



Jeferson Oliveira

A blue rounded square with a thin green border, containing the white text '01' in a bold, sans-serif font.

01

Operadores Ternários

Operadores Ternários

A Arte da Decisão Compacta

Os operadores ternários são uma maneira concisa de tomar decisões em seu código. Aqui está um exemplo:

```
int vida = 100;  
string status = vida > 0 ? "Vivo" : "Morto";
```

A blue rounded square with a thin green border, containing the number 02 in a large, bold, dark blue font.

02

**LINQ (Language
Integrated Query)**

LINQ

Consultas Poderosas em C#

LINQ permite que você realize consultas complexas em coleções de dados. Veja este exemplo:

```
List<int> numeros = new List<int> {1, 2, 3, 4, 5};  
var pares = numeros.Where(n => n % 2 == 0);
```

A blue rounded square with a thin green border, containing the number 03 in a large, bold, dark blue font.

03

Método TryParse

TryParse

Conversão Segura de Tipos

O método TryParse tenta converter uma string em um tipo específico, retornando um booleano que indica se a conversão foi bem-sucedida.

```
string valor = "123";  
int numero;  
bool sucesso = Int32.TryParse(valor, out numero);
```

A blue rounded square with a thin green border, containing the number 04 in a large, bold, dark blue font.

04

Método
String.IsNullOrEmpty

String.IsNullOrEmpty

Validando Entradas do Usuário

O método

`String.IsNullOrEmpty` verifica se uma string é nula ou vazia, sendo uma maneira eficiente de validar entradas do usuário.

```
string nome = Console.ReadLine();
if (String.IsNullOrEmpty(nome))
{
    Console.WriteLine("Nome não pode ser vazio");
}
```



05

**Métodos Substring,
IndexOf**

Substring e IndexOf

Manipulando Strings com Facilidade

Os métodos Substring e IndexOf são usados para manipular strings. Veja como eles funcionam:

```
string frase = "O C# é uma linguagem poderosa";  
int posicao = frase.IndexOf("linguagem");  
string novaFrase = frase.Substring(posicao);
```

A blue rounded square with a thin green border, containing the number 06 in a dark blue, bold, sans-serif font.

06

Métodos Split e Join

Split e Join

Trabalhando com Arrays de Strings

Os métodos Split e Join são usados para dividir uma string em um array de strings e unir um array de strings em uma única string, respectivamente.

```
string texto = "C#;Python;JavaScript";  
string[] linguagens = texto.Split(';');  
Console.WriteLine(String.Join(", ", linguagens));
```