

1. Como você se atualiza tecnicamente?

Grupos de facebook, vídeos ao vivo, podcasts, cursos

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

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

Para números pequenos como 2 ou 5, não existe uma diferença significativa no tempo de execução, mas para números maiores como 10, 20 ou 30, o método recursivo vai ficando mais e mais lento, isso devido à necessidade de alocação de memória para cada chamado ao método.

numero	LoopFibonacci	RecursiveFibonacci
40	00:00:00.0001678	00:00:01.8598866

ii. *Optional*: Qual é o 5287º elemento da sequência?

É um número muito alto para ser calculado num computador normal.

3. O que significa SOLID?

- Single Responsibility Principle
- Open-Closed Principle
- Liskov Principle
- Interface segregation principle
- Dependency Inversion Principle

4. O que são design patterns?

i. Quais são os tipos de design patterns?

Criacional, comportamental e estrutural

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

Facade: Simplificar e/o segmentar o acesso a recursos complexos.

Repository: Simplificar e generalizar o acesso a dados.

Decorator: Permite adicionar dinamicamente, novas responsabilidades a um objeto.

Factory: Fornecer uma interface para instanciar objetos.

Strategy: Define um contrato, a classe que for implementar a interface, pode definir internamente, o comportamento do método implementado.

ServiceLocator: Resolve a implementação de serviços.

Unit of work: Considera um conjunto de processos como uma unidade, por exemplo: um dado será salvo no banco de dados só se todos os passos de um processo foram executados com sucesso, senão nenhum dado será persistido no banco.

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

É uma boa prática que fornece soluções rápidas para problemas que alguém já resolveu antes de você, também ajuda a desacoplar um desenvolvedor específico de uma solução de software, ou seja, não será necessário que o criador do software de manutenção nele, qualquer desenvolvedor familiarizado com os design patterns, poderia entender o código.

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

Micro-serviços prontos para produção- Susan J. Fowler

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

6. Cite 3 maneiras diferentes de implementar Dependency Inversion.

- Injeção de dependência
- Pattern Service Locator

7. O que são ORMs?

- i. Quais você conhece bem?

entityFramework ,dapper, nhibernet

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

Entity Framework

Vantagens:

- Padrão de aplicações .NET
- CodeFirst
- Fácil configuração
- Aprendizado rápido.
- Funciona nativamente com LINQ.

Desvantagens:

- Baixa performance na leitura de grande volume de dados.

Dapper

Vantagens:

- Alta performance na leitura de dados.
- Fácil de configurar.

Desvantagens:

- A tarefa de debbuging não é fácil.
- Precisa de conhecimento em SQL.

8. O que são microsserviços?

É uma abordagem arquitetural para construção de sistemas não monolíticos

i. Quais são suas vantagens e desvantagens?

Vantagens:

- Baixo acoplamento
- Permite várias equipes trabalharem de forma paralela.
- Permite construir sistemas robustos e resilientes.
- Permite criar sistemas de alta disponibilidade e performance.

Desvantagens:

- É necessário dispor de vários recursos para manter ativos os micro-serviços, estes recursos podem ser de um alto custo para as empresas.
- A manutenção requer de pessoal especializado.