INSTITUTO INFNET

ESCOLA SUPERIOR DE TECNOLOGIA

GRADUAÇÃO EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

TP1

Aluno: Enzo Furtini

17 de mar. de 2025

Exercícios Práticos: C# e .NET

https://github.com/G-itch/Enzo_Furtini_DR2_TP1.git

Exercícios Práticos: C# e .NET

1. Relação entre C# e .NET

O .NET é uma plataforma de desenvolvimento criada pela Microsoft, usada para construir aplicações em diferentes ambientes (Windows, web, mobile). Uma de suas vantagens é a capacidade de integrar diversas linguagens, como C#, VB.NET e F#.

O C# foi desenvolvido especificamente para o .NET. Quando escrevemos código em C#, ele é compilado para uma linguagem intermediária (IL), que depois é executada pelo CLR (Common Language Runtime). Isso permite recursos como gerenciamento automático de memória e segurança de tipos.

Componentes-chave:
- CLR: Responsável por executar o código, convertendo IL para instruções nativas via compilação JIT.
- FCL (Framework Class Library): Oferece classes prontas para operações comuns como acesso a arquivos, redes e bancos de dados.

2. Componentes para Desenvolvimento Web

Para desenvolver aplicações web em C#, são essenciais:

- ASP.NET Core: Framework moderno para criar APIs e sistemas web. Suporta padrões como MVC e Razor Pages.
- Entity Framework Core: Simplifica o acesso a bancos de dados usando mapeamento objeto-relacional (ORM).
- Razor Pages: Permite misturar código C# com HTML para criar páginas dinâmicas de forma eficiente.

Exemplo de uso: Um e-commerce pode usar ASP.NET Core para o backend e Razor Pages para a interface administrativa.

3. Comparação de IDEs

Principais opções para desenvolvimento em C#:

Visual Studio:
- Prós: Ferramentas de depuração avançadas, suporte nativo ao .NET.
- Contras: Consumo alto de recursos, limitado a Windows/macOS.

Visual Studio Code:
- Prós: Leve, extensível com plugins como C# Dev Kit.
- Contras: Requer configuração manual para projetos complexos.

Rider (JetBrains):
- Prós: Multiplataforma, integração com Unity.
- Contras: Licença paga, curva de aprendizado para iniciantes.

5/6. Programa 'Hello World'

```
using System;

namespace PrimeiroProjeto {
    class Program {
        static void Main(string[] args) {
            // Imprime mensagem no console
            Console.WriteLine("Hello World!");
            Console.ReadKey();       // Aguarda tecla para
            fechar
        }
    }
}
```

8. Ler Entrada do Usuário

```
using System;

class EntradaUsuario {
    static void Main() {
        Console.Write("Digite sua cidade: ");
        string cidade = Console.ReadLine();
        Console.WriteLine($"Você mora em {cidade}!");
    }
}
```

9. Variáveis e Tipos de Dados

Em C#, é importante declarar o tipo das variáveis explicitamente

- int idade = 25; - Armazena números inteiros.

- double preco = 19.99; - Usado para valores decimais.

- string nome = "Ana"; - Guarda texto.

```
using System;

class Variaveis {
    static void Main() {
        int quantidade = 10;
        string produto = "Caneta";
        double peso = 0.15;
        Console.WriteLine($"{quantidade} {produto}s pesam {peso}kg no total.");
    }
}
```

10. Exercício 10

```
using System;

namespace ExercicioVariaveis
{
    class Program
    {
        static void Main(string[] args)
        {
            string nome = "Carlos";
            int idade = 25;

            Console.WriteLine($"Meu nome é {nome} e eu tenho {idade} anos.");
        }
    }
}
```