Engenharia de Software

17/2/2025

Samuel Hermany

INSTITUTO INFNET

**TP1**

Fundamentos de Desenvolvimento com C#

Professor(a): Rinaldo Ferreira Júnior

# 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.

# 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.

# Exercício



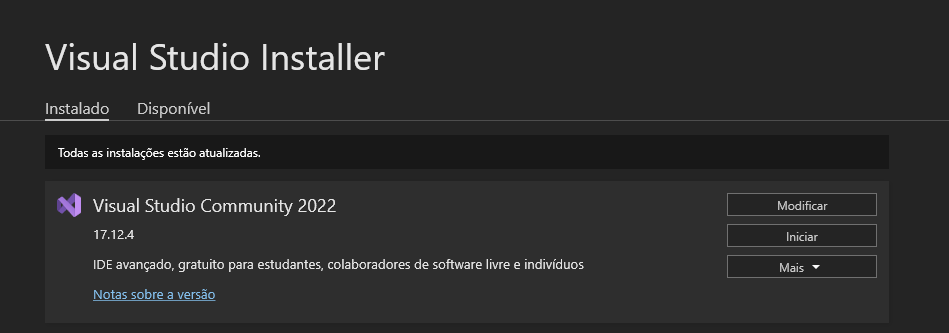
Um resumo das comparações em detalhes na questão abaixo seria:

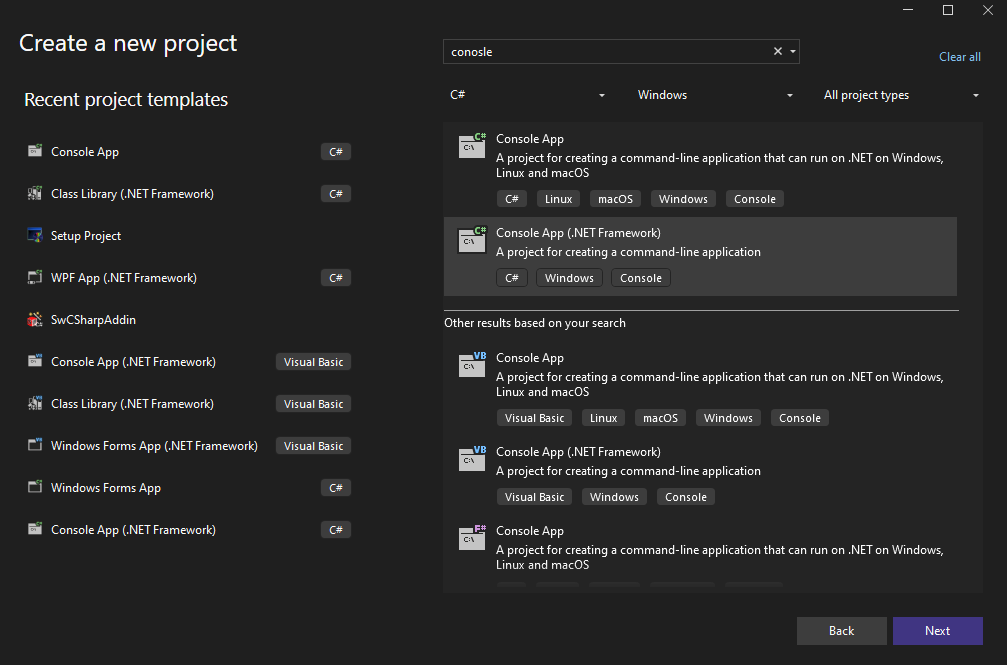
* Visual Studio, para quem deseja um ambiente completo para desenvolvimento C# e .NET;
* VS Code, para quando se quer um editor leve e flexível para vários tipos de projetos;
* JetBrains: Rider, se busca alto desempenho para projetos grandes e integração.

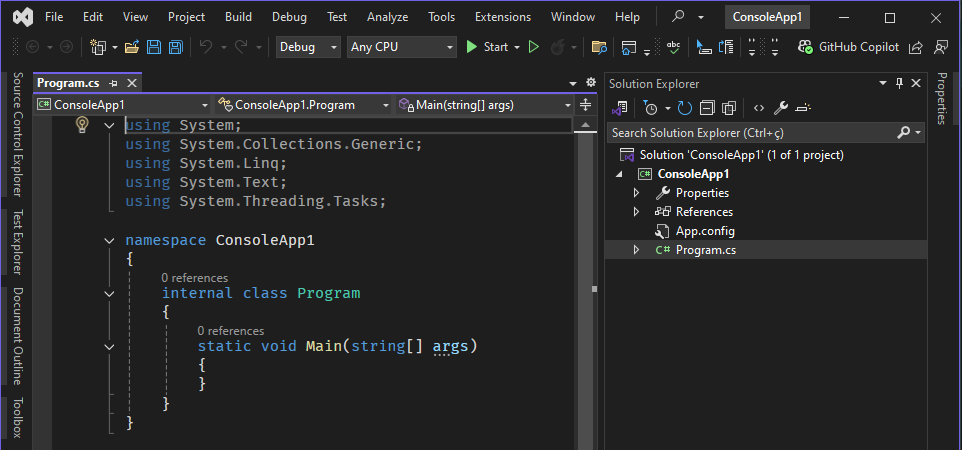


|  |  |  |  |
| --- | --- | --- | --- |
| **Característica** | **Visual Studio** | **Visual Studio Code** | **Rider** |
| **Suporte a múltiplas linguagens** | Excelente | Muito bom (com extensões) | Excelente |
| **Integração com ferramentas** | Completo (Azure, Git, SQL, Docker) | Extensível (Git, Docker,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

# Exercício

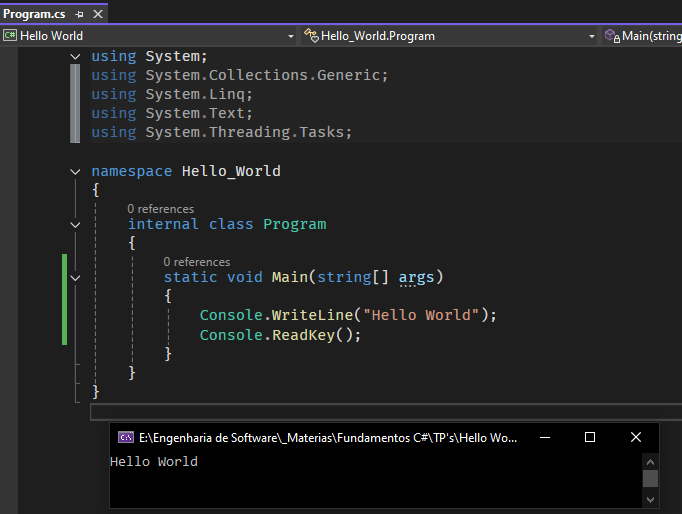






# Exercício

* **Saída no terminal**

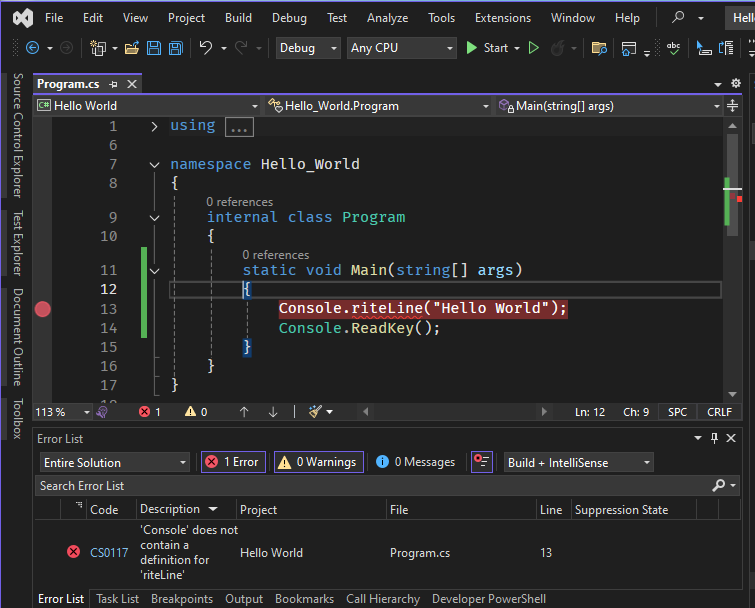


# Exercício

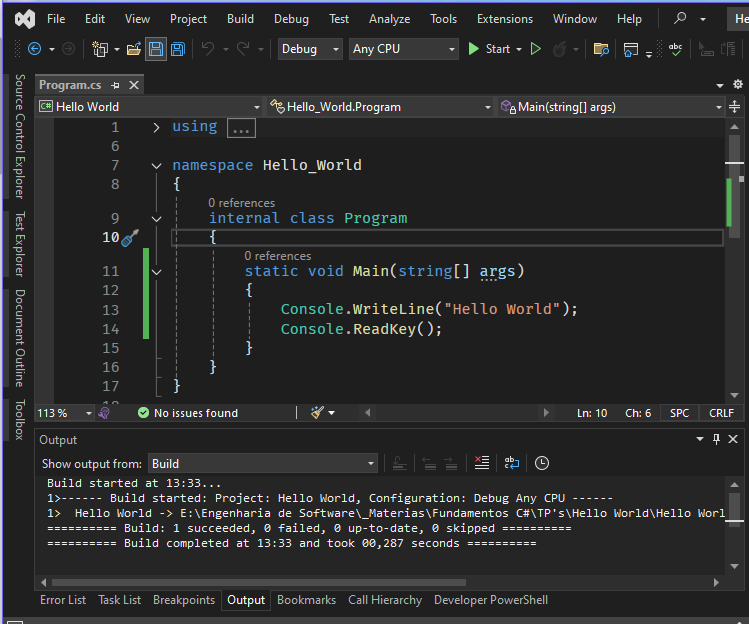
* **Build Solution**
* **Output**

# Exercício

* **Break Point**
* **Output com erro**

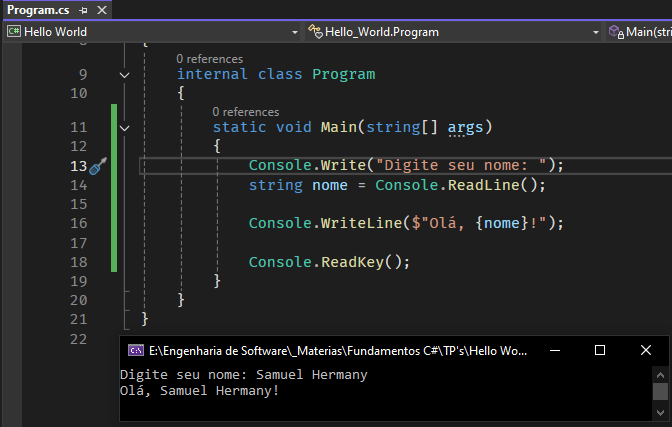


* **Erro Resolvido**



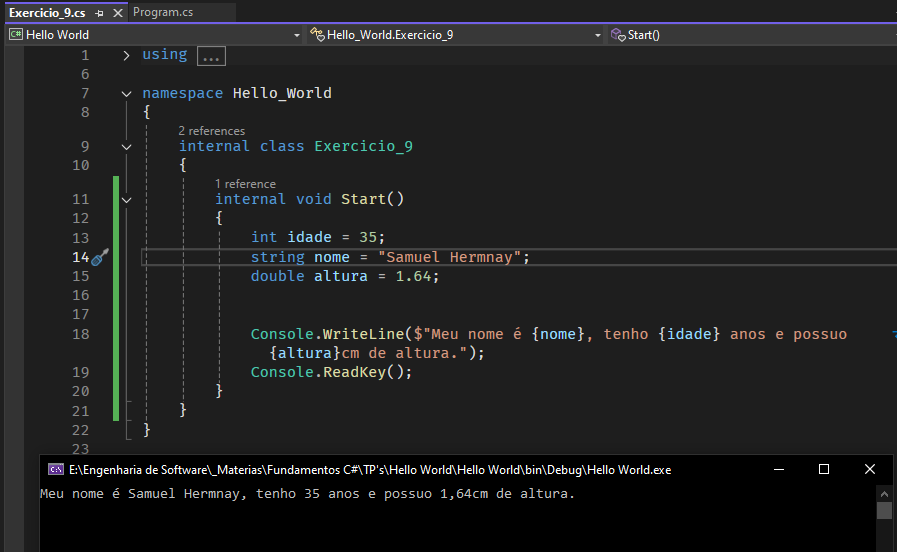
# Exercício

* **Saída no terminal**



# Exercício

* **Saída no terminal**



# Exercício

* **Saída no terminal**

