

1. Como você se atualiza tecnicamente?

Assino canais de desenvolvimento no youtube. Vejo as conferências oficiais das empresas (como a BUILD da Microsoft ou NG-Conf do angular). Sigo os criadores e perfis oficiais dos frameworks no twitter como <https://twitter.com/angular>, vou a eventos organizados pela comunidade como o angularSP e também faço cursos na udemy para aprender as novas tecnologias.

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

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

Acredito que a solução mais performática seria a iterativa pois a recursiva gasta mais memória (já que tem que armazenar as N chamadas recursivas na pilha até o fim da execução) e na iterativa o valor da variável é substituída.

- b. *Opcional*: Qual é o 5287º elemento da sequência?

14.516.649.649.369.212.109

3. O que significa SOLID?

SOLID é uma sigla dos 5 princípios da programação orientada a objeto.

Single Responsibility: Uma classe deve ter somente uma responsabilidade.

Open-Closed: A classe tem que ser aberta para extensão mas fechada pra modificação

Liskov Substitution: As classes devem ser substituídas por suas classes bases

Interface Segregation: Criar interfaces específicas ao invés de uma abrangente. Uma classe não deve ser forçada a implementar interfaces e métodos que não irão utilizar

Dependency Inversion: Dependendo de uma abstração e não de uma implementação

4. O que são design patterns?

É um modelo de arquitetura de software para resolver uma determinada situação (problema) no desenvolvimento de software.

- a. Quais são os tipos de design patterns?

Criação, estrutural e comportamental

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

Singleton: Garante que a classe só deve ter uma instância.

Factory: Define uma interface para criar um objeto, mas deixa as subclasses decidirem qual classe específica instanciar.

Facade: Fornece uma interface simplificada para um framework ou qualquer outro conjunto complexo de classes.

- c. *Opcional*: Qual é sua opinião quanto ao uso de design patterns?

São ótimos para resolver problemas específicos. O desafio é escolher o pattern correto para o problema e não tentar encaixar algum pattern só porque ele é bonito.

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

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

ng-Book (<https://www.newline.co/ng-book/2/>)

6. Cite 3 maneiras diferentes de implementar Dependency Inversion.

1. Injeção de dependência

2. Design pattern Abstract Factory

3. Sempre ter interface e fazer a comunicação entre as camadas/classes via interface

7. O que são ORMs?

São frameworks de abstração ao banco de dados.

- Quais você conhece bem?

Entity Framework, dapper

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

Vantagens: agilidade no desenvolvimento, código mais elegante

Desvantagens: pode gerar queries mais lentas (menos performáticas) que as feitas manualmente, pode trazer mais dados que o necessário.

8. O que são microsserviços?

São aplicações grandes divididas em blocos menores que, ao serem recompostos, oferecem toda a funcionalidade de uma aplicação altamente complexa e de grande escala.

Quais são suas vantagens e desvantagens?

Vantagem: escalabilidade, poder desenvolver cada bloco em uma stack diferente, arquitetura do serviço geralmente é mais simples e menor, sistemas independente (se o módulo A cair, o B não cai).

Desvantagens: Arquitetura geral pode ser torna complexa (garantir filas e transações), maior complexidade operacional (publicar, monitorar)