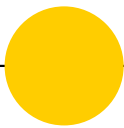


Introdução à Plataforma .NET



Profa. Luciana Balieiro Cosme

Ciência da Computação/IFNMG Montes Claros

Template SlidesCarnival



Desenvolvimento para Ambiente .NET

Ementa: Conceito de desenvolvimento de aplicativos utilizando a **plataforma .NET**. Acessando um **banco de dados** com o uso do IDE. Utilização do ADO.NET, **ASP.NET** e formulários Windows. Criação de Web Services XML. Conceitos sobre a utilização da estrutura .NET. Definição de **variáveis** de memória, expressões e operadores, **estruturas** de controle, utilização das **funções** internas, criação de procedimentos, escopo de variáveis, arrays, criação de menus personalizados, **formulários**, definição de controles, depuração de código, e **geração de aplicações**.



Avaliação

- ⦿ Exercícios práticos (30 pontos).
- ⦿ Avaliação (30 pontos)
- ⦿ Projeto de conclusão da disciplina (40 pontos).



Plataforma .NET

- ◉ Ambiente de desenvolvimento de software
- ◉ Microsoft
- ◉ Ferramentas, linguagens e bibliotecas
- ◉ Aplicativos para uma variedade de dispositivos e sistemas operacionais.



Principais Componentes

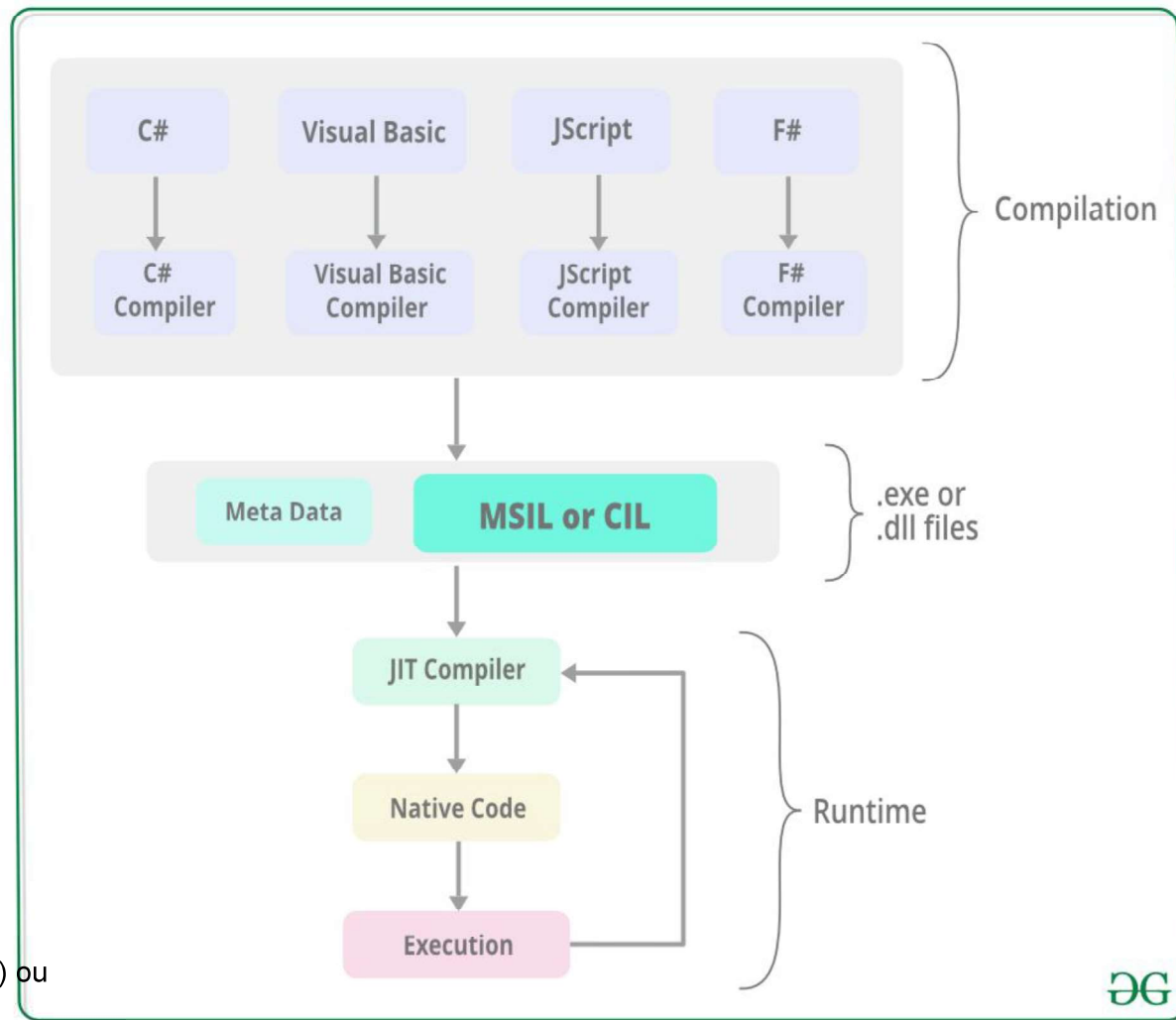
- ⦿ Common Language Runtime (CLR)
 - coleta de lixo, gerenciamento de memória, segurança e suporte a várias linguagens
- ⦿ Biblioteca de Classes
- ⦿ Compilador => código fonte para o formato intermediário

Processo de Execução.

Fonte:

<https://www.geeksforgeeks.org/cil-or-msil-microsoft-intermediate-language-or-common-intermediate-language/>

* Microsoft Intermediate Language (MSIL) ou Common Intermediate Language (CIL)






“Vantagens”

- ◉ Suporte a várias linguagens
- ◉ Várias plataformas (Windows, macOS e Linux)
- ◉ Comunidade de desenvolvedores
- ◉ Bibliotecas, frameworks e ferramentas



Aplicações da Plataforma .NET

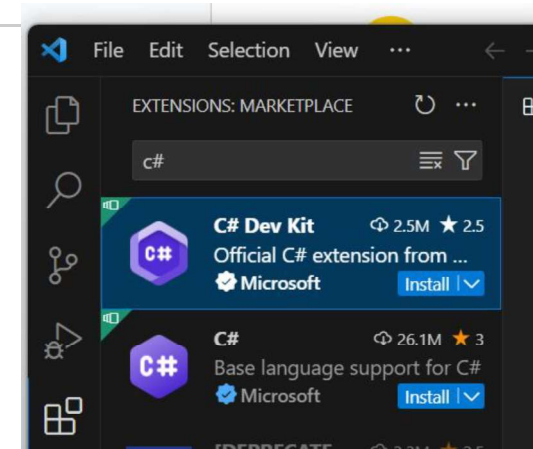
- Aplicativos de desktop: Windows Forms, WPF.
- Aplicativos web: ASP.NET, Blazor.
- Aplicativos móveis: Xamarin.
- Serviços e APIs: ASP.NET Web API, gRPC.
- Jogos: Unity.



Introdução à Linguagem C#

● Hello, World!

- Pré-requisitos: VS. Code + extensão C# ▼
- Verifique se o dotnet está instalado (Exibir->Terminal):
dotnet --version
 - <https://dotnet.microsoft.com/pt-br/download>



▼ <https://learn.microsoft.com/pt-br/dotnet/core/tutorials/with-visual-studio-code?pivots=dotnet-7-0>



Hello, World!

- ⦿ Abra e/ou criar pasta com nome do projeto (hello world)
- ⦿ Selecione a pasta criada
- ⦿ Msg: “Você confia nos autores dos arquivos nesta pasta?” => Sim, confio nos autores.



Hello, World!

- ⦿ Abrir no terminal (Exibir->Terminal)
- ⦿ No terminal => `dotnet new console`

Hello, World!

(Instruções de nível superior ▼)

```
Console.WriteLine("Hello, World!"); ➔
```

```
namespace HelloWorld {  
    class Program {  
        static void Main(string[] args) {  
            Console.WriteLine("Hello,  
World!");  
        }  
    }  
}
```

▼ <https://aka.ms/new-console-template>



Hello, World!

- Msg: adicionar os ativos ausentes para compilar e depurar o aplicativo? => Sim
- No terminal => `dotnet run`

```
PS C:...\\projetos\\hello world> dotnet run  
Hello, World!
```

Histórico C#: <https://github.com/dotnet/csharp-lang/blob/main/Language-Version-History.md>



Entrada e saída

String interpolada:

```
Console.WriteLine($"Olá, {name}, em  
{currentDate:d}!");
```

```
Console.WriteLine("Qual é o seu nome?");  
var name = Console.ReadLine();  
var currentDate = DateTime.Now;  
Console.WriteLine("Olá " + name + ", em "  
+ currentDate + "!");
```

(saída)

>Qual é o seu nome?

>Luciana

>Olá Luciana, em 19/02/2024 14:32:46!

Sistemas de tipos

◉ Fortemente tipada

```
int a = 5;  
int b = a + 2;  
bool test = true;  
int c = a+test; //??
```

Msg: Operator '+' cannot be applied
to operands of type 'int' and 'bool'

```
bool d = (2 == 3) ;  
float x = 3/2;  
//1 ou 1.5??
```




Declaração de variáveis/inicialização

- Declaração

```
float temperature;
```

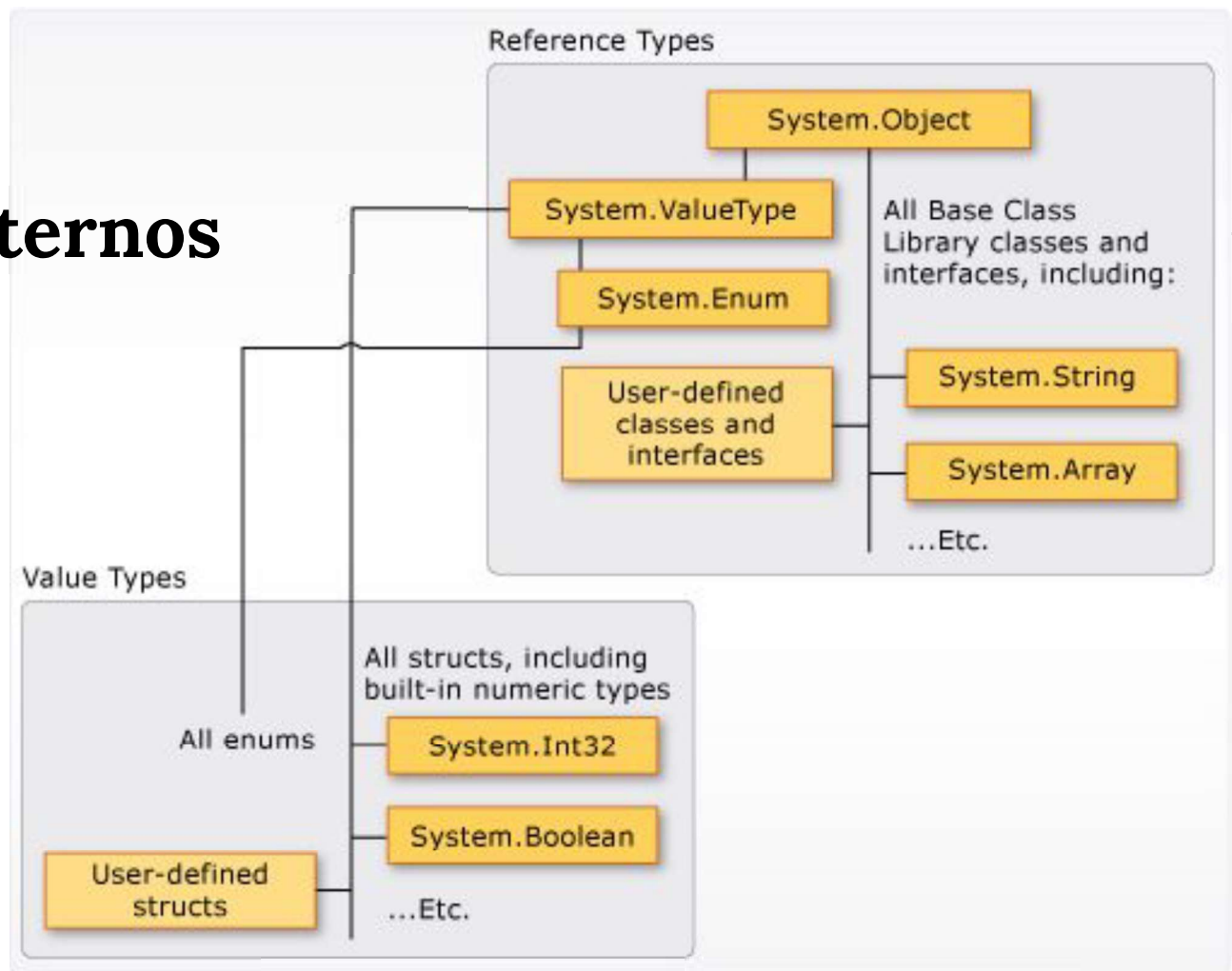
- Inicialização

```
int[] vetor = { 0, 1, 2, 3, 4, 5 };
```


```
int limite = 3;
```

Tipos internos

bool, byte,
char, int, long,
float, double...



Fonte: <https://learn.microsoft.com/pt-br/dotnet/csharp/fundamentals/types/>



Common Type System(CTS)

- Suporta herança
- System.Object (tipo base)
- Tipos por valor e por referência
- Exemplos:

```
Console.WriteLine($"int.MaxValue={int.MaxValue}");
```

```
string s = i.ToString();
```

```
string t = 123.ToString();
```



Tipos de valor

- ◉ `System.Object` => `System.ValueType`
- ◉ São selados (não é possível derivar)
- ◉ Contém diretamente seus valores
- ◉ Principais tipos: `short`, `int`, (...) , `float`, `double`, `decimal`, `char`, `bool`, `enum` e `struct`



Tipos de valor (compostos)

```
public struct Coords {  
    public int x, y;  
    public Coords(int p1, int p2) {  
        x = p1; y = p2;  
    }  
}  
  
public enum TipoPessoa{  
    PF= 1, PJ = 2,  
}
```

Tipos de referência

- Principais tipos: classes, vetores e matrizes
- null quando não inicializados
- podem ser herdados

```
int[] nums = { 1, 2, 3, 4, 5 };
```

```
int [] nums2 = nums;
```

```
nums2[3] = 5;
```

```
Console.WriteLine (nums[3]); //??
```



Tipos de valores literais

```
string s = "A resposta é " + 5.ToString();  
Console.WriteLine(s);  
Type type = 12345.GetType();  
Console.WriteLine(type);
```

A resposta é 5

System.Int32

Membros String

```
string a = "string";  
string b = a.Replace("i", "o"); //strong  
      b = a.Insert(0, "a "); //a string  
      b = a.Remove(0, 3); //ing  
      b = a.Substring(0, 3); //str  
      b = a.ToUpper(); //STRING  
int i = a.Length; //6
```




Tipos genéricos

```
using System.Collections.Generic;
```

```
List<string> stringLista = new  
List<string>();  
stringLista .Add("String de exemplo");
```

```
stringLista.Add(4); //????
```

● Tipos implícitos

- variável local: palavra chave **var**

```
var s = "texto qualquer";  
Console.WriteLine(s.GetType());  
s=1;    //????
```

Saída: System.String



Exercício 1.1

É possível comparar tanto tipos de valor quanto tipos de referência usando o método *Equals*? Dê, pelo menos, três exemplos distintos de uso com tipos de valor e referência.



Exercício 1.2

Como atribuir o valor de um array 'a' para um array 'b' e, em seguida, modificar 'b' sem afetar o array original?



Exercício 1.3

O que será impresso no seguinte código?

```
List<string> nomes = new List<string>();  
nomes.AddRange(new [] { ".net", "2023",  
"ifnmg" });  
Console.WriteLine(nomes);
```



Exercício 1.4

Pesquise e apresente uma forma de exibir todos os elementos da variável “*nomes*” do exemplo anterior, sem usar uma estrutura de repetição, nos moldes tradicionais.



Referências

- Hello World =>
<https://learn.microsoft.com/pt-br/dotnet/core/tutorials/with-visual-studio-code?pivot=dotnet-7-0>
- Sistemas de tipos = >
<https://learn.microsoft.com/pt-br/dotnet/csharp/fundamentals/types/>
- Tipos primitivos =>
<https://learn.microsoft.com/pt-br/dotnet/csharp/fundamentals/types/https://learn.microsoft.com/pt-br/dotnet/csharp/language-reference/builtin-types/built-in-types>