Fundamentos de Desenvolvimento com C#

Aula 05: Nulos

Professor: Rinaldo Ferreira Junior

E-mail: rinaldo.fjunior@prof.infnet.edu.br

- **Professor:** Rinaldo Ferreira Junior
- **Graduação:** Pós-graduado em Arquitetura de Softwares
- **Atuação:** .Net | C# | SQL | NoSQL | Engenheiro de Software
- **E-mail:** rinaldo.fjunior@prof.infnet.edu.br
- **Linkedin:** https://www.linkedin.com/in/rinaldo-ferreira-junior-787326a

Aula 05: Roteiro

- Nulos
- Operadores
- Tipos Anuláveis

Aula 05: Nulos

- No C#, null representa a ausência de valor. Podem ser usados quando se quer considerar um dado como ‘indefinido’.
- Pode ser atribuído à reference types e tipos nullable.
  - Não pode ser atribuído à value types não nullable
- Referências nulas podem causar diversos problemas durante a execução de um programa.
  - Tratar nulos corretamente, é importante para garantir um código robusto e resistente à erros de execução

Aula 05: Nulos

- A partir do C# 7.0 o operador is pode ser usado na checagem:

```
static void Main(string[] args)
{
    string semConteudo = null;

    Console.WriteLine($"ToString: {semConteudo}");
    Console.WriteLine($"Variável é nula?: {string.IsNullOrWhiteSpace(semConteudo)}");
    Console.WriteLine($"Variável é nula?: {semConteudo is null}");
    Console.WriteLine($"Variável é nula?: {semConteudo is not null}");

    if (semConteudo is null)
    {
        Console.WriteLine("Variável 'semConteudo' é nula");
    }
    else
    {
        Console.WriteLine("Variável 'semConteudo' não é nula");
    }

    Console.Read();
}
```

Aula 05: Operador de Coalescência

- O operador ?? avalia o valor, e retorna o dado da esquerda, caso o valor não seja nulo:

```
static void Main(string[] args)
{
    string semConteudo = null;
    Console.WriteLine($"Valor de semConteudo: {semConteudo ?? """Indefinido"""}");
    Console.WriteLine($@"Valor de semConteudo: {semConteudo ?? "Indefinido"}");

    semConteudo = "Eu existo!";
    Console.WriteLine($"Valor de semConteudo: {semConteudo ?? """Indefinido"""}");

    Console.Read();
}
```

Aula 05: Operador de Condicional Nula (?.)

- O operador ?. Permite acessar os membros do valor, caso o valor não seja nulo:

```
static void Main(string[] args)
{
    string semConteudo = null;
    Console.WriteLine($"Valor de semConteudo: {semConteudo ?? """Indefinido"""}");
    Console.WriteLine($@"Valor de semConteudo: {semConteudo ?? "Indefinido"}");

    semConteudo = "Eu existo!";
    Console.WriteLine($@"Valor de semConteudo: {semConteudo ?? "Indefinido"}");
    Console.WriteLine($"Caracteres: {semConteudo?.Length}");
    Console.WriteLine($"Maiúsculas: {semConteudo?.ToUpper()}");
    Console.WriteLine($"Trim: {semConteudo?.TrimEnd('!')}");

    Console.Read();
}
```

Aula 05: Tipos anuláveis (Nullable Types)

- Nullable possui propriedades read only para verificar se o valor é nulo, e obter seu valor caso não seja.

```
static void Main(string[] args)
{
    int? idade = 25;

    if (idade.HasValue)
    {
        Console.WriteLine($"A idade informada é: {idade.Value} anos");
    }
    else
    {
        Console.WriteLine($"A idade não foi informada");
    }

    Console.Read();
}
```

Aula 05: Tipos anuláveis (Nullable Types)

- Representam o value type original e um nulo adicional.
- Também pode representar um valor 'indefinido', mas agora, em um value type.
- Um tipo anulável pode ser criado usando-se também o caractere ?, porém, na declaração da variável. Ou como Nullable<T>.

```
static void Main(string[] args)
{
    int? idade = null;
    Console.WriteLine($"Idade foi informada? {idade is not null}");

    idade = 25;
    Console.WriteLine($"Idade foi informada? {idade is not null}");

    Nullable<double> salarioProfessor = 100d;
    if (salarioProfessor is double valorSalario)
    {
        Console.WriteLine($"O salário do professor é: {valorSalario:C}");
    }
    else
    {
        Console.WriteLine($"O salário do professor não foi informado");
    }

    Console.Read();
}
```

Aula 05: Exceção de Referência Nula

- A exceção NullReferenceException é disparada ao tentar manipular um nulo.

```
static void Main(string[] args)
{
    string? nome = "Tião Veneno ";

    try
    {
        nome = null;
        string message = $"O nome possui {nome.Length} caracteres";

        Console.WriteLine(message);
    }
    catch (NullReferenceException nEx)
    {
        Console.WriteLine("Nome não foi informado: " + nEx.Message);
    }
    catch (Exception ex)
    {
        Console.WriteLine("Erro inesperado");
    }

    Console.Read();
}
```