




Tipos de Construtores

Leonardo de Lima Buta
Desenvolvedor .NET - Expert DIO

Mais sobre mim

- Desenvolvedor FullStack .NET, Nuvem, Mobile
- O que me motiva é mudar a vida das pessoas através da tecnologia
- Gosto de viajar, jogos, e tudo do mundo da tecnologia
- Canal Youtube: Leonardo Buta 
- GitHub: <https://github.com/leonardo-buta> 
- LinkedIn: <https://www.linkedin.com/in/leonardo-buta> 

Objetivo do curso

Apresentar como funciona um construtor, seu uso e funcionalidades, tipos e modos de usar na linguagem C#.

Apresentar também delegates e eventos.

Percurso

Aula 1

Construtores

Aula 2

Getters e Setters

Aula 3

Modificadores

Aula 4

Delegates

Aula 5

Eventos

Dúvidas durante o curso?

- > Fórum do curso
- > Comunidade online (discord)

Aula 1: Construtores

Tipos de Construtores

Construtores

Um construtor é um método especial, que contém o mesmo nome do seu tipo classe, e tem o objetivo de definir valores padrão, limitar uma instância e facilitar a instanciação de um objeto.

- Um construtor não possui um retorno.
- Um construtor padrão é sempre definido quando não declaramos nenhum para sua classe.
- Uma classe pode ter mais de um construtor.



Construtores

```
public class Pessoa
{
    private string nome;

    public Pessoa() 1
    {
    }

    public Pessoa(string nome) 2
    {
        this.nome = nome;
    }
}

static void Main(string[] args)
{
    Pessoa p1 = new Pessoa(); 3
    Pessoa p2 = new Pessoa("Leo");
}
```

Construtores

Chamando os construtores

Aula 2:

Getters e Setters

Tipos de Construtores

Get e Set

O Get e Set serve para que você possa atribuir um valor em uma variável de maneira controlada e com validações.

```
namespace ExemploConstrutores.Models
{
    public class Data
    {
        private int mes; 1
        public int GetMes() 2
        {
            return this.mes;
        }
        public void SetMes(int mes) 3
        {
            if (mes > 0 && mes <= 12)
            {
                this.mes = mes;
            }
        }
    }
}
```

Variável

Get

Set

Aula 3:

Modificadores

Tipos de Construtores

Readonly

O modificador readonly (somente leitura) bloqueia um campo contra alterações que não sejam em sua inicialização ou pelo próprio construtor.

```
public class Pessoa
{
    private readonly string nome = "Leo";
    private readonly string sobrenome;

    public Pessoa(string nome, string sobrenome)
    {
        this.nome = nome;
        this.sobrenome = sobrenome;
    }

    public void AlterarNome()
    {
        this.nome = "Nome alterado";
    }
}
```

Constante

Uma constante representa um valor que somente pode ser atribuído no momento de sua inicialização, e não pode ser modificado posteriormente.

```
static void Main(string[] args)
{
    const double pi = 3.14;
    System.Console.WriteLine(pi);

    pi = 0;
```

Aula 4:

Delegates

Tipos de Construtores

Delegates

Um delegate é uma maneira de passar um método como referência, podendo ser usado como um callback, aceitando qualquer método que contenha a mesma assinatura.

```
public class Calculadora
{
    public static void Somar(int num1, int num2)
    {
        System.Console.WriteLine($"Adição: {num1 + num2}");
    }

    public static void Subtrair(int num1, int num2)
    {
        System.Console.WriteLine($"Subtração: {num1 - num2}");
    }
}
```

```
public delegate void operacao(int x, int y);

static void Main(string[] args)
{
    operacao op = Calculadora.Somar;
    op.Invoke(10, 10);
}
```

Aula 5: Eventos

Tipos de Construtores

Eventos

Eventos é um mecanismo de comunicação entre objetos, onde existe um publisher, que realiza o evento e o subscriber, que se inscreve em um evento.

```
public class Calculadora
{
    // Delegate
    public delegate void DelegateCalculadora();

    // Evento
    public static event DelegateCalculadora EventoCalculadora;

    public static void Somar(int x, int y)
    {
        if (EventoCalculadora != null)
        {
            System.Console.WriteLine($"Adição: {x + y}");
            EventoCalculadora();
        }
        else
        {
            System.Console.WriteLine("Nenhum inscrito");
        }
    }
}
```

Dúvidas?

- > Fórum do curso
- > Comunidade online (discord)

Próximos passos

Aprimorar seus conhecimentos com C#

- Construtores
- Delegates e Eventos
- Modificadores
- Propriedades

<https://www.c-sharpcorner.com/UploadFile/84c85b/delegates-and-events-C-Sharp-net/>

<https://www.tutlane.com/tutorial/csharp/csharp-events>



DIGITAL
INNOVATION
ONE

Agradecimentos



<https://www.linkedin.com/in/leonardo-buta/>



Leonardo Buta



<https://github.com/leonardo-buta>