



C#



# Sumário

```
arrays.cs
using System;
class DeclareArray
{
    public static void Main()
    {
        // Single-dimensional array
        int[] numbers = new int[5];

        // Multidimensional array
        string[,] names = new string[5,4];

        // Jagged array
        byte[][][] jagged = new byte[5][][];

        for (int i = 0; i < jagged.Length; i++)
        {
            for (int j = 0; j < jagged[i].Length; j++)
            {
                for (int k = 0; k < jagged[i][j].Length; k++)
                {
                    jagged[i][j][k] = (byte) i * j * k;
                }
            }
        }
    }
}
```

The image features a stylized cartoon character with brown hair and a red shirt, looking towards the left. The background is divided into three vertical sections: a red pixelated area on the left, a light blue area with white circuit lines in the middle, and a purple area with white binary code and circuit lines on the right. The C# .NET logo is positioned on the left side, with 'Microsoft' in small text above '.net'.

Microsoft  
C#.net™

# Sumário

*1-Fundador*

*2-O que é C#*

*3-Características do C#*

*4-Vantagens e desvantagens do C#*

*5-5 Frameworks populares do C#*

*6-Desenvolvendo uma calculadora em C#*

*7-Oficina do C#*

*8-Componentes da equipe*



Fundador

```
arrays.cs
using System;
class DeclareArray
{
    public static void Main()
    {
        // Single-dimensional array
        int[] numbers = new int[5];

        // Multidimensional array
        string[,] names = new string[5,4];

        // Jagged array
        byte[][] scores = new byte[5][];

        for (int i = 0; i < scores.Length; i++)
        {
            scores[i] = new byte[scores[i].Length];
        }
    }
}
```



*Anders Hejlsberg*



*Marcos conquistados*

*1-Ambiente de desenvolvimento Turbo Pascal*

*2-Compilador Delphi*

*3-Plataforma de desenvolvimento e execução de sistemas e aplicações chamado .NET*



O que é o C#

```
arrays.cs
using System;
class DeclareArray
{
    public static void Main()
    {
        // Single-dimensional array
        int[] numbers = new int[5];

        // Multidimensional array
        string[,] names = new string[5,4];

        // Jagged array
        byte[][] data = new byte[5][];

        for (int i = 0; i < data.Length; i++)
        {
            data[i] = new byte[5];
        }

        // Accessing elements
        Console.WriteLine(data[0][0]);
        Console.WriteLine(data[1][1]);
        Console.WriteLine(data[2][2]);
        Console.WriteLine(data[3][3]);
        Console.WriteLine(data[4][4]);
    }
}
```

*Boa noite! Sou o assistente virtual Vilton! Estarei fornecendo algumas informações sobre a Linguagem de programação C# durante a apresentação. Então sejam bem vindos e vamos nessa!*

*C# é uma linguagem de programação, multiparadigma, de tipagem forte, desenvolvida pela Microsoft como parte da plataforma .NET.*

*A sua sintaxe orientada a objetos foi baseada no C++ mas inclui muitas influências de outras linguagens de programação, como Object Pascal e, principalmente, Java.*

*O código fonte é compilado para Common Intermediate Language (CIL) que é interpretado pela máquina virtual Common Language Runtime (CLR).*

*C# é uma das linguagens projetadas para funcionar na Common Language Infrastructure da plataforma .NET Framework.*





## Características do C#

```
arrays.cs
using System;
class DeclareArray
{
    public static void Main()
    {
        // Single-dimensional array
        int[] numbers = new int[5];

        // Multidimensional array
        string[,] names = new string[5,4];

        // Jagged array
        byte[,] = new byte[5][1];

        for (int i = 0; i < array.Length; i++)
        {
            for (int j = 0; j < array[i].Length; j++)
            {
                scores[i].Length;
            }
        }
    }
}
```



# Características do C#

- *Simplicidade*
- *Completamente orientada a objetos*
- *Fortemente tipada*
- *Gera código gerenciado*
- *Tudo é um objeto*
- *Controle de versões*
- *Suporte a código legado*
- *Flexibilidade*
- *Linguagem gerenciada*

```
)).done(function(response) {  
    for (var i = 0; i < response.length; i++) {  
        var layer = L.marker(  
            [response[i].latitude, response[i].longitude]  
            // , {icon: myIcon}  
        );  
        layer.addTo(group);  
  
        layer.bindPopup(  
            "<p>" + "Species: " + response[i].species + "</p>" +  
            "<p>" + "Description: " + response[i].description + "</p>" +  
            "<p>" + "Seen at: " + response[i].latitude + " " + response[i].longitude + "</p>" +  
            "<p>" + "On: " + response[i].sighted_at + "</p>"  
        )  
    }  
  
    $('select').change(function() {  
        species = this.value;  
    });  
});  
}  
$.ajax({  
    url: queryURL,  
    method: "GET"  
}).done(function(response) {  
    for (var i = 0; i < response.length; i++) {  
        var layer = L.marker(  
            [response[i].latitude, response[i].longitude]  
            // , {icon: myIcon}  
        );  
        layer.addTo(group);  
    }  
});
```



# *Vantagens e Desvantagens do C#*

```
arrays.cs
using System;
class DeclareArray
{
    public static void Main()
    {
        // Single-dimensional array
        int[] numbers = new int[5];

        // Multidimensional array
        string[,] names = new string[5,4];

        // Jagged array
        byte[][] f_arrays = new byte[5][];
        f_arrays[0] = new byte[5][1];

        for (int i = 0; i < f_arrays.Length; i++)
        {
            for (int j = 0; j < f_arrays[i].Length; j++)
            {
                f_arrays[i][j] = (byte)j;
            }
        }
    }
}
```

*1-Integração com  
Windows*

*2-Linguagem compilada*

*3-Modelo de  
desenvolvimento de  
software orientado a objetos*

*4-Desenvolvimento de Jogos  
no programa Unity*

*5-Aplicações de servidor*

*6-Aplicações de Desktop*

*Vantagens*

*Desvantagens*

*1-Código compilado*

*2-A Microsoft deixou de  
suportar a .NET*

*3-Fugas de memória*



## 5 Frameworks populares do C#

```
arrays.cs
using System;
class DeclareArray
{
    public static void Main()
    {
        // Single-dimensional array
        int[] numbers = new int[5];

        // Multidimensional array
        string[,] names = new string[5,4];

        // Jagged array
        byte[][][] f_arrays = new byte[5][1];

        for (int i = 0; i < f_arrays.Length; i++)
        {
            for (int j = 0; j < f_arrays[i].Length; j++)
            {
                f_arrays[i][j] = new byte[5][1];
            }
        }
    }
}
```



## *O que é um framework?*

*Framework é um conjunto de códigos prontos com diversas funções que podem ser utilizadas no desenvolvimento de sites. Sobretudo, o objetivo de uso dessa ferramenta é aplicar funcionalidades e estruturas já programadas para garantir mais produtividade e qualidade no desenvolvimento de um projeto.*

*1-ASP.NET MVC*

*2-ASP.NET WEB API*

*3-NancyFX*

*4-SignalR*

*5-Cash Cow*



# *Desenvolvendo uma calculadora em C#*

```
arrays.cs
using System;
class DeclareArray
{
    public static void Main()
    {
        // Single-dimensional array
        int[] numbers = new int[5];

        // Multidimensional array
        string[,] names = new string[5,4];

        // Jagged array
        byte[][] data = new byte[5][];

        for (int i = 0; i < data.Length; i++)
        {
            data[i] = new byte[5];
        }

        // Jagged array of arrays
        int[][] scores = new int[5][];

        for (int i = 0; i < scores.Length; i++)
        {
            scores[i] = new int[5];
        }
    }
}
```

*Aqui você aprenderá os conceitos básicos sobre variáveis, tipos de dados, condicionais e a estrutura de repetição switch case em C# que auxiliará no melhor entendimento do código.*



*Vamos detonar galera do Youtube!*



# Oficina do C#

```
arrays.cs
using System;
class DeclareArray
{
    public static void Main()
    {
        // Single-dimensional array
        int[] numbers = new int[5];

        // Multidimensional array
        string[,] names = new string[5,4];

        // Jagged array
        byte[][] scores = new byte[5][];

        for (int i = 0; i < scores.Length; i++)
        {
            scores[i] = new byte[scores[i].Length];
        }
    }
}
```



*Após o aprendizado adquirido no tópico anterior, iremos agora para a parte prática onde desenvolveremos juntos as 2 últimas operações matemáticas.*



# *Lógica de programação*

*Componentes da equipe:*

*Thiago de Pinho Santos Santana*

*Lucas do Rosario dos Santos*

*Marcos Vinícius Nascimento Moreira*

*Eduardo Gonzaga dos Santos Junior*