

1. Crie um fork desse repositório, faça os testes, responda as perguntas e depois submeta um pull request.  
**R:** OK
2. Os testes envolvendo código devem ser feitos preferencialmente 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.  
**R:** OK
3. Como você se atualiza tecnicamente?  
**R:** Ao longo da carreira a atualização técnica foi ocorrendo junto com o lançamento de novas versões, principalmente colocando em prática novas tecnologias e buscando informações complementares na internet.
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?  
**R:** Acredito que seja a iterativa, só usaria algo recursivo neste cenário caso precisasse tirar uma prova real, ou algo do tipo.
  - ii. *Optional:* Qual é o 5287º elemento da sequência?  
**R:** 14516649649369212109
5. O que significa SOLID?  
**R:** Sigla de 5 atitudes já estudados que facilitam e resolvem problemas no desenvolvimento de software.
6. O que são design patterns?  
**R:** Padrões de projeto genéricos que podem resolver problemas comuns.
  - i. Quais são os tipos de design patterns?  
**R:** De criação, de estrutura e de comportamento
  - ii. Com quais você está familiarizado? Qual é a função deles?  
**R:** Sem dúvida a herança de objetos/classes sempre esteve presente nos projetos que participei, servindo para subir o nível das propriedades, reaproveitar código, facilitar manutenção...
  - iii. *Optional:* Qual é sua opinião quanto ao uso de design patterns?  
**R:** Ainda espero participar de um projeto que o uso de design mais complexos seja realmente necessário, acredito que a maioria só sei na teoria. Na prática soluções mais diretas e focadas em cada problema foram mais utilizadas.

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.*  
**R:** Não lembro de ter lido um livro técnico por completo. No início eram os MOCs e os treinamentos oficiais da Microsoft, mas sempre indo direto ao ponto em questão. Estou aberto a sugestões.
8. Cite 3 maneiras diferentes de implementar Dependency Inversion.
- R:** Conheço 1, e sempre em dupla: inversão de controle + injeção de dependência. Já trabalhei com esse padrão porque iríamos configurar testes no projeto
9. O que são ORMs?
- R:** Ferramentas para mapear bancos de dados relacionais
- i. Quais você conhece bem?  
**R:** Somente o Entity Framework trabalhei pra valer. nhibernate e dapper só por curiosidade
  - ii. *Opcional:* Cite pelo menos 2 vantagens e 2 desvantagens de seu uso.  
**R:** Vantagem: Codificação rápida e limpa  
**R:** Desvantagem: Dificuldade em execuções mais complexas e lentidão no processamento
10. O que são microsserviços?
- R:** Técnica para desacoplar as funcionalidades de uma arquitetura
- i. Quais são suas vantagens e desvantagens?  
**R:** A vantagem é que em um cenário onde a publicação individual de funcionalidades seja necessária, é mais fácil e confiável de se fazer, o que também pode ser uma desvantagem em cenários mais compactados.
11. Com a seguinte representação de produto (crud):
- R:** OK