Introdução a

linguagem C#

Leite, Mário; Leite, Mário. Linguagem C#: Com Acesso a Bancos de Dados. Sinergia Casa Editorial. Edição do Kindle.

Objetivos

1. O que framework.NET?
2. O que é linguagem C#?
3. Orientações importantes
4. Ambiente de programação

Framework

- **REUSABILIDADE;**
- Conjunto de bibliotecas ou componentes que são usados para criar a base onde seu software será construído!
- Um framework que nós já conhecemos e já usamos: Bootstrap!

Framework .NET

- É um ambiente de execução gerenciado para o Windows que oferece uma variedade de serviços
- Ele consiste em dois componentes principais: o CLR (Common Language Runtime), o mecanismo de execução que manipula aplicativos em execução, e a biblioteca de classes.
- Versão atual é a 4.8

A linguagem C#

- Orientação a objetos é o paradigma mais empregado no mundo da programação;
- C# é uma das linguagens mais empregadas quando o assunto é orientação a objetos;
- É orientada a objetos e a eventos;
- É multiparadigma e de tipagem forte;
- Foi desenvolvida para o framework .NET;
- Com C# podemos desenvolver aplicações: desktop, mobile, web e nuvem;

Por que #?

- É a sobreposição de quatro operadores +, estendendo o par de C++;
- Sustenido (#) é uma nota musical está um tom acima dando um sentido maior que C e C++;

Visual Studio

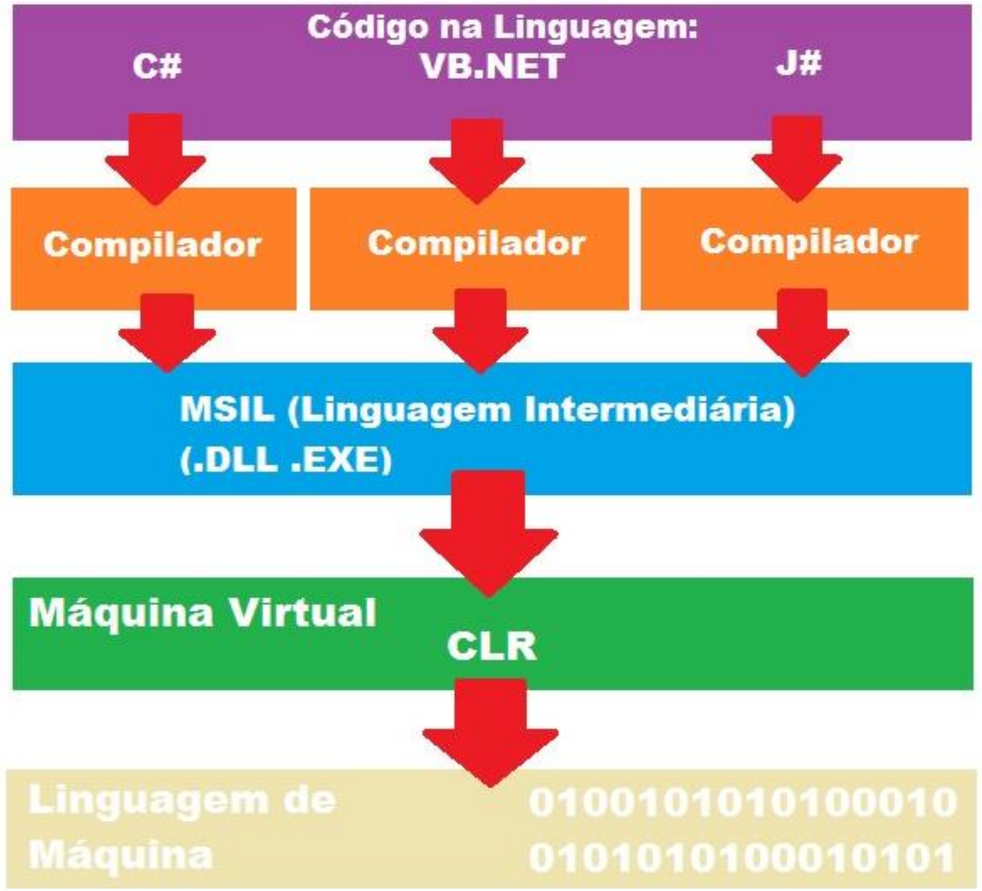
Ambiente integrado de desenvolvimento incluindo inteface gráfica e código;

Visual Studio Code

Editor de códigos;

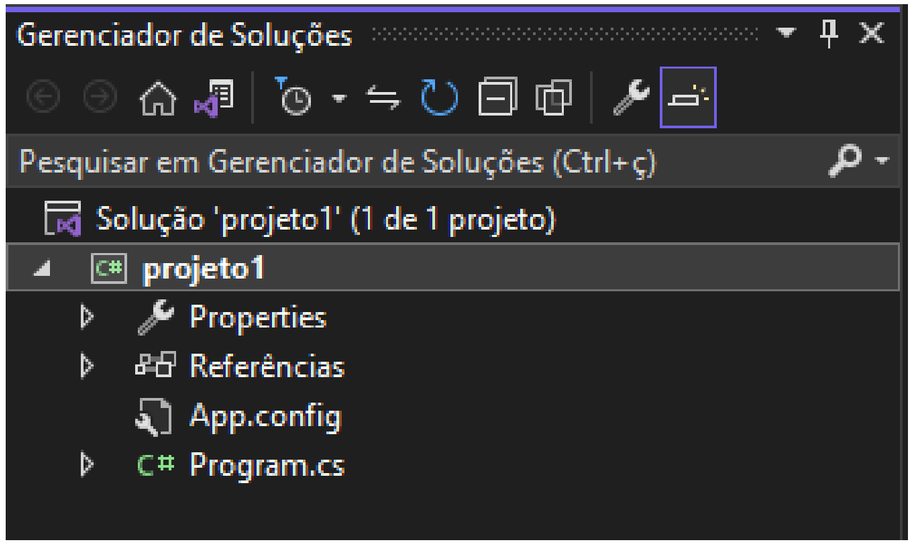
Histórico

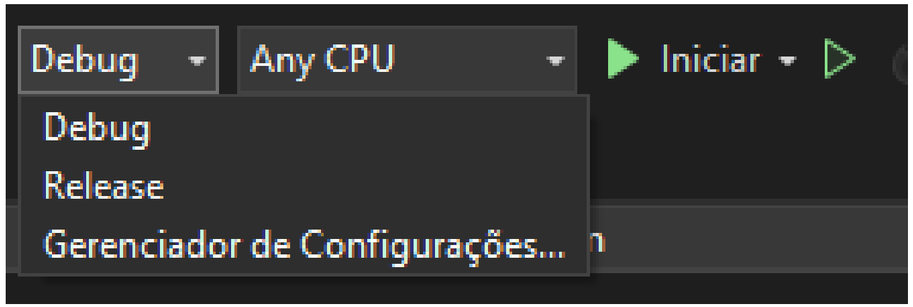
- C# é uma evolução da linguagem C;
- Foi criada em 2000, por Henjsberg, na Microsoft;
- No paradigma de orientação a objetos sua sintaxe é baseada na linguagem C++;
- Tem influência de outras linguagens como o Objective C e Java;



Compilação

- Seu código fonte é compilado;
- Gera um código intermediário chamado Intermediate Language (IL) que é interpretado por uma máquina virtual Commom Language Rumtime (CLR) e depois para código executável.





Debug e release

- Os códigos C# são representados pela extensão .cs e são apresentados no gerenciador de soluções;
- Para localizar os arquivos executáveis basta abrir a pasta no gerenciador de arquivos: .bin/debug ou release/.exe;
- Debug: gera um executável e procura os erros;
- Release: gera um excutável mas não procura erros;

Orientações importantes

01

C# é *case sensitive*. Difere maiúsculas de minúsculas

02

Todas as linhas decódigo terminam em **;**

03

Todos os elementos do programa devem ser declarados antes de serem iniciados

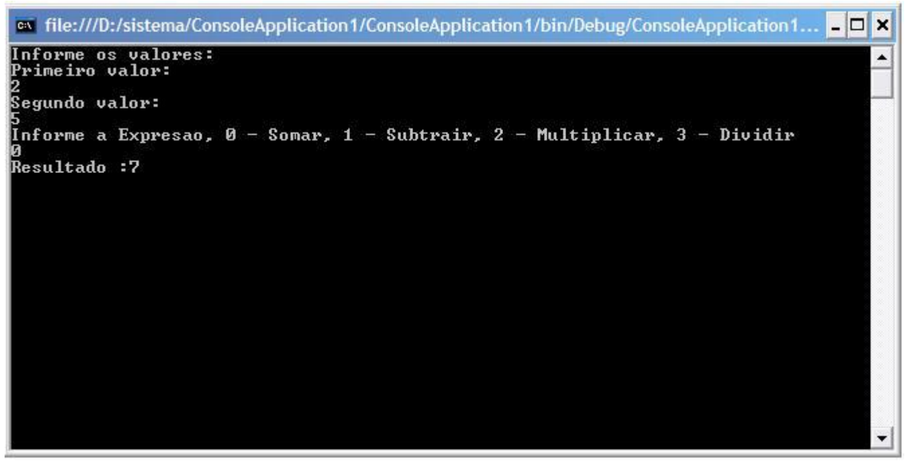
04

Os blocos de código são escritos entre **{ }**

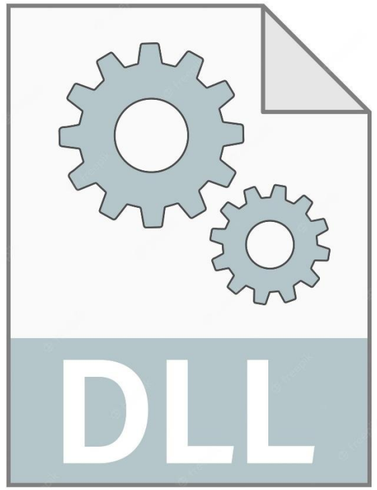
Ambiente de programação

- Vamos usar o Visual Studio 2022;
- Link para baixar a versão gratuita: https://visualstudio.microsoft.com/pt-br/vs/community/
- Depois de baixado e iniciado a instalação escolheremos os módulos que vamos usar para programar: .NET e C++;

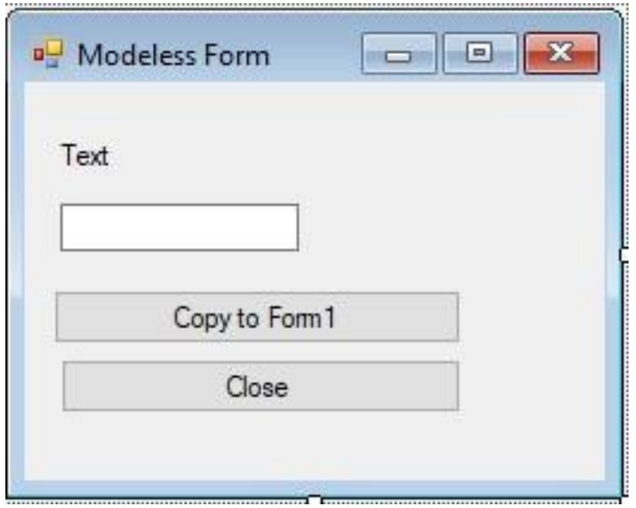
Tipos de Projetos em C#



Projeto console



Bibliotecas



Windows forms

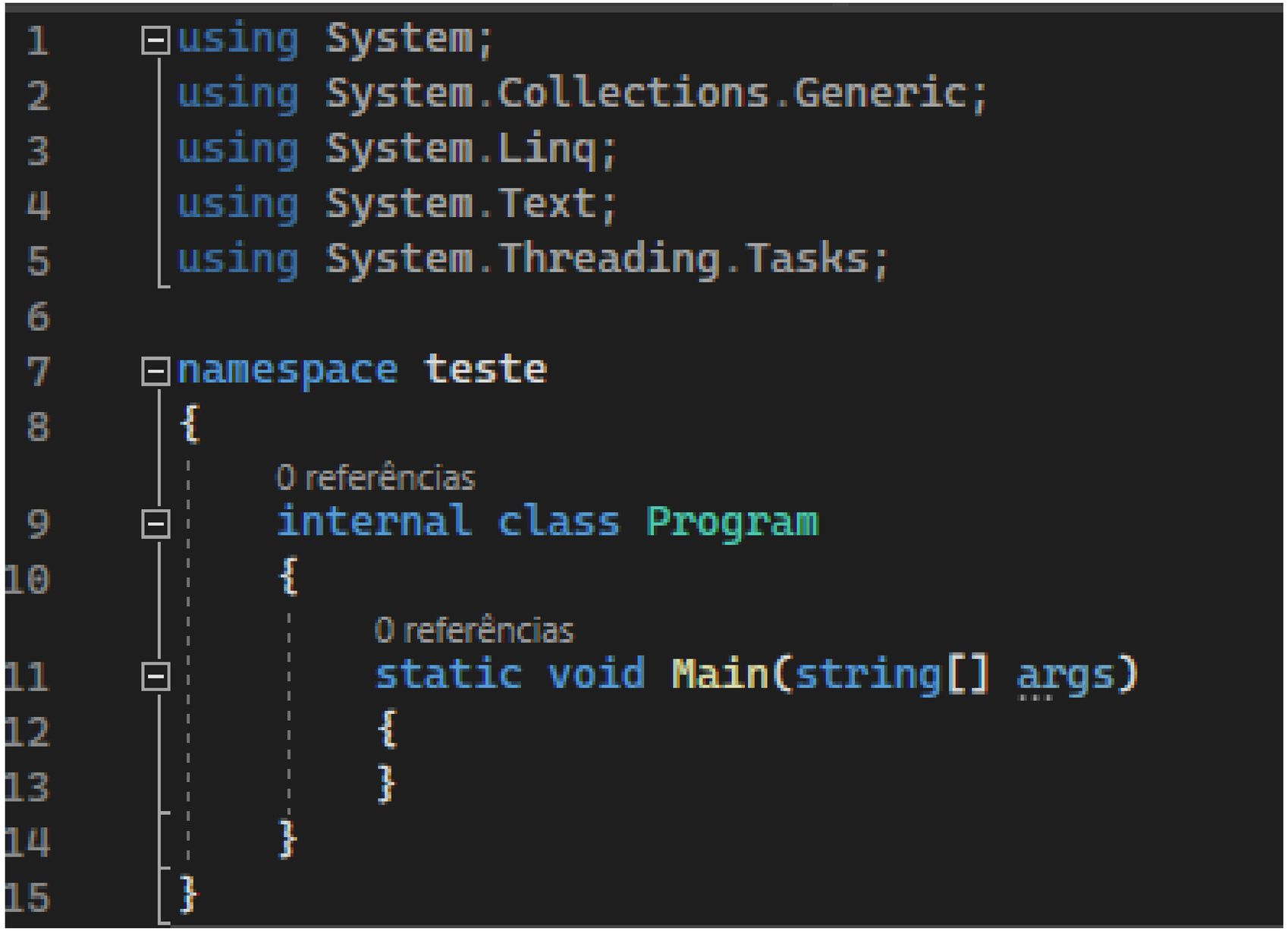
Comentários em C#

- // comentário de linha única
- /* comentário de

Várias linhas */

Estrutura básica do código

- Os programas C# podem consistir em um ou mais arquivos.
- Diretiva using + namespace;
- Namespaces fornecem um meio hierárquico de organizar bibliotecas e programas em C#. Os namespaces contêm tipos e outros namespaces;
- Internal class: criação de classes;
- Static void main: é o método mais importante pois é o primeiro a ser executado;


using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace teste
{
0 referências
internal class Program
{
0 referências
static void Main(string[] args)
{
}
}
}

Como criar o 1º projeto console

Compilando

Arquivos gerados e propriedades

Escrevendo e imprimindo texto

- Escrever linha a linha:
- Console.WriteLine(“ ”); -> classe console e método WriteLine

- Escrever palavras ou frases na mesma linha:
- Console.Write(“”);
- Pular linhas /r /n

- Travar a execução do código: Console.ReadKey();

Prática

- Hello World!

- programas\imprimir_texto

- https://www.programiz.com/csharp-programming/online-compiler/

Prática

- Imprimir uma mensagem de saudação: “Olá bem vindo ao meu programa C#!”
- Imprimir o seu nome: “Meu nome é: Carolina”;
- Imprimir o seu nome e pular uma linha usando o /r/n: “Meu nome é:
- Carolina”;

Leitura de tecla

- Console.ReadKey() lê apenas uma tecla do teclado retornando informações sobre a tecla pressionada!
- programas\leitura_tecla

Tipos de variáveis em C#

- C# suporta vários tipos de variáveis:
- Tipos numéricos inteiros: sbyte, byte, short, ushort, int, uint, long e ulong;
- Tipos numéricos de ponto flutuante: float, double, decimal;
- Tipo caractere: char;
- Tipo cadeia de caracteres: string;
- Tipos booleanos: bool;
- Tipo data: datetime;

Tipos de variáveis em C#

- Podemos escolher o tipo de variável adequado ao código de acordo com o tipo de dado que você precisa armazenar e o tamanho de memória necessário para armazená-los.

- programas\variaveis_conversoes

palavra-chave/tipo C#	Intervalo	Tamanho	Tipo .NET
`sbyte`	-128 a 127	Inteiro de 8 bits com sinal	System.SByte
`byte`	0 a 255	Inteiro de 8 bits sem sinal	System.Byte
`short`	-32.768 a 32.767	Inteiro de 16 bits com sinal	System.Int16
`ushort`	0 a 65.535	Inteiro de 16 bits sem sinal	System.UInt16
`int`	-2.147.483.648 a 2.147.483.647	Inteiro assinado de 32 bits	System.Int32
`uint`	0 a 4.294.967.295	Inteiro de 32 bits sem sinal	System.UInt32
`long`	-9.223.372.036.854.775.808 a 9.223.372.036.854.775.807	Inteiro assinado de 64 bits	System.Int64
`ulong`	0 a 18.446.744.073.709.551.615	Inteiro de 64 bits sem sinal	System.UInt64

Palavra reservada var

- É reconhecida como “variável implicitamente tipada”;
- Quando declaramos uma variável usando a palavra reservada var, o compilador determina o tipo da variável com base no valor atribuído a ela automaticamente.
- IMPORTANTE: O tipo da variável só pode ser determinado no momento da inicialização e não podemos mudar o tipo depois de iniciada;

```
var nome = "Carolina"; //nome é do tipo string
var idade = 33; //idade é do tipo int
var salario = 10000,00; //salário é do tipo double
```

Conversão de variáveis

- As conversões em C# são uma maneira de mudar o tipo de uma variável
- Convert.To
- programas\variaveis_conversoes

Prática

- Declare as seguintes variáveis usando var e mostre usando o Console.WriteLine():
- Nome;
- Idade;
- Data de nascimento;
- Salário;
- Endereço;
- Número da casa;

Prática

- Converta os tipos de variáveis e mostre:
- String numero = “100” para int;
- Byte b = 106 para char
- Double d = 1.5 para Int;

Entrada de dados

- Console.ReadLine(): É uma função em C# que lê uma linha de texto digitada pelo usuário a partir da entrada padrão (geralmente o teclado) e retorna o texto como uma string.
- programas\entrada_de_dados

Desafio!

- Crie um pequeno sistema em C# que solicitará do usuário as seguintes informações:
- Nome, idade, documento, rua, número e gênero;
- Mostre essas informações no console!