

# Fundamentos de Desenvolvimento com C#

*Professor: Orlando Fonseca*



**Kaike Torres da silva**

16.02.2025

Análise e Desenvolvimento de Sistemas (ADS)

## INTRODUÇÃO

Este relatório apresenta a exploração inicial do desenvolvimento com C# no ecossistema .NET, abordando conceitos fundamentais e práticas essenciais. Os exercícios incluíram a compreensão do .NET, a configuração do Rider como ambiente de desenvolvimento, a criação e execução de programas básicos, além do uso de variáveis, entrada do usuário e depuração de código. O objetivo foi estabelecer uma base sólida para aprofundar os estudos em C# e suas aplicações no desenvolvimento de software.

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

O que é .NET?

- .NET é um framework multiplataforma da Microsoft para desenvolvimento de aplicações (web, desktop, mobile).
- Composto por CLR (execução do código) e FCL (bibliotecas prontas).

Integração do C# com .NET

- C# é uma linguagem moderna e orientada a objetos, criada para o .NET.
- O código C# é compilado para CIL (linguagem intermediária), que o CLR converte em código nativo (via JIT) na execução.

Papel do CLR e FCL

1. CLR (Common Language Runtime):
  - Gerencia a execução do código (memória, segurança, exceções).
  - Compila CIL para código nativo (JIT) e faz coleta de lixo.
2. FCL (Framework Class Library):
  - Biblioteca com funcionalidades prontas (ex: arquivos, redes, interfaces).
  - Organizada em namespaces (ex: `System.IO`, `System.Net`).

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

1. **ASP.NET Core**
  - Framework para criar aplicações web, APIs e microserviços.
  - Multiplataforma (Windows, Linux, macOS).
2. **Entity Framework Core (EF Core)**
  - ORM para interagir com bancos de dados.
  - Facilita operações CRUD e migrações.
3. **Razor Pages**
  - Modelo para criar páginas web dinâmicas com HTML e C#.
  - Ideal para páginas simples e diretas.

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

### 1. Visual Studio

- **Vantagens:** Completo, melhor suporte para .NET, ferramentas avançadas.
- **Desvantagens:** Pesado, complexo para iniciantes, versão paga.

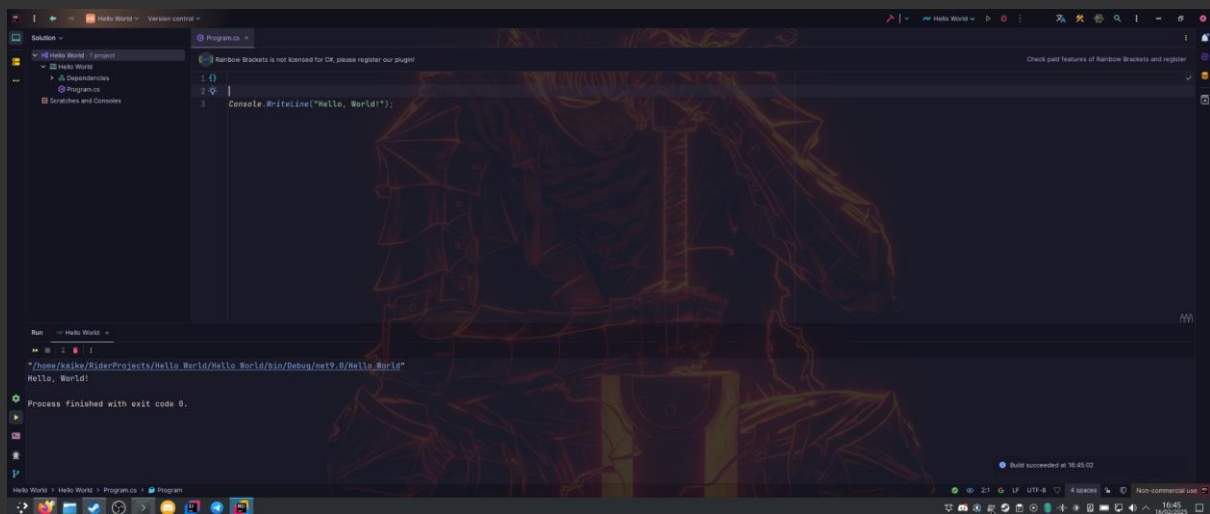
## 2. Visual Studio Code (VS Code)

- **Vantagens:** Leve, gratuito, multiplataforma, personalizável.
- **Desvantagens:** Requer configuração manual para C#, menos recursos nativos.

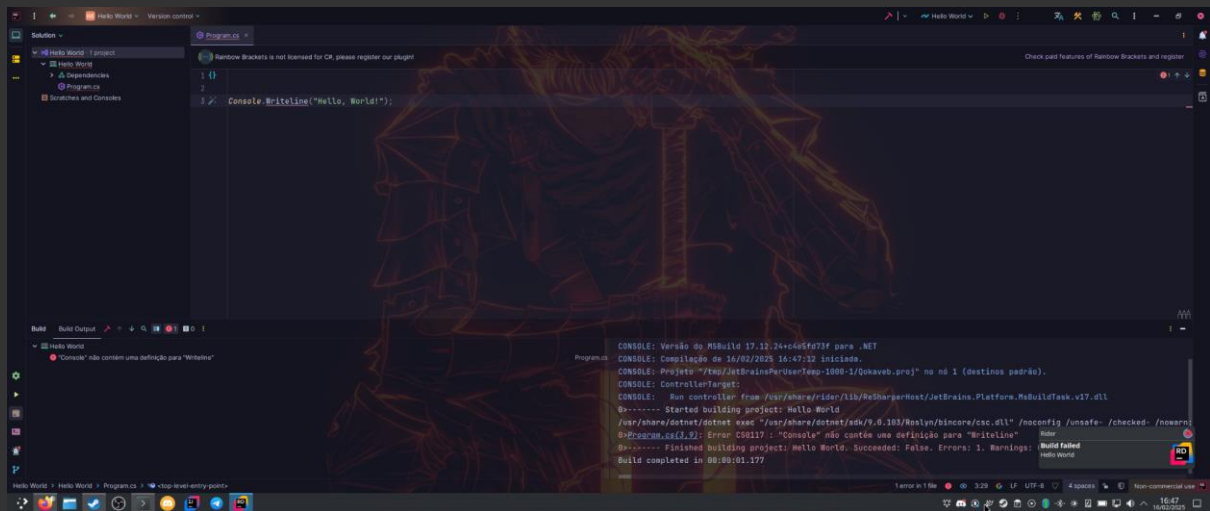
## 3. Rider

- **Vantagens:** Rápido, multiplataforma, boas ferramentas de análise.
- **Desvantagens:** Pago, menos popular.
- 

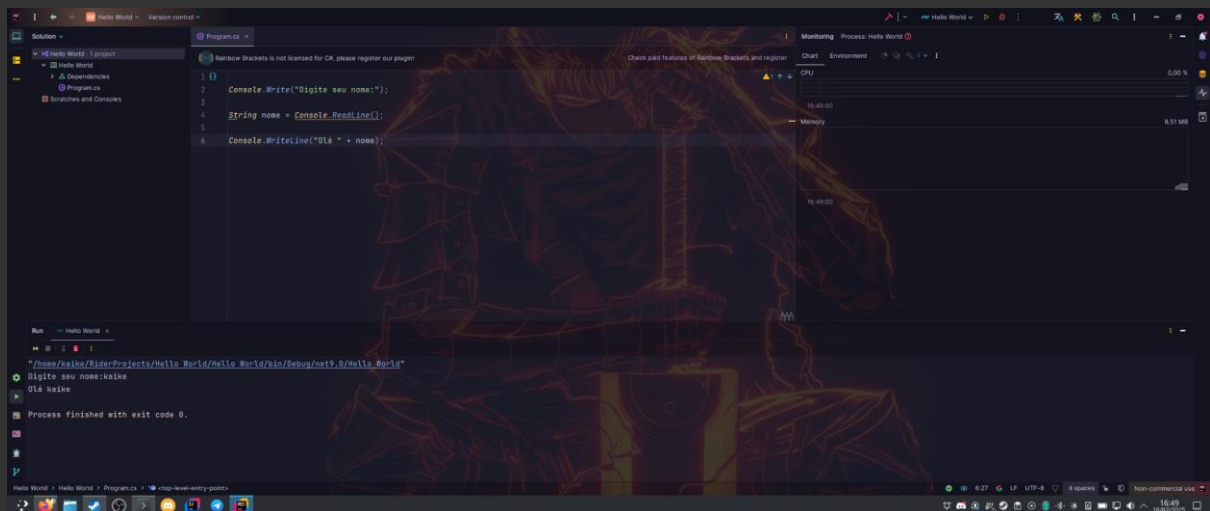
## Exercício 5: Criar um Programa "Hello World"ri



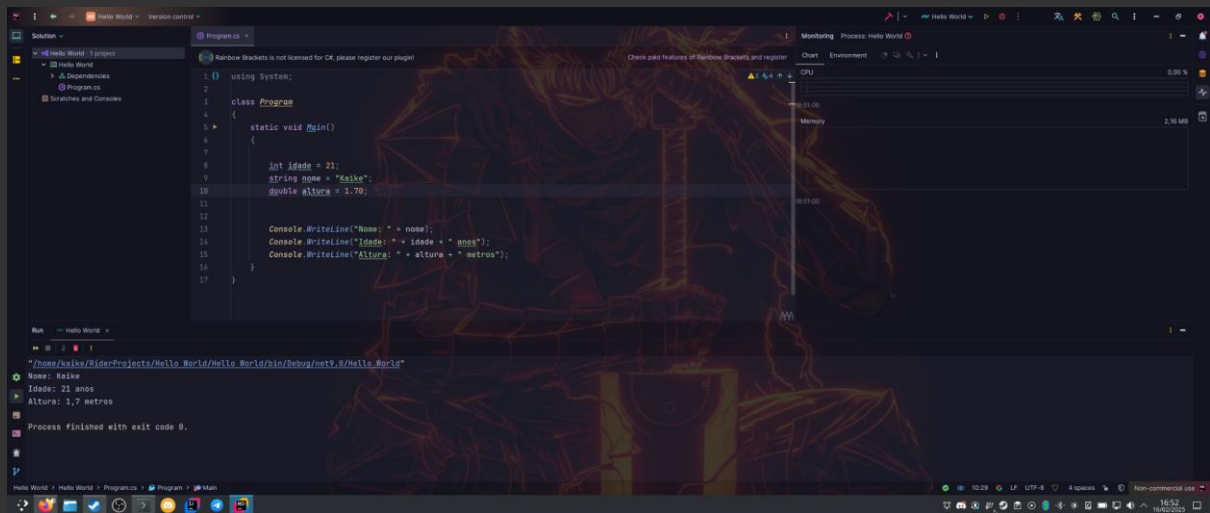
## Exercício 7: Depurar um Programa Simples



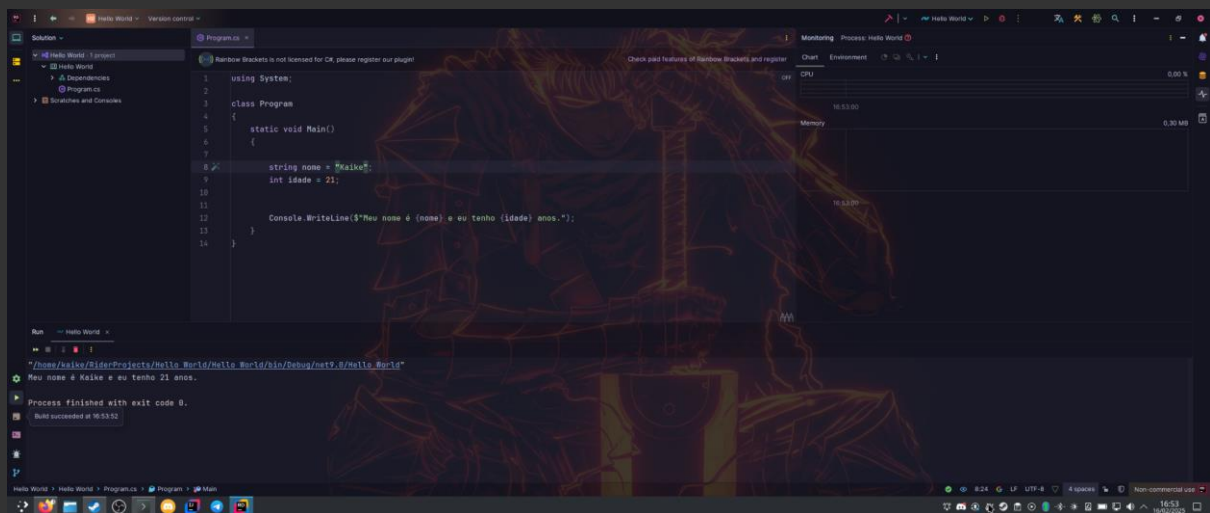
## Exercício 8: Ler Entrada do Usuário



## Exercício 9: Uso de Variáveis e Tipos



## Exercício 10: Uso de Variáveis e Tipos Simples



## Conclusão

Os exercícios proporcionaram uma base sólida sobre C# e o ecossistema .NET. Foram explorados conceitos fundamentais, como CLR e FCL, além de ferramentas

essenciais para desenvolvimento web. A prática incluiu a configuração do Rider, criação de programas simples, manipulação de entrada do usuário e uso de variáveis. Também foi possível entender a importância da depuração para correção de erros. Com isso, o aprendizado adquirido serve como ponto de partida para aprofundar os conhecimentos em C# e desenvolvimento de aplicações.