

**RESPOTAS DO COLABORADOR:** Renisson Machado Santos  
**Códigos no Github:** [GitHub Renisson](#)

## **RESPOSTAS SOBRE C#**

### **1. Em quais linguagens o C# foi inspirado?**

C# foi inspirada em três linguagens: C, C++ e Java. Do C herdou sua sintaxe, muitas palavras reservadas e operadores. De C++ herdou o modelo de objetos, do Java teve muita influência na parte de verificação de erros tem tempo de execução, segurança e gerenciamento da execução. Pode se dizer que o C# reúne a segurança de tipos do Java e a potência do C++.

### **2. Inicialmente o C# foi criado para qual finalidade?**

O .NET foi criado para suprimir um dos maiores problemas de interoperabilidade das linguagens de programação com o sistema Windows. Um exemplo era o Java, que apesar de rodar no Windows com sua JVM, a Microsoft sentia necessidade de corrigir esse problema.

Então a Microsoft montou uma equipe para desenvolver em sua estratégia global uma linguagem que tivesse total integração com o sistema Windows e nomeou o arquiteto-chefe Anders Hejlsberg, um grande especialista mundial na linguagem de programação.

### **3. Quais os principais motivos para a Microsoft ter migrado para o Core?**

A necessidade de cruzar plataformas, ou seja, ter uma linguagem capaz de ser executada em Windows, macOS e Linux. Modernizar a linguagem, colocando por exemplo conceitos como: programação assíncrona e estruturas e recursos para contêineres. Consistências entre ambientes para que o mesmo código que for executado em uma plataforma x64, x86 ou ARM tenha o mesmo resultado.

A Flexibilidade que o .NET Core disponibiliza com este recursos, permite que a Microsoft adentre em vários mercados da tecnologia.

### **4. Cite as principais diferenças entre .Net Full Framework e .Net Core?**

- O .NET Core é multiplataforma, ou seja, pode ser executado em Windows, Linux ou macOS.
- O .NET Core permite a sua utilização em contêineres do Docker por exemplo.
- O .NET Core é Open-Source.
- O .NET Core suporta UWP.

## **RESPOSTAS SOBRE POO**

### **1. O que é POO?**

POO significa Programação Orientada a Objetos. O POO veio para quebrar paradigma da programação estruturada e permitir a manipulação de objetos, a herança, polimorfismo, encapsulamento e classes.

Esse paradigma de objetos permite expandir a maneira em que os softwares podem ser desenvolvidos, facilitando o reaproveitamento de código e evitando códigos duplicados na construção dos Softwares.

## **2.O que é polimorfismo?**

O polimorfismo permite a utilização de objetos abstratos para determinar o comportamento dos objetos concretos.

Isso permite ter um melhor reaproveitamento de código e evitar erros ou divergências na implementação das classes concretas.

## **3.O que é abstração?**

É a capacidade de transmitir características de um objeto "MAIOR" para uma entidade "MENORES" existente na modelagem de software.

Um exemplo é uma Classe chamada PESSOA, onde um funcionário é um cliente são uma Pessoa. Logo devem ter as características de uma pessoa.

## **4.O que é encapsulamento?**

Encapsulamento pode ser visto como uma forma de dividir propriedades na orientação a objetos. Restringindo atributos, funções ou ações a objetos específicos. Utilizando o exemplo acima se quisermos restringir uma método de mostrar idade a entidade PESSOA podemos definir como private, assim mesmo que funcionario herde de pessoa ele não terá acesso a este método.

São utilizadas as palavras reservadas private, public e protected para definir o acesso. Onde public permite acesso aberto, private acesso privado e protected acesso apenas a membros que herdarem da entidade.

## **5.Quando usar uma classe abstrata e quando devo usar uma interface?**

As interfaces são interessantes para criar um "Contrato" para a entidade a qual será desenvolvida. Isso permite obrigar a criação de métodos que devem ser implementados de qualquer forma.

A classe abstrata permite o reaproveitamento de código e métodos por meio de herança. Um bom exemplo para classes abstratas é o exemplo citado acima de PESSOA.

## **6.O que faz as interfaces IDisposable, IComparable, ICloneable e IEnumerable?**

- IDisposable: Utilizado para liberar recursos não gerenciáveis (EX: Forçar a coleta de lixo);
- IComparable: Implementada para que os tipos e valores possam ser ordenados ou classificados. O CompareTo é um bom exemplo do uso dessa interface;
- ICloneable: Permite criar uma cópia de um objeto existente e fornecer sua implementação personalizada;
- IEnumerable: Essa interface é a base de todas as coleções não genéricas que podem ser enumeradas. Implementar permite utilizar recursos como foreach.

## **7.Existe herança múltipla (de classes) em C#?**

Não é permitido.

## **RESPOSTAS SOBRE TESTES**

### **1. Quais os principais frameworks que podemos usar no desenvolvimento de testes?**

R: MS Test, NUnit, xUnit, Moq, NSubstitute e Fluent Assertions.