**4.Crie uma função para calcular o \_n-ésimo\_ elemento da Sequência de Fibonacci (fibonacci).**

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

A Solução mais performática é a Iterativa pois a cada ciclo acontece apenas uma soma, enquanto na recursiva cada chamada à função acontece duas novas chamadas.

**4.2. \_Opcional:\_ Qual é o 5287º elemento da sequência?**

O 5287º da sequência é o número:

3699721211173786484437838765123148419466707898251351522448137480285857913565834984520929216272716418016617994714385706765899530119354140728762185202815777338147008903977749657309606292580372503125339015900648037446131017454455862827667286892454440241565074532997862818830158805546197789485773192422757030757222261581444340972226701085729760308472331417673312770100555817510353085244841566702932433838879894021591016349334069905158081254197855346709280263141652935496803290379831445043302749345673846641984494178362767517508794916843636324641389000529946745799124325256979617113189408580120982396374099693648553960185095218493316678336284241270766611700540081238466107264659575911480924834151162756989179951884273526183862471003751633145921455212111399457568503514564417826619590918889335234088266697280105428040803227654120652991398143150951746110043739443754100271772151849655229460523456438017797003287877132113141122300484831184715566762621656769729907758837617330777906284940461734974408911647738932397589929774952184814644006493763068496727152688598522153343537297991570632774799688636613263250121933

**5. O que significa SOLID?**

SOLID é um acrônimo dos cinco primeiros princípios da Programação Orientada a Objetos:

SRP - Single responsibility principle

Princípio da Responsabilidade Única - Uma classe deve ter um, e somente um, motivo para mudar.

OCP - Open/closed principle

Princípio do Aberto/Fechado - Você deve ser capaz de estender um comportamento de uma classe sem a necessidade de modificá-lo.

LSP - Liskov substitution principle

Princípio da substituição de Liskov - As classes derivadas devem ser substituíveis por suas classes bases.

ISP - Interface segregation principle

Princípio da segregação de interfaces - Muitas interfaces específicas são melhores do que uma interface única geral.

DIP - Dependency inversion principle

Princípio da inversão de dependência - Dependa de abstrações e não de implementações.

Se for seguindo garante benefícios como um sistema fácil de entender e realizar manutenções/testes.

**6. O que são design patterns?**

Design Patterns são soluções utilizadas para tipos de problemas recorrentes em sistemas.

**6.1. Quais são os tipos de design patterns?**

Os tipos de Design Patterns são: Criação, Estruturais e Comportamentais.

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

Estou familiarizado com:

Singleton (apenas uma instância de uma classe deve existir) para algumas aplicações. Não utilizo para todas as classes que possuo no projeto;

Model View Controler (MVC);

Injeção de Dependência;

**6.3. \_Opcional:\_ Qual é sua opinião quanto ao uso de design patterns?**

Eu gosto de aplicar os modelos Design Patterns que conheço em meu sistema pois através deles consigo um sistema bem organizado e qualquer manutenção ou produção de novas funcionalidades se dá com maior facilidade.

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

Visual C# 2005, Passo a Passo da Microsoft por John Sharp

Eu li completo em 2008 e em 2017.

**8. Cite 3 maneiras diferentes de implementar Dependency Inversion.**

Através do Construtor, Propriedades ou Interface.

**9. O que são ORMs?**

ORM é Object Relational Mapper e em português Mapeador Objeto Relacional que permite fazer um mapeamento entre os objetos de um sistema e os dados que estes representam.

* 1. **Quais você conhece bem?**

Eu conheço bem Entity Framework.

* 1. **\_Opcional:\_ Cite pelo menos 2 vantagens e 2 desvantagens de seu uso.**

**Vantagens:**

Acelera o Desenvolvimento de Software (uso de Migrations agiliza bastante)

Tira a necessidade do Desenvolvedor conhecer T-SQL, por exemplo.

**Desvantagens:**

Têm uma tendência a maior lentidão que em sistemas sem ORM

O Desenvolvedor tem um maior controle quando domina a Liguagem do Banco de Dados ao invés de ficar dependente do ORM.

**10. O que são microsserviços?**

Microserviços são componentes que realizam tarefas específicas e que podem ser acoplados ao sistema que o necessitar ao invés de fazer parte e estar dentro do sistema que utiliza aquela tarefa.

**10.1. Quais são suas vantagens e desvantagens?**

Vantagens: Escalabilidade, Manutenção mais Eficiente, Facilidade de Implantação, Agilidade para Entregar;

Desvantagens: Exigem uma equipe de programadores altamente especializados (com SOLID e Design Patters, por exemplo).