

C# E LINQ

A ALQUIMIA DOS DADOS



INTRODUÇÃO AO LINQ NO C#



O LINQ (Language Integrated Query) é uma das ferramentas mais poderosas do C# para manipular e consultar coleções de dados de forma clara e concisa. Ele oferece métodos para trabalhar com dados como se estivéssemos fazendo consultas em bancos de dados, mas diretamente em coleções na memória, como arrays ou listas.

Vamos explorar alguns dos principais métodos do LINQ, sempre com exemplos práticos.



01

WHERE

FILTRANDO COLEÇÕES

O método **Where** é utilizado para filtrar dados em uma coleção, baseado em uma condição.

```
C# e LINQ: A Alquimia dos Dados

List<int> numeros = new List<int> { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };
var numerosPares = numeros.Where(n => n % 2 == 0);
foreach (var numero in numerosPares)
{
    Console.WriteLine(numero); // Saída: 2, 4, 6, 8, 10
}
```



02

SELECT

TRANSFORMANDO DADOS

O **Select** permite projetar (ou transformar) cada item de uma coleção, alterando seu formato ou valor.

```
● ● ● C# e LINQ: A Alquimia dos Dados

List<string> nomes = new List<string> { "Ana", "Bruno", "Carlos" };
var nomesEmMaiusculas = nomes.Select(nome => nome.ToUpper());
foreach (var nome in nomesEmMaiusculas)
{
    Console.WriteLine(nome); // Saída: ANA, BRUNO, CARLOS
}
```



03

ORDERBY E ORDERBYDESCENDING

ORDENANDO COLEÇÕES

Os métodos **OrderBy** e **OrderByDescending** são usados para ordenar uma coleção em ordem crescente ou decrescente.

```
C# e LINQ: A Alquimia dos Dados

List<int> numeros = new List<int> { 5, 2, 8, 3, 1 };
var numerosOrdenados = numeros.OrderBy(n => n);
foreach (var numero in numerosOrdenados)
{
    Console.WriteLine(numero); // Saída: 1, 2, 3, 5, 8
}
```



04

FIRST E FIRSTORDEFAULT

OBTENDO O PRIMEIRO ELEMENTO

O método **First** retorna o primeiro elemento de uma coleção que satisfaz uma condição. Se não houver, ele lança uma exceção. Para evitar exceções, usamos **FirstOrDefault**.

```
C# e LINQ: A Alquimia dos Dados

List<int> numeros = new List<int> { 10, 20, 30 };

var primeiroMaiorQue15 = numeros.First(n => n > 15);
Console.WriteLine(primeiroMaiorQue15); // Saída: 20

var nenhumNumero = numeros.FirstOrDefault(n => n > 50);
Console.WriteLine(nenhumNumero); // Saída: 0 (valor padrão de int)
```



05

GROUPBY

AGRUPANDO DADOS

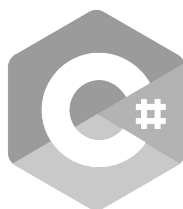
O **GroupBy** permite agrupar elementos de uma coleção com base em uma chave comum.

```
C# e LINQ: A Alquimia dos Dados

var alunos = new List<Aluno>
{
    new Aluno { Nome = "Ana", Nota = 9 },
    new Aluno { Nome = "Bruno", Nota = 7 },
    new Aluno { Nome = "Carlos", Nota = 9 }
};

var gruposPorNota = alunos.GroupBy(a => a.Nota);

foreach (var grupo in gruposPorNota)
{
    Console.WriteLine($"Nota: {grupo.Key}");
    foreach (var aluno in grupo)
    {
        Console.WriteLine(aluno.Nome); // Agrupados por nota
    }
}
```



06

ANY



VERIFICANDO SE EXISTE UM ELEMENTO



O **Any** verifica se existe ao menos um elemento em uma coleção que atenda a uma condição.



C# e LINQ: A Alquimia dos Dados

```
List<int> numeros = new List<int> { 1, 2, 3 };  
  
bool existePar = numeros.Any(n => n % 2 == 0);  
Console.WriteLine(existePar); // Saída: True
```



07

ALL



OPERAÇÕES NUMÉRICAS

Diferente do Any, o **All** verifica se todos os elementos atendem a uma condição.

```
● ● ● C# e LINQ: A Alquimia dos Dados  
  
List<int> numeros = new List<int> { 2, 4, 6 };  
  
bool todosPares = numeros.All(n => n % 2 == 0);  
Console.WriteLine(todosPares); // Saída: True
```



08

SUM, MAX, MIN, AVERAGE

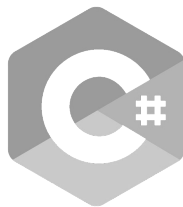
FILTRANDO COLEÇÕES

Esses métodos realizam operações como somar, obter o maior ou menor valor, ou calcular a média dos elementos.

```
C# e LINQ: A Alquimia dos Dados

List<int> numeros = new List<int> { 1, 2, 3, 4, 5 };

int soma = numeros.Sum();           // Saída: 15
int maior = numeros.Max();          // Saída: 5
int menor = numeros.Min();          // Saída: 1
double media = numeros.Average();   // Saída: 3
```



Conclusão



O **LINQ** facilita muito a manipulação de coleções no C#, oferecendo uma maneira elegante de consultar e transformar dados. Explore esses métodos e incorpore-os no seu dia a dia para escrever códigos mais limpos e eficientes!



AGRADECIMENTOS



OBRIGADA POR LER ATÉ AQUI



Este Ebook foi gerado com o auxílio de IA, toda a diagramação foi realizada por um humano.



<https://github.com/katiealeuterio/prompts-for-ebook-generate-by-ia>

