

17/2/2025

**TP1**

Fundamentos de Desenvolvimento com C#

Professor(a): Rinaldo Ferreira Júnior

## 1. EXERCÍCIO

- **.NET** é uma plataforma de desenvolvimento versátil criada pela Microsoft, com um ambiente robusto e eficiente que permite o desenvolvimento de software para Windows, Linux, macOS, Android e iOS. A mesma ainda pode ser utilizada para aplicações dos mais diversos tipos, como:
  - Desktop
  - WEB
  - CLOUD
  - Mobile
  - Gaming
  - IOT
  - IA
- **O C# se integra ao .NET**, pois foi projetado para interagir diretamente ao ecossistema da plataforma, utilizando o CLR (Common Language Runtime) para execução do código e o FCL (Framework Class Library) para acessar bibliotecas prontas, permitindo assim a escrita de código eficiente e seguro, aproveitando recursos do CLR citados logo mais abaixo.
- **O papel do CLR** é gerenciar a execução do código, pois é ele o ambiente de execução do .NET, sendo também responsável por funcionalidades como:
  - Gerenciamento de memória e garbage collection;
  - Execução do código IL;
  - Gerenciamento de threads;
  - Tratamento de exceções;
  - Verificações de segurança;
  - Compilação Just-In-Time (JIT) do código IL para código nativo da máquina.
- **O papel do FCL** é ser uma biblioteca abrangente de classes, interfaces e tipos reutilizáveis que aceleram o desenvolvimento de aplicações. Ela fornece:
  - Funcionalidades básicas como manipulação de strings e coleções
  - Acesso a banco de dados
  - Operações de rede
  - Manipulação de arquivos
  - Recursos para desenvolvimento web
  - Ferramentas para criptografia e segurança.

## 2. EXERCÍCIO

- **ASP.NET** é um framework open-source e multiplataforma para o desenvolvimento de aplicações web modernas. Suas principais funções incluem:
  - Processamento de requisições HTTP e gerenciamento do ciclo de vida das páginas web;
  - Implementação do padrão MVC (Model-View-Controller);
  - Gerenciamento de rotas e endpoints da aplicação;
  - Sistema de middleware para processamento de requisições;
  - Injeção de dependência nativa;
  - Suporte a APIs RESTful;
  - Sistema de autenticação e autorização integrado.
- **Entity Framework Core (EF Core)** é um ORM (Object-Relational Mapper) para C# que visa facilitar a interação com bancos de dados. Suas principais funções incluem:
  - Mapeamento de classes C# para tabelas do banco de dados;
  - Geração e execução automática de consultas SQL;
  - Rastreamento de mudanças em objetos e sincronização com o banco de dados;
  - Gerenciamento de relacionamentos entre entidades;
  - Migrações de banco de dados;
  - Suporte a diversos provedores de banco de dados (SQL Server, PostgreSQL, SQLite, etc.).
- **Razor Pages** é um modelo de desenvolvimento dentro do ASP.NET Core voltado para a criação de páginas web dinâmicas.
  - Criação de páginas web usando sintaxe Razor (mistura de HTML e C#);

- Modelo baseado em páginas com arquivos .cshtml para markup e .cshtml.cs para lógica;
- Sistema de data binding entre formulários HTML e modelos C#;
- Helpers para geração de HTML dinâmico;
- Suporte a layouts e partial views para reutilização de código;
- Integração nativa com o sistema de roteamento do ASP.NET Core.

### 3. EXERCÍCIO

1)

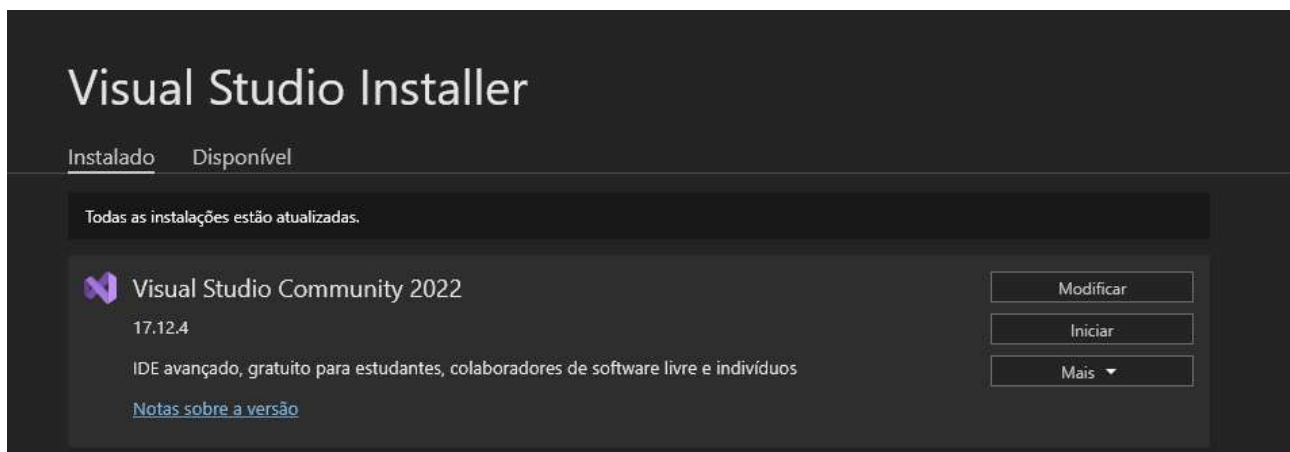
Um resumo das comparações em detalhes na questão abaixo seria:

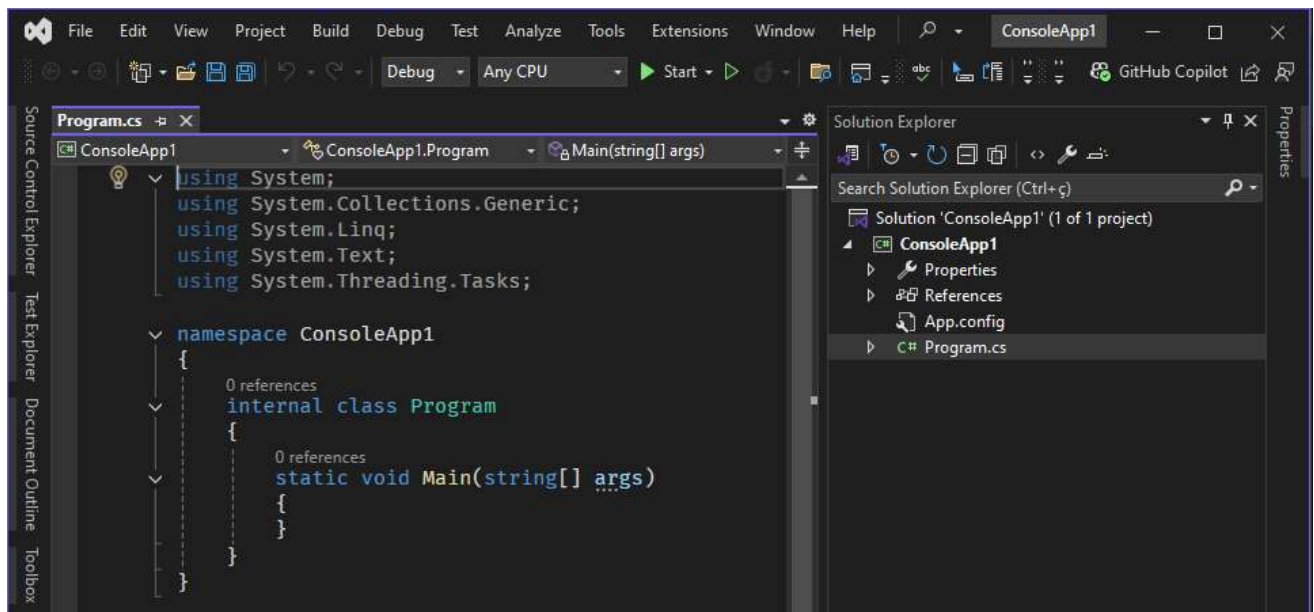
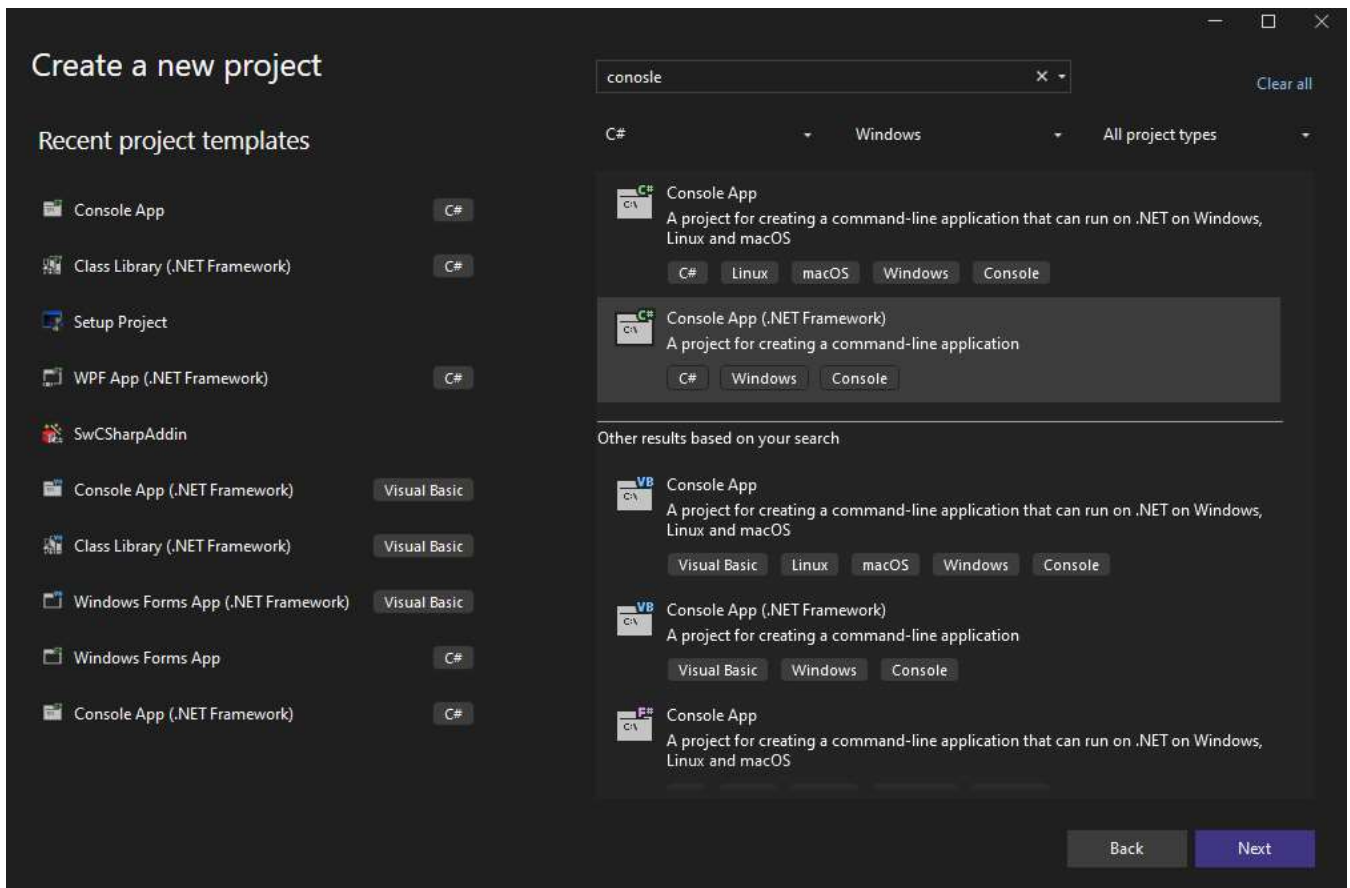
- Visual Studio, para quem deseja um ambiente completo para desenvolvimento C# e .NET;
- VS Code, para quando se quer um editor leve e flexível para vários tipos de projetos;
- JetBrains: Rider, se busca alto desempenho para projetos grandes e integração.

2)

Característica	Visual Studio	Visual Studio Code	Rider
<b>Suporte a múltiplas linguagens</b>	Excelente	Muito bom (com extensões)	Excelente
<b>Integração com ferramentas</b>	Completo (Azure, Git, SQL, Docker)	Extensível (Git, Docker, etc.)	Muito bom (JetBrains, Git)
<b>Facilidade de uso para iniciantes</b>	Médio (pode ser complexo)	Fácil (leve e intuitivo)	Médio (rico em recursos, mas tem curva de aprendizado)
<b>Desempenho</b>	Pesado	Leve e rápido	Médio
<b>Preço</b>	Gratuito (Community) / Pago (Pro, Enterprise)	Totalmente gratuito	Pago (trial disponível)

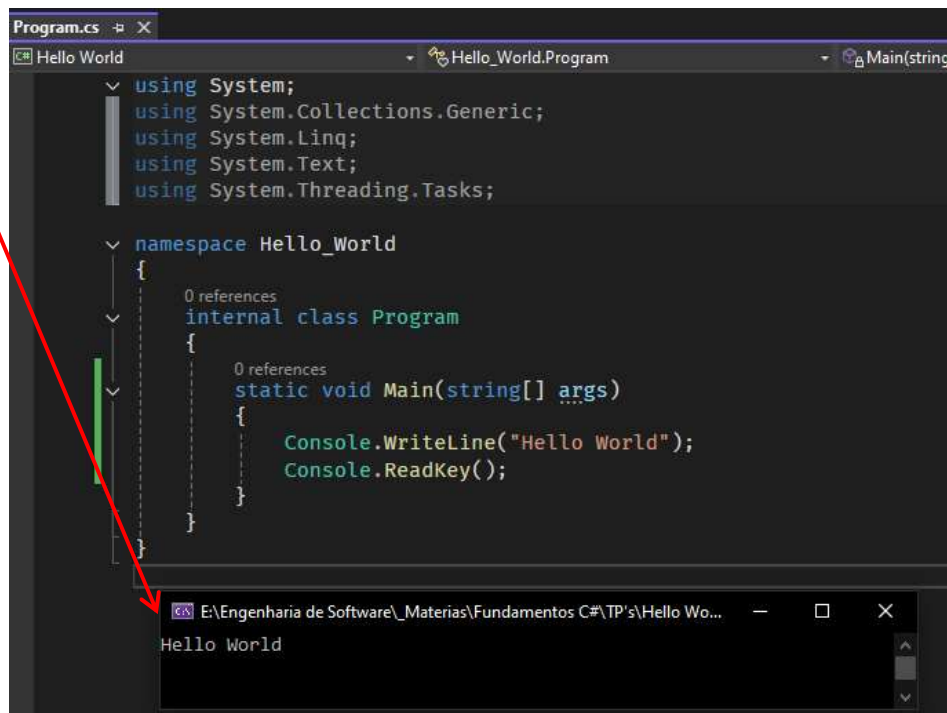
### 4. EXERCÍCIO





## 5. EXERCÍCIO

- Saída no terminal



The screenshot shows the Visual Studio IDE with the `Program.cs` file open. The code defines a `namespace Hello_World` containing an `internal class Program` with a `static void Main(string[] args)` method. Inside the `Main` method, `Console.WriteLine("Hello World");` and `Console.ReadKey();` are called. Below the code editor, the Output window is visible, showing the text `Hello World` as the result of the execution.

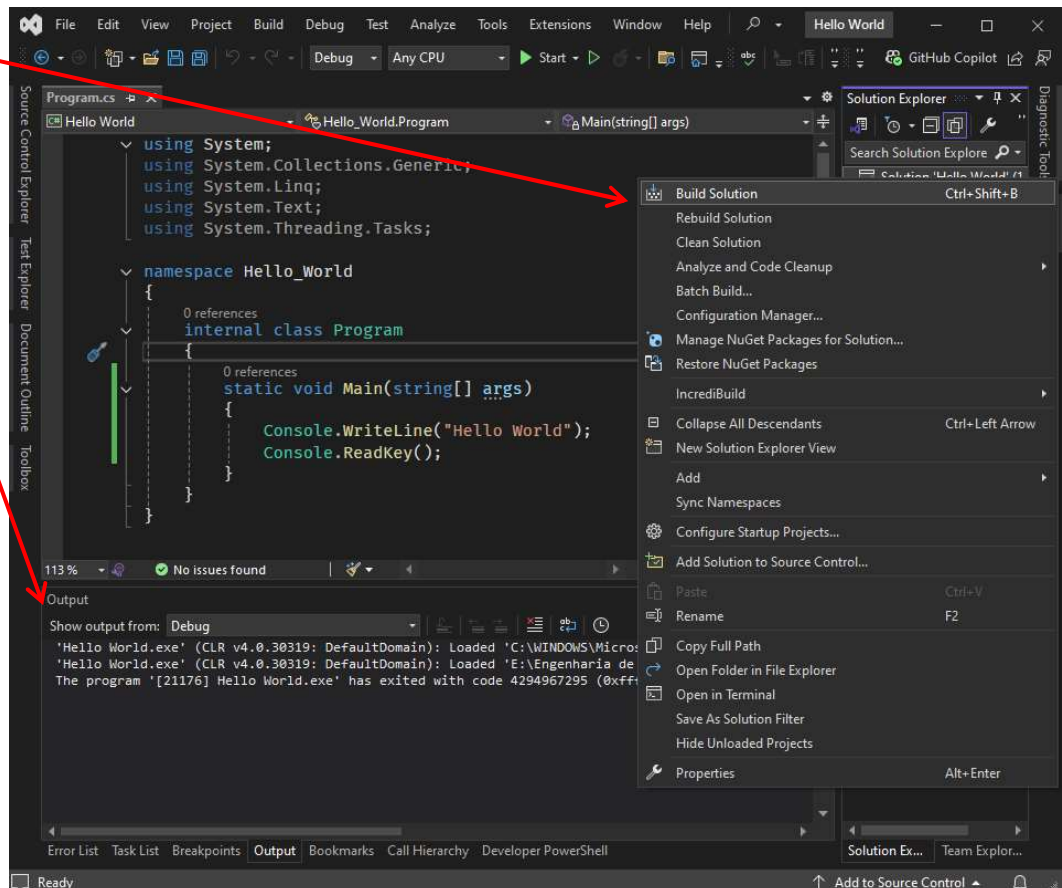
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Hello_World
{
    0 references
    internal class Program
    {
        0 references
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World");
            Console.ReadKey();
        }
    }
}
```

Hello World

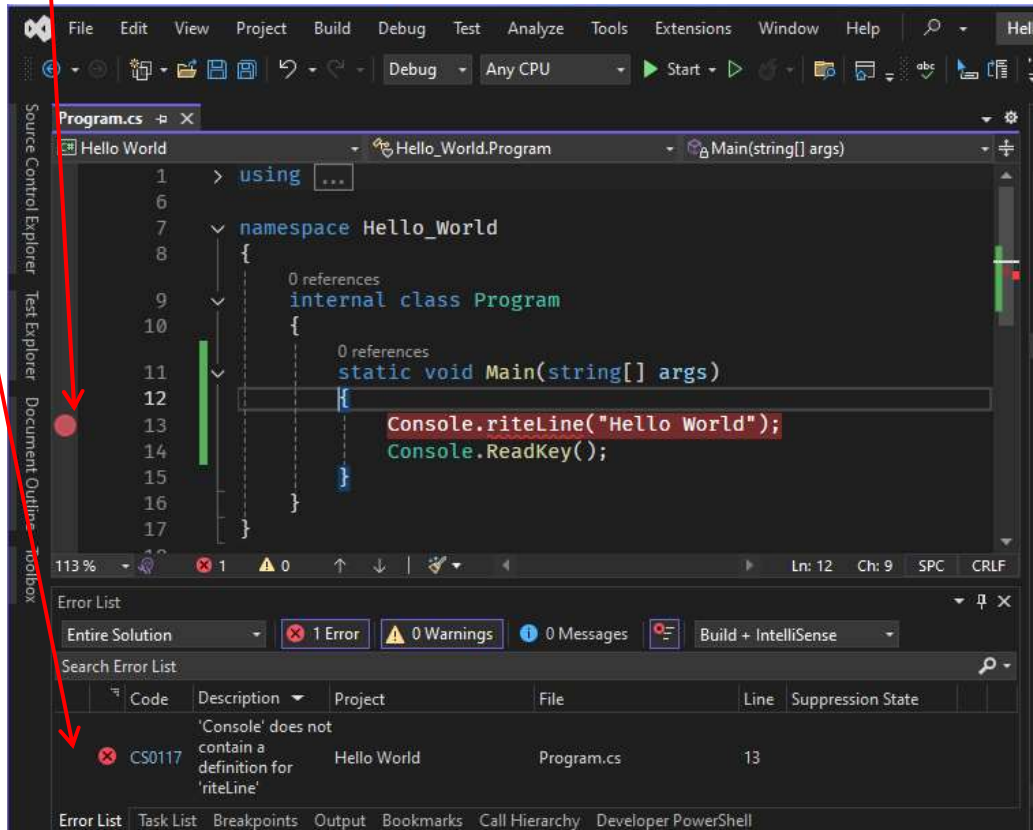
## 6. EXERCÍCIO

- Build Solution
- Output

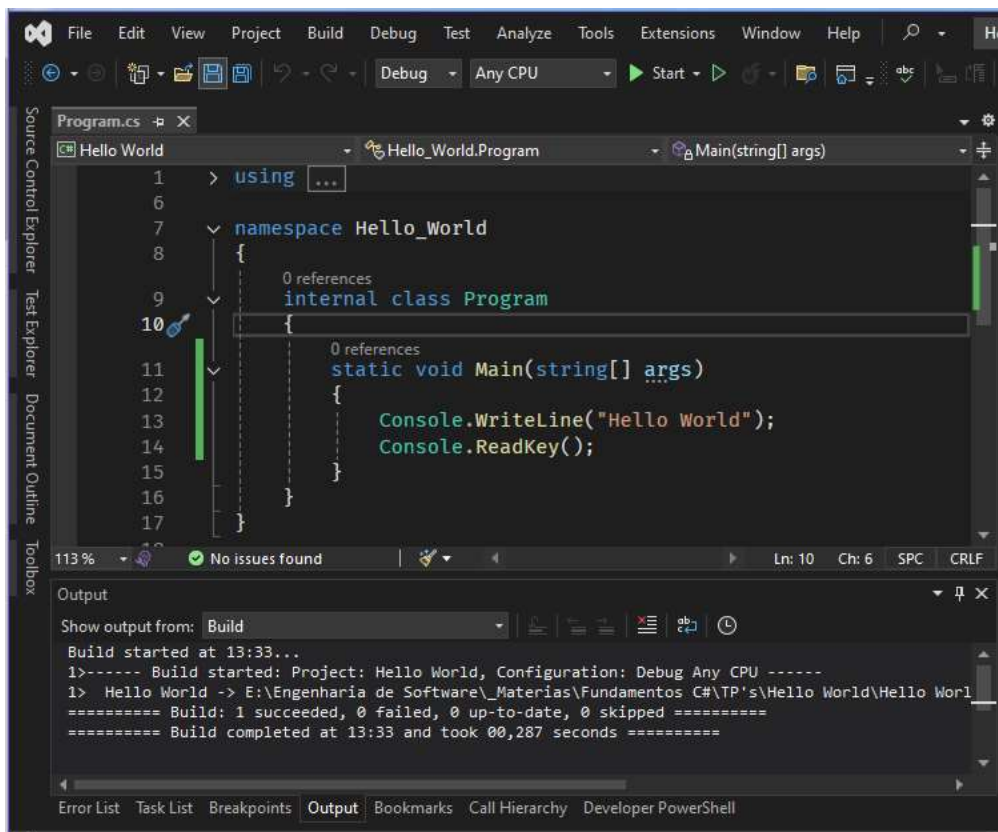


## 7. EXERCÍCIO

- Break Point
- Output com erro



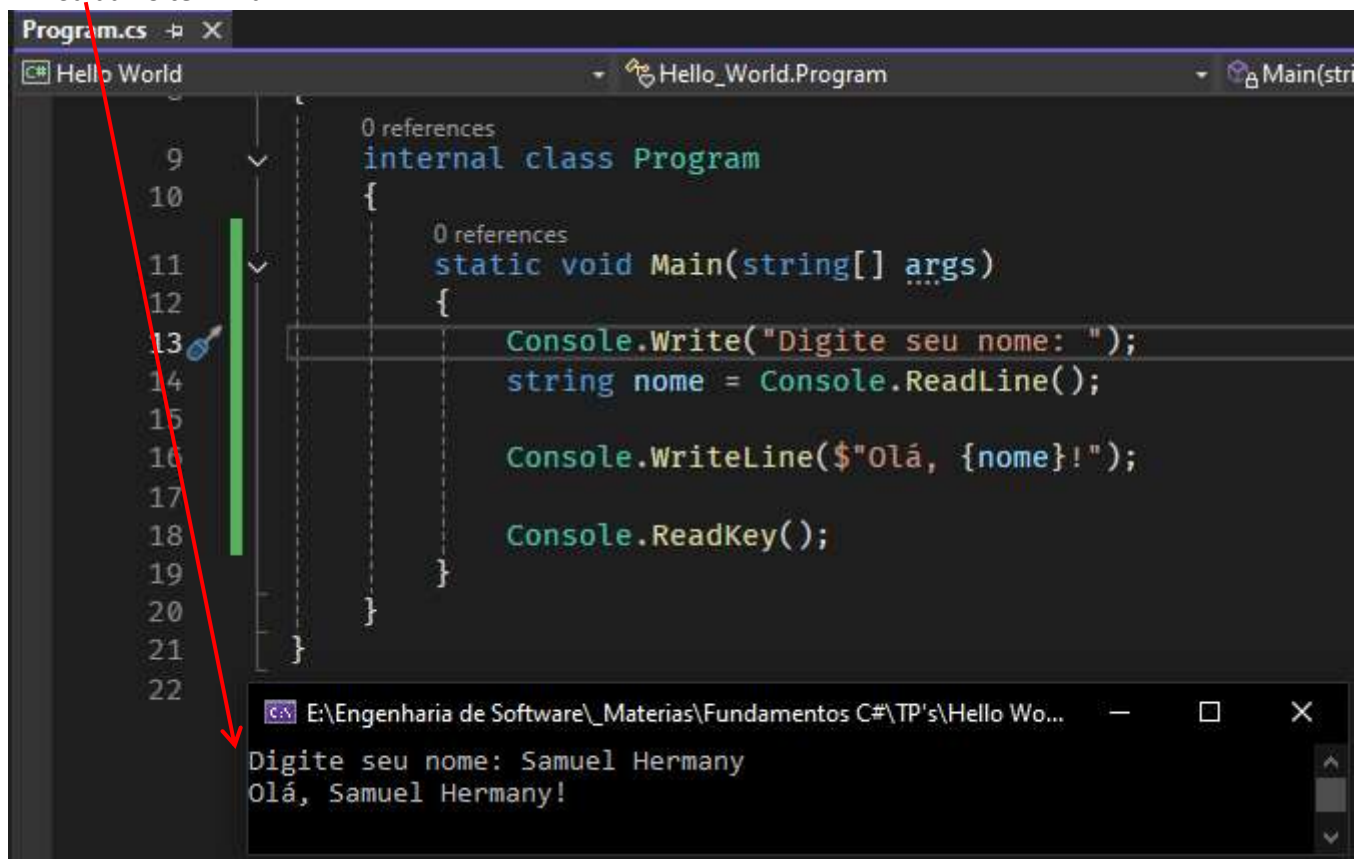
- Erro Resolvido





## 8. EXERCÍCIO

- Saída no terminal



```
Program.cs
C# Hello World
Hello_World.Program
Main(string[] args)

0 references
internal class Program
{
    0 references
    static void Main(string[] args)
    {
        Console.Write("Digite seu nome: ");
        string nome = Console.ReadLine();

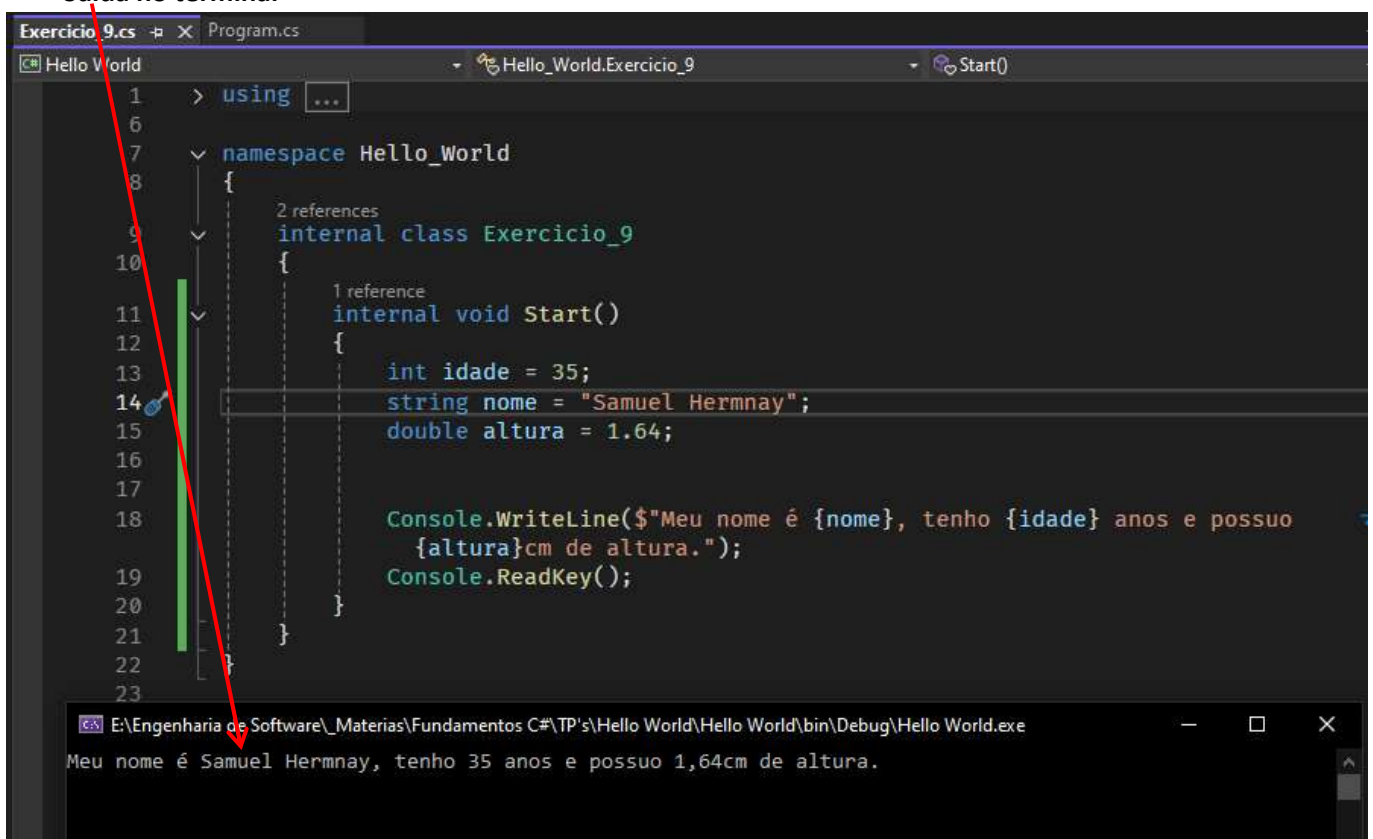
        Console.WriteLine($"Olá, {nome}!");

        Console.ReadKey();
    }
}
```

E:\Engenharia de Software\Materias\Fundamentos C#\TP's\Hello Wo...  
Digite seu nome: Samuel Hermnay  
Olá, Samuel Hermnay!

## 9. EXERCÍCIO

- Saída no terminal



```
Exercicio_9.cs
C# Hello World
Hello_World.Exercicio_9
Start()

1 > using ...
6
7 namespace Hello_World
8 {
9     2 references
10    internal class Exercicio_9
11    {
12        1 reference
13        internal void Start()
14        {
15            int idade = 35;
16            string nome = "Samuel Hermnay";
17            double altura = 1.64;
18
19            Console.WriteLine($"Meu nome é {nome}, tenho {idade} anos e possuo
20                {altura}cm de altura.");
21            Console.ReadKey();
22        }
23    }
}
```

E:\Engenharia de Software\Materias\Fundamentos C#\TP's\Hello World\Hello World\bin\Debug\Hello World.exe  
Meu nome é Samuel Hermnay, tenho 35 anos e possuo 1,64cm de altura.

## 10. EXERCÍCIO

- Saída no terminal

