



# Aula 12

Prof: Henrique Augusto Maltauro

# Desenvolvendo Algoritmos

# Métodos

# Métodos

Um método é um bloco de código que contém uma série de instruções.

Normalmente, uma aplicação de software faz com que as instruções sejam executadas através de um método.

# Métodos

No C#, todas as instruções executadas são realizadas no contexto de um método.

O método **Main** é o ponto de início para todos os aplicativos C# e é executado quando o programa é iniciado.

# Métodos

- C#: Declaração

Para fazer a **declaração** de um **método**, precisamos de 3 coisas:

- Modificador de acesso
- Tipo de retorno
- Nome

# Métodos

- C#: Declaração (modificador de acesso)

O modificador de acesso NÃO vai ser explicado aqui, para não confundir o seu entendimento sobre os métodos nesse momento, e nós vamos abordar esse assunto quando formos estudar sobre Programação Orientada a Objetos.

Para todos os efeitos, no momento, vamos utilizar o modificador de acesso `public static`.

# Métodos

- C#: Declaração (tipo de retorno)

Os métodos podem retornar um valor, e obrigatoriamente, nós precisamos definir o tipo de retorno de um método, inclusive, se o tipo de retorno dele for nada.

O tipo de retorno de um método, pode ser qualquer tipo de dado, e para métodos que não retornam nada nós definimos o tipo de retorno como void.

# Métodos

- C#: Declaração (nome)

Todos os métodos precisam ter um nome único, o qual vai ser utilizado para chamar a sua execução.



# Métodos

- C#: Declaração

```
public static void ExecutarSoma()  
{  
    // Bloco de código  
}
```

# Métodos

- C#: Chamadas

A **chamada** de um **método** é o processo de acessar e executar o **método**, e esse processo é feito através do nome do método, por isso ele precisa ser único, para que na hora de executar um **método** o computador saiba qual **método** executar.

# Métodos

- C#: Chamadas

```
public static void Main()
{
    ExecutarSoma();
}

public static void ExecutarSoma()
{
    // Bloco de código
}
```

# Métodos

- C#: Parâmetros

No processo de **declaração** de um **método**, opcionalmente, podemos definir **parâmetros**, para passar valores para dentro do **método**.

Com isso, a execução do **método** pode ser feita de forma personalizada, já que cada execução do **método** pode ter **parâmetros** com valores diferentes.

# Métodos

- C#: Parâmetros

Os **parâmetros** são como variáveis internas do **método**, que recebem valores pré-definidos de fora do **método**.

A declaração desses **parâmetros** deve respeitar a mesma estrutura de declaração de variáveis normais, porém, separados por vírgula.

# Métodos

- C#: Parâmetros

```
public static void ExecutarSoma(int numero1, int numero2)
{
    // Bloco de código
}
```

# Métodos

- C#: Chamadas com parâmetros

```
public static void Main()  
{  
    ExecutarSoma(4, 5);  
}  
  
public static void ExecutarSoma(int numero1, int numero2)  
{  
    // Bloco de código  
}
```

# Métodos

- C#: Retorno

Conforme já dito, os métodos podem ter um retorno, ou seja, eles podem devolver um valor depois de terminar de executar o seu bloco de código, que deve ser especificado pela palavra-chave return.

Conforme já dito, o tipo de retorno é definido na declaração do método, e esse retorno pode ser atribuído a uma variável.



# Métodos

- C#: Retorno sem parâmetros

```
public static int ExecutarSoma()  
{  
    return /* Valor de retorno */;  
}
```

# Métodos

- C#: Chamadas com retorno e sem parâmetros

```
public static void Main()
{
    int soma = ExecutarSoma();
}

public static int ExecutarSoma()
{
    return /* Valor de retorno */;
}
```

# Métodos

- C#: Retorno com parâmetros

```
public static int ExecutarSoma(int numero1, int numero2)
{
    return numero1 + numero2;
}
```

# Métodos

- C#: Chamadas com retorno e com parâmetros

```
public static void Main()  
{  
    int soma = ExecutarSoma(4, 5);  
}  
  
public static int ExecutarSoma(int numero1, int numero2)  
{  
    return numero1 + numero2;  
}
```

# Exercício

## Exercício

1 - Receber dois valores `int` e apresentar no console a adição, subtração, divisão e multiplicação dos dois valores. Porém, cada um desses cálculos devem ser feitos através de `métodos` com `retorno`.

## Exercício

2 - Receber três valores **int** e apresentar no console os valores digitados e a média dos valores.

O processo de receber cada valor, deve ser feito em **métodos diferentes**, que retornem esses valores.