

1) Crie um fork desse repositório, faça os testes, responda as perguntas e depois submeta um pull request.

2) Os testes envolvendo código devem ser feitos em C# ou Java/Kotlin (Android). Cada codificação deve estar em uma pasta com o nome que está entre parênteses nas questões. As questões teóricas devem ser respondidas em um pdf que também será adicionado ao GIT. O primeiro teste será pegar os códigos e dar um run. Tem que rodar de primeira.

3) Como você se atualiza tecnicamente? **Hoje utilizo a udeemy como uma das minhas principais base de atualização de conhecimento, mas também os sites dos fabricantes como Microsoft por exemplo.**

4) 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? **Eu acredito que seja a Iterativa. A recursiva embora resolva o problema em porções menores é necessário realizar várias checagens a cada chamada, enquanto que a iterativa em uma única chamada resolve o problema.**

ii. Opcional: Qual é o 5287º elemento da sequência?

5) O que significa SOLID? **É um acrônimo dos cinco primeiros princípios da programação orientada a objetos (criado por Robert C. Martin)**

S – Princípio de Responsabilidade Única

O – Princípio Open Close

L – Princípio de Substituição Liskov

I – Princípio de Segregação de Interface

D – Princípio de Inversão de Dependência

6) O que são design patterns? **São formas de organização um projeto, estruturando-os com padrões já estudados, testados e aceitos.**

i. Quais são os tipos de design patterns? Creational, Structural, Behavioral

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

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

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

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

8) Cite 3 maneiras diferentes de implementar Dependency Inversion.

Em DOT.NET, poderia ser feito via containers como Ninject ou Unity, via Construtor ou na class Startup.cs.

9) O que são ORMs? **Object Relational Mapping (Ferramentas para mapeamento de banco de dados relacionais)**

i. Quais você conhece bem? **EntityFramework e Dapper**

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

10) O que são microsserviços? **É uma abordagem arquitetural que facilita o desenvolvimento de sistemas distribuídos e escaláveis.**

i. Quais são suas vantagens e desvantagens? **As vantagens mais comuns são a facilidade para escalar o sistema. Desvantagens, além de ter muitos códigos repetidos, a manutenção em sistemas distribuídos não é uma coisa simples.**