO DE CLIENTE.

## - pRODUTO.

## - CARRINHO(FECHAMENTO.

## nO MICROSERVICES TERIAMOS UM SERVIÇO PARA CLIENTES, UM PARA PRODUTO E OUTRO PARA O FECHAMENTO DE PEDIDOS, O GANHO NESSE MODELO É ESCALABIDADE, SEPARAÇÃO DOS PROJETOS FICANDO MAIS FÁCIL SUA MANUTENÇÃO. PORÉM É UM DESAFIO MANTER TUDO SE CONVERSANDO UMA UI PARA CADA SERVIÇO E GARANTIR SEU O ESTADO DA APLICAÇÃO.

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

## a COMUNICAÇÃO SINCRONA É BLOQUEANTE OU SEJA TEM QUE ESPERAR EXECUTAR ELA PARA DEPOIS EXECUTAR OUTRAS TAREFA.

## jÁ A ASSINCRONA PODE-SE EXECUTAR VARIAS TAREFAS ATÉ QUE ELA TERMINE.

## 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