



INSTITUTO INFNET

LÍVIA FARIA BRAZ

TESTE DE PERFORMANCE 1

Disciplina Regular 2: Fundamentos de
Desenvolvimento com C#

Professor: Luiz Paulo Maia

ITAÚNA - MG

12 de fevereiro de 2025



Fundamentos de Desenvolvimento com C# - TP1

➤ Exercício 1: Entender a Relação entre C# e .NET

○ O que é .NET?

O .NET é tipo uma plataforma gigantesca que a Microsoft criou pra rodar aplicativos. Ele funciona como um ambiente onde várias linguagens (como C#, F# e VB.NET) podem ser usadas pra desenvolver aplicações web, desktop, mobile e até jogos.

○ Como o C# se integra ao .NET?

O C# foi feito praticamente pra ser o par perfeito do .NET. Ele é uma das linguagens principais que roda dentro dessa plataforma e aproveita todas as ferramentas e bibliotecas disponíveis no .NET pra facilitar o desenvolvimento. Basicamente, se você escreve código em C#, o .NET cuida de tudo por trás pra fazer esse código funcionar direitinho.

○ O papel do CLR (Common Language Runtime) e do FCL (Framework Class Library):

- **CLR (Common Language Runtime):** Ele é tipo a “mágica” do .NET. Cuida da execução do código, gerenciando memória, segurança e até otimizando o desempenho do programa. É ele que transforma o código C# (ou de outras linguagens do .NET) em algo que o sistema consegue rodar.
- **FCL (Framework Class Library):** Essa é a biblioteca recheada de funcionalidades prontas, tipo classes, métodos e ferramentas que ajudam a gente a não precisar reinventar a roda. Quer manipular arquivos? Tem biblioteca pra isso. Trabalhar com banco de dados? Também tem!

➤ Exercício 2: Componentes Necessários para o Desenvolvimento Web com C#

○ ASP.NET Core

É o framework do .NET pra desenvolver aplicações web. Ele serve pra criar APIs, sites e até sistemas complexos, sendo bem rápido e modular.

○ Entity Framework Core

É um ORM (Object-Relational Mapper), ou seja, uma ferramenta que facilita a conexão do C# com o banco de dados. Em vez de escrever SQL puro, a gente usa comandos C# pra mexer nos dados, o que torna tudo mais simples.

○ Razor Pages

Um jeito mais direto de criar páginas web dinâmicas no ASP.NET Core. Ele mistura HTML com C#, permitindo gerar conteúdo de forma mais organizada e sem precisar de um monte de arquivos separados.



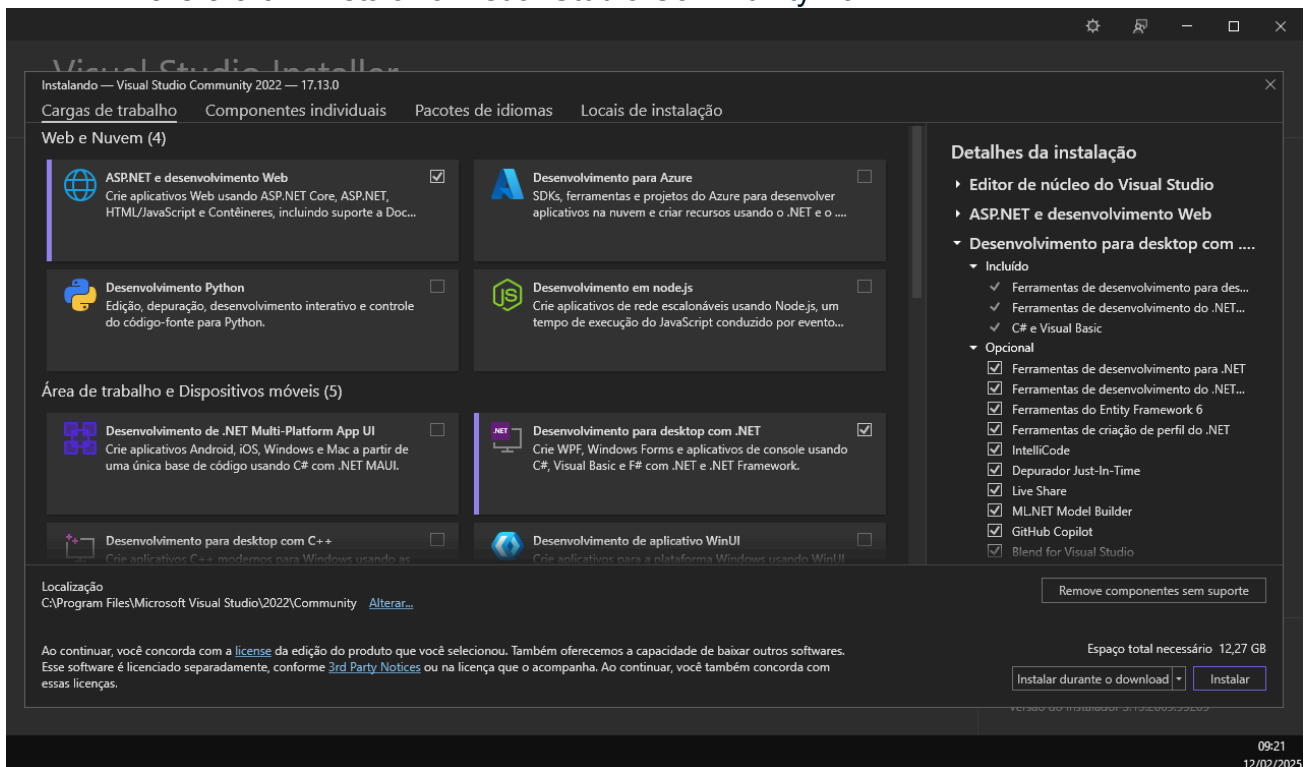
Fundamentos de Desenvolvimento com C# - TP1

➤ Exercício 3: Diferenças entre IDEs para Desenvolvimento em C#

Características	Visual Studio	Visual Studio Code	Rider
Suporte a múltiplas linguagens	Suporta várias linguagens, mas é focado em C# e .NET	Suporta várias linguagens, mas precisa de extensões	Suporta múltiplas linguagens, mas brilha no C#
Integração com ferramentas	Melhor integração com o ecossistema da Microsoft	Extensível, mas precisa configurar algumas coisas	Excelente integração com ferramentas JetBrains
Facilidade para iniciantes	Completo, mas pode ser pesado e complexo	Mais leve e simples, mas precisa configurar extensões	Interface intuitiva, mas pago

- **Visual Studio:** Melhor pra quem quer usar todos os recursos do .NET sem preocupação. Tem tudo integrado, mas pode ser meio pesado.
- **Visual Studio Code:** Mais leve, mas exige um pouco de configuração pra rodar bem projetos em C#. Ótimo pra quem gosta de customizar.
- **Rider:** Da JetBrains, tem um suporte incrível pra C# e várias ferramentas legais, mas é pago. Se você já gosta dos produtos da JetBrains, vai curtir.

➤ Exercício 04: Instalar o Visual Studio Community 2022





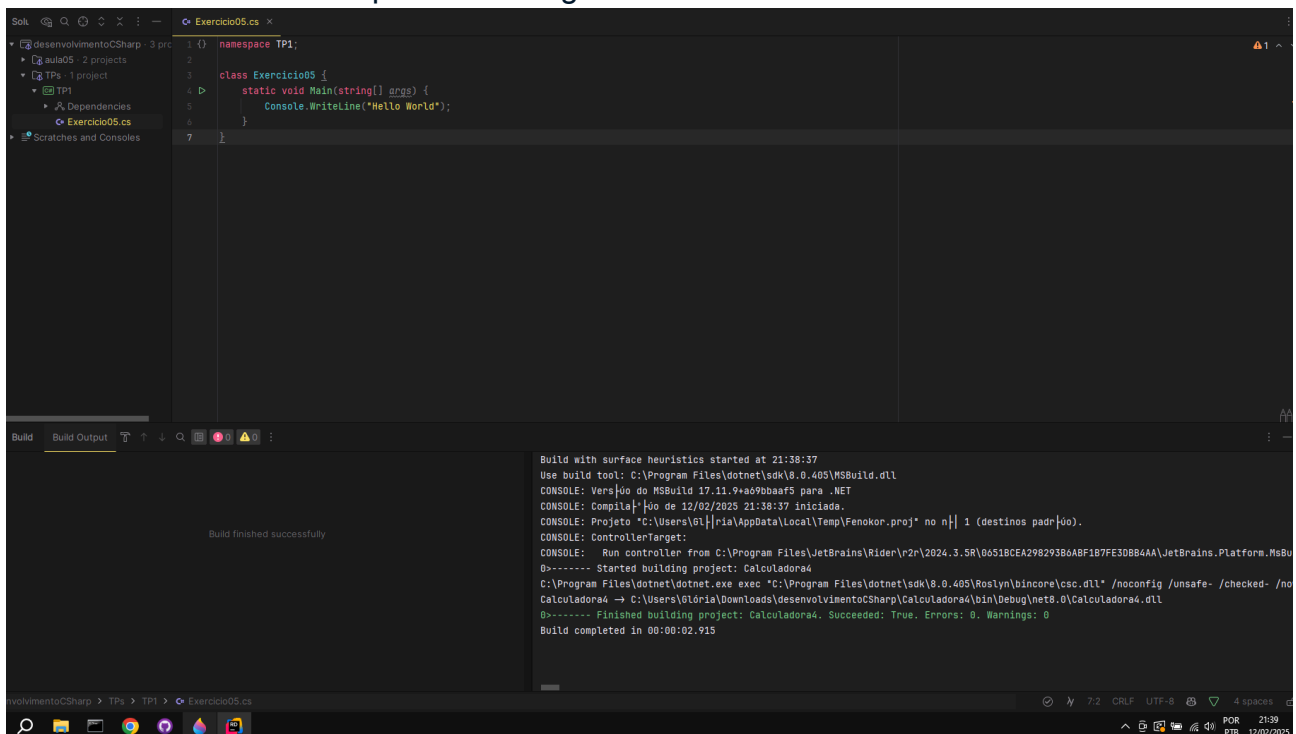
Fundamentos de Desenvolvimento com C# - TP1

➤ Exercício 05: Criar um Programa "Hello World"

```
namespace TP1;

class Exercicio05 {
    static void Main(string[] args) {
        Console.WriteLine("Hello World");
    }
}
```

➤ Exercício 06: Compilar um Código Usando Visual Studio



➤ Exercício 07: Depurar um Programa Simples

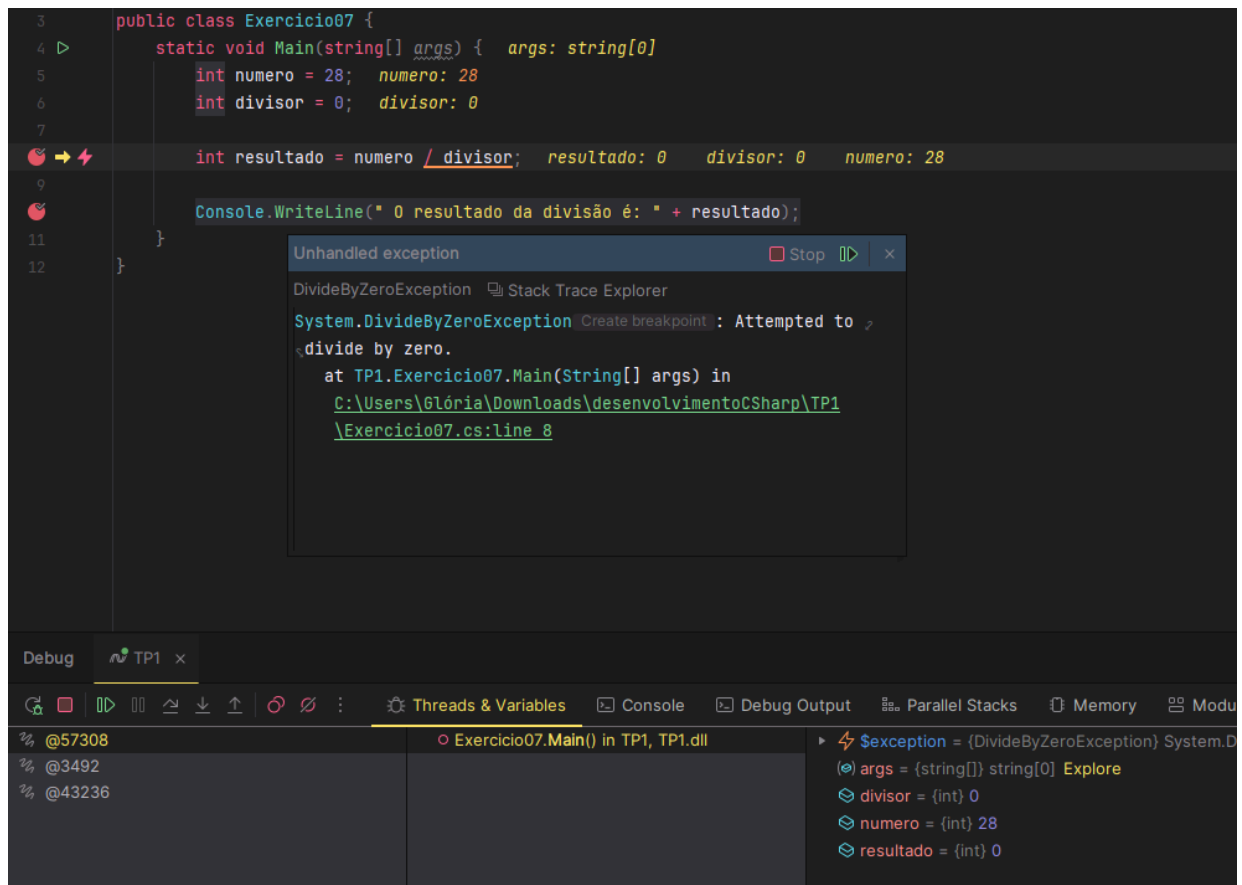
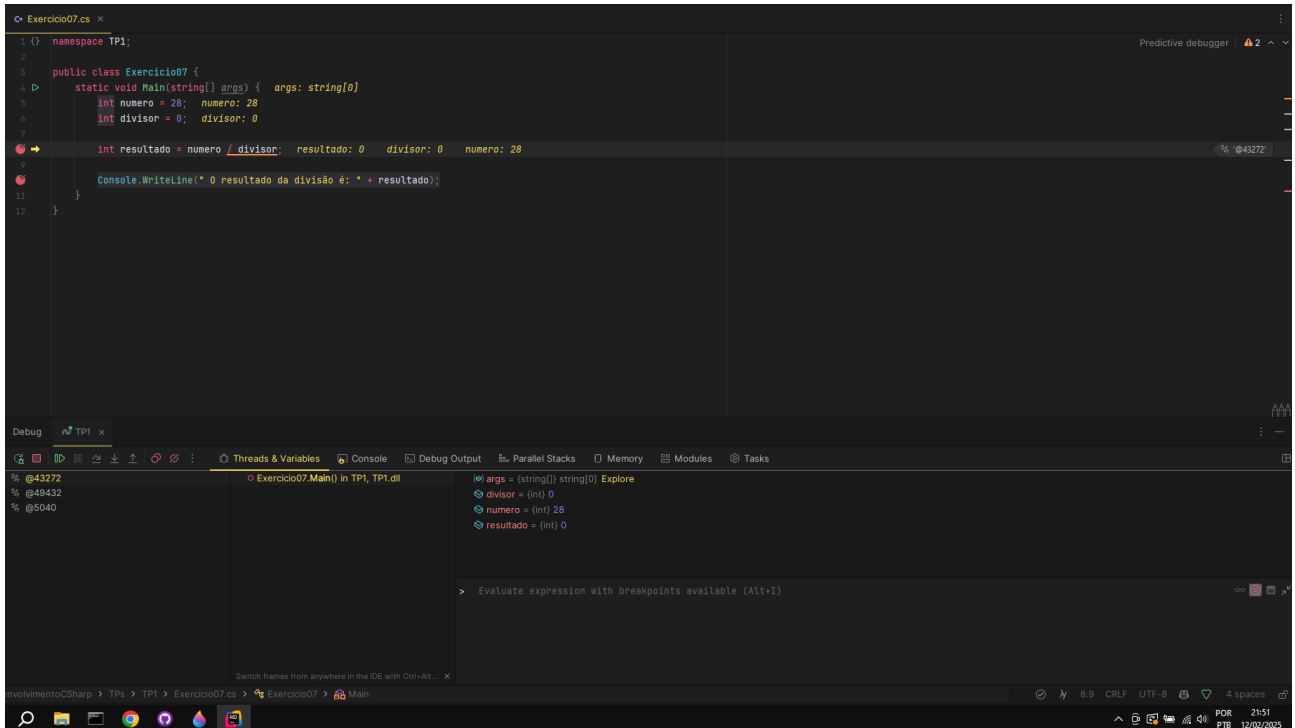
```
public class Exercicio07 {
    static void Main(string[] args) {
        int numero = 28;
        int divisor = 0;

        int resultado = numero / divisor;

        Console.WriteLine(" O resultado da divisão é: " + resultado);
    }
}
```



Fundamentos de Desenvolvimento com C# - TP1





Fundamentos de Desenvolvimento com C# - TP1

Código consertado:

```
public class Exercicio07 {  
    static void Main(string[] args) {  
        int numero = 28;  
        int divisor = 2;  
  
        int resultado = numero / divisor;  
  
        Console.WriteLine(" O resultado da divisão é: " + resultado);  
    }  
}
```

E ai fazendo novamente o debug, temos sucesso :]

The screenshot shows the Visual Studio IDE with the following components:

- Code Editor:** Displays the C# code for `Exercicio07`. The `Main` method calculates `resultado = 28 / 2` and prints it. The current line of execution is `Console.WriteLine(" O resultado da divisão é: " + resultado);`.
- Output Window:** Shows the output of the program: `O resultado da divisão é: 14`.
- Debug Console:** Shows the state of the program at the point of the `Console.WriteLine` statement. The variables are: `args = {string[]} string[0]`, `divisor = {int} 2`, `numero = {int} 28`, and `resultado = {int} 14`.



Fundamentos de Desenvolvimento com C# - TP1

➤ Exercício 08: Ler Entrada do Usuário

```
Solution ▾
└─ desenvolvimentoCSharp · 5 projects
  └─ aula05 · 2 projects
    └─ TP1 · 3 projects
      └─ Exercício08
        └─ Program.cs

1 {
2
3 namespace Exercício08;
4
5 class Program {
6     static void Main(string[] args) {
7         Console.WriteLine("Digite seu nome: ");
8         String nome = Console.ReadLine();
9
10        Console.WriteLine("Olá, " + nome + "!");
11    }
12 }
```

Run C# Interactive x Exercício08 x

Digite seu nome:
Glória
Olá, Glória!

Process finished with exit code 0.



Fundamentos de Desenvolvimento com C# - TP1

➤ Exercício 09: Uso de Variáveis e Tipos

The screenshot displays the Visual Studio IDE with the file `Exercicio09\Program.cs` open. The code defines a namespace `Exercicio09` and a `Program` class with a `Main` method. Inside `Main`, three variables are declared and assigned: `idade` (int, 22), `nome` (string, "Gloria"), and `altura` (double, 1.68). These values are then printed to the console using `Console.WriteLine`.

```
1 {} namespace Exercicio09;
2
3 class Program {
4     static void Main(string[] args) {
5
6         int idade = 22;
7         string nome = "Gloria";
8         double altura = 1.68;
9
10        Console.WriteLine("Nome: " + nome);
11        Console.WriteLine("Idade: " + idade);
12        Console.WriteLine("Altura: " + altura);
13    }
14 }
```

Below the code editor, the `Run` window shows the execution output for `Exercicio09`. The command line shows the path to the executable and the arguments. The output displays the values of the variables: `Nome: Gloria`, `Idade: 22`, and `Altura: 1,68`. The process finished with exit code 0.

```
C:\Users\Glória\AppData\Local\Programs\Rider\plugins\dpa\DotFiles\JetBrains.DPA.Runner.exe --handle=14492 -
-backend-pid=23348 --etw-collect-flags=67108622 --detach-event-name=dpa.detach.23348.15 --refresh-interval=
1 -- C:/Users/Glória/Downloads/desenvolvimentoCSharp/Exercicio09/bin/Debug/net8.0/Exercicio09.exe
Nome: Gloria
Idade: 22
Altura: 1,68
Process finished with exit code 0.
```

The status bar at the bottom indicates the file path `envolvimentoCSharp > TP1 > Exercicio09 > Program.cs`, the encoding `UTF-8`, and the line length `14:2`.



Fundamentos de Desenvolvimento com C# - TP1

➤ Exercício 10: Uso de Variáveis e Tipos Simples

```
1 {} namespace Exercicio10;
2
3 class Program {
4     static void Main(string[] args) {
5         string nome = "Glória";
6         int idade = 22;
7
8         Console.WriteLine($"Meu nome é {nome} e eu tenho {idade} anos.");
9     }
10 }
```

Run C# Interactive x Exercicio10 x

Meu nome é Glória e eu tenho 22 anos.

Process finished with exit code 0.

envolvimentoCSharp > TP1 > Exercicio10 > C# Program.cs 10:2 CRLF UTF-8 4 spaces

POR 22:15
PTB 12/02/2025