

Nome: Guilherme de Moraes China Costa

Título: Teste de Performance - Fundamentos de Desenvolvimento com C#

Data: 08/05/2025

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

O que é .NET?

O **.NET** é uma **plataforma de desenvolvimento open-source, moderna e multiplataforma** criada pela Microsoft. Ela permite desenvolver aplicações para diferentes ambientes, como:

- Web (com ASP.NET Core)
- Desktop (Windows Forms, WPF)
- Mobile (com .NET MAUI/Xamarin)
- APIs, jogos, IoT, nuvem, entre outros.

Desde o .NET 5, a Microsoft unificou .NET Core e .NET Framework em uma única plataforma chamada simplesmente de **.NET**.

Como o C# se integra ao .NET?

C# é uma das **linguagens oficiais** do .NET (assim como F# e VB.NET). Quando um programa C# é compilado, ele gera código intermediário chamado **IL (Intermediate Language)**. Esse código é executado pela **CLR (Common Language Runtime)**, o ambiente de execução do .NET.

Ou seja, o C# depende do .NET para ser compilado, executado, e para acessar suas bibliotecas e recursos.

Papel do CLR (Common Language Runtime)

O **CLR** é o **ambiente de execução do .NET**. Ele:

- Compila o código IL em **código de máquina** na hora da execução (JIT – Just-In-Time compilation)
 - Faz **gerenciamento de memória**, coleta de lixo (garbage collection)
 - Trata **segurança, exceções, e multithreading**
 - Permite que diferentes linguagens (como C# e F#) funcionem juntas no .NET
-

Papel do FCL (Framework Class Library)

A **FCL** (hoje mais comumente chamada **BCL – Base Class Library**) é um **conjunto de bibliotecas e classes padrão** do .NET, fornecendo funcionalidades prontas para:

- Manipular arquivos e strings
 - Acesso a banco de dados
 - Operações de rede
 - Serialização
 - Tarefas assíncronas e mais
-

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

Componente	Função
ASP.NET Core	Framework multiplataforma e de alto desempenho para construir aplicações web e APIs REST. Substitui o ASP.NET tradicional com mais performance e flexibilidade.
Entity Framework Core (EF Core)	ORM (Object-Relational Mapper) usado para interagir com bancos de dados relacionais em C#. Permite consultar e salvar dados no banco usando objetos e LINQ.
Razor Pages	Abordagem simplificada para criar páginas web dinâmicas com C# e HTML. Foca em organização por página (Page Model) e reduz a complexidade de MVC tradicional. Ideal para aplicações menores ou com menos separação de responsabilidades.

Exercício 3: Diferenças entre IDEs para Desenvolvimento em C#

Comparação entre Visual Studio, Visual Studio Code e JetBrains Rider:

IDE	Suporte a Múltiplas Linguagens	Integração com Ferramentas do .NET	Facilidade para Iniciantes	Vantagens	Desvantagens
Visual Studio	Suporte completo a C#, F#, VB.NET e outras	Excelente integração com .NET, NuGet, Azure, Debugger	Alta	Interface completa, excelente para projetos grandes, depuração avançada	Pesado, exclusivo do Windows (Community), consumo de memória alto
VS Code	Suporte a diversas linguagens via extensões	Integração boa com C# via extensões (C# e Debugger for .NET)	Média a alta	Leve, rápido, multiplataforma, ideal para aprender ou projetos pequenos	Menos recursos nativos para C# comparado ao Visual Studio
JetBrains Rider	Suporte a várias linguagens, especialmente C#, Kotlin, Java	Integração avançada com .NET, excelente para ASP.NET, EF	Média (mais técnico)	Multiplataforma, inteligente, rápido, refatorações poderosas, bom para projetos complexos	Pago (licença), curva de aprendizado inicial um pouco maior

Exercícios 5, 8, 9 e 10

```
1 TP1 x | 2 Program.cs x
3 class Program {
4     static void Print(string msg) { Console.WriteLine(msg); }
5     static void Exerc5() { Print("Hello world"); }
6
7     static void Exerc8() {
8         Print("Type your name: ");
9         string name = Console.ReadLine();
10        Print($"Hello, {name}!");
11    }
12
13    static void Exerc9() {
14        int age = 25;
15        string name = "Ana";
16        double height = 1.68;
17
18        Print("Name: " + name);
19        Print("Age: " + age);
20        Print("Height: " + height + "m");
21    }
22
23    static void Exerc10() {
24        string name = "Lucas";
25        int age = 22;
26
27        Print($"My name is {name} and I'm {age} years old.");
28    }
29
30    static void Separator() {
31        Print("-----");
32    }
33
34    static void Main() {
35        Separator();
36        Exerc5();
37        Separator();
38        Exerc8();
39        Separator();
40        Exerc9();
41        Separator();
42        Exerc10();
43        Separator();
44    }
45 }
```

NORMAL master Program.cs
"~/dev/infnet/FundamentosDesenvolvimentoC#/TP1/Program.cs" 42L, 822B written