

Respostas teste MC1

3. Como você se atualiza tecnicamente?

Procuro me atualizar verificando novidades das versões, fazendo tutorias, e assistindo vídeo aulas, sites de EAD e lendo livros técnicos.

4. Crie uma função para calcular o n-ésimo elemento da Sequência de Fibonacci (fibonacci).

Qual solução é mais performática, iterativa ou recursiva? Por que?

A solução iterativa. Porque a recursiva no dot.net vai para a pilha e a pilha tem um processamento mais lento, outra questão é o GC que se a função recursiva possuir variáveis locais fica ainda mais lento, porque tenta limpar a cada interação.

Opcional: Qual é o 5287º elemento da sequência?

-1426463864 Duration Iterative 00:00:00.5325764

-1426463864 Duration Recursive 00:00:00.6477712

5. O que significa SOLID?

SOLID — acrônimo para 5 práticas essenciais que devem ser seguidas para construir sistemas que possam crescer com uma boa manutenção.

Single Responsibility Principle

Open/Closed Principle

Liskov Substitution Principle

Interface Segregation Principle

Dependency Inversion Principle

6. O que são design patterns?

Na engenharia de software, um padrão de design é uma solução repetível geral para um problema comum no design de software. Um padrão de design não é um design final que pode ser transformado diretamente em código. É uma descrição ou modelo de como resolver um problema que pode ser usado em muitas situações diferentes.

i. Quais são os tipos de design patterns?

Os padrões de design são divididos em três grupos fundamentais:

- Comportamental(Behavioral,
- Criadora(Criational) e.
- Estrutural(Structural).

ii. Com quais você está familiarizado? Qual é a função deles?

DDD (Domain Drive Design) objetivo é facilitar a implementação de regras e processos complexos, onde visa a divisão de responsabilidades por camadas.

TDD (Test Drive Development) desenvolvimento dirigido por testes. Neste caso o teste só atende a regra de negócio e portanto não existe desenvolvimento desnecessário.

MVC(Model View Controller) divide a aplicação em 3 camadas.

iii.Opcional: Qual é sua opinião quanto ao uso de design patterns?

O uso de padrões facilita a criação de projetos e futuras manutenções.

7. Qual foi o último livro técnico que você leu? Quando foi isso?

Foi um e-book da Data Science Academy sobre Machine Learning. Faz um ano.

8. Observação: se já tivermos lido e você for chamado para uma entrevista, perguntas poderão ser feitas a respeito do mesmo.

9.Cite 3 maneiras diferentes de implementar Dependency Inversion.

Só conheço com criação de interface que pode ser herdada pela classe concreta

9. O que são ORMs?

É uma técnica utilizada para fazer o mapeamento entre sistemas orientados a objetos e bancos de dados relacionais, onde as tabelas do banco de dados são representadas em classes e os registros das tabelas seriam instâncias destas classes.

i. Quais você conhece bem?

Eu conheço bem Entity Framework. Já trabalhei em algumas empresas.

ii. Opcional: Cite pelo menos 2 vantagens e 2 desvantagens de seu uso.

Vantagens:

- permite que você use OOP como herança de dados sem dor de cabeça.
- escreve seu modelo de dados em um só lugar, e é mais fácil de atualizar, manter e reutilizar o código

Desvantagens:

- desempenho é OK para consultas habituais, mas um mestre SQL sempre fazer melhor com seu próprio SQL para grandes projetos.
- abstrai o DB. Enquanto é OK se você sabe o que está acontecendo por trás, é uma armadilha para novos programadores que podem gravar declarações muito gananciosos, como uma batida pesada em um loop for.

10. O que são microserviços?

os microserviços são um estilo arquitetônico em que as aplicações são decompostas em serviços acoplados.

i. Quais são suas vantagens e desvantagens?

Vantagem:

- oferecendo modularidade crescente, tornando as aplicações mais fáceis de desenvolver, testar, implantar e, o mais importante, mudar e manter.

Desvantagem:

- É mais difícil de criar uma camada para encapsular preocupações transversais, comuns a vários módulos numa arquitetura de microsserviços.

- É impossível ser produtivo se não houver uma boa automatização de deploy, pois os microsserviços são freqüentemente implantados em suas próprias máquinas virtuais ou contêineres, o que é muito trabalhoso para gerenciar manualmente.

11. Com a seguinte representação de produto (crud):

```
{
  "sku": 43264,
  "name": "Batata frita Ruffles Cebola & Salsa",
  "inventory": {
    "quantity": 15,
    "warehouses": [
      {
        "locality": "SP",
        "quantity": 12,
        "type": "ECOMMERCE"
      },
      {
        "locality": "MOEMA",
        "quantity": 3,
        "type": "PHYSICAL_STORE"
      }
    ]
  },
  "isMarketable": true
}
```

Crie endpoints para as seguintes ações:

- Criação de produto onde o payload será o json informado acima (exceto as propriedades isMarketable e inventory.quantity)
- Edição de produto por sku
- Recuperação de produto por sku
- Deleção de produto por sku

Requisitos:

Toda vez que um produto for recuperado por sku deverá ser calculado a propriedade: `inventory.quantity`

A propriedade `inventory.quantity` é a soma da `quantity` dos warehouses

Toda vez que um produto for recuperado por sku deverá ser calculado a propriedade: `isMarketable`

Um produto é marketable sempre que seu `inventory.quantity` for maior que 0

Caso um produto já existente em memória tente ser criado com o mesmo sku uma exceção deverá ser lançada

Dois produtos são considerados iguais se os seus skus forem iguais

Ao atualizar um produto, o antigo deve ser sobrescrito com o que esta sendo enviado na requisição

A requisição deve receber o sku e atualizar com o produto que tbm esta vindo na requisição