TECNOLOGIA EM DESENVOLVIMENTO DE SISTEMAS

LINQ (Consulta Integrada à Linguagem)

Barbara Rodrigues

VISÃO GERAL

O QUE É LINQ?

O LINQ é um conjunto de tecnologias com base na integração e recursos de consulta diretamente na linguagem C#.

COMO É ESCRITO

O LINQ é escrito através de uma sintaxe de consulta, utilizando operadores para filtragem, ordenação e agrupamento em fontes de dados com o minimo de código possível.

UNIVERSALIDADE

Ao invés de ter que aprender uma linguagem para cada tipo de banco de dados o LINQ possibilita a escrita do mesmo código de consulta para: objetos (LINQ to Objects) e banco de dados relacionais SQL (LINQ to SQL) e XML (LINQ to XML) .

```
int[] scores = new int[] { 81, 92, 97, 60 };

IEnumerable<int> scoreQuery =
    from score in scores
    where score > 80
    select score;

foreach (int i in scoreQuery)
{
    Console.Write(i + " ");
}
```

Consultas com o LINQ

O QUE É UMA CONSULTA LINQ?

- Uma **consulta** é uma expressão que **recupera** dados de uma fonte de dados.
- Possui modelo consistente para trabalhar com dados em vários tipos de fontes de dados e formatos.
- Trabalho com objetos.

AS TRÊS PARTES DA OPERAÇÃO DE CONSULTA

- Obter a fonte de dados
- Criar a consulta
- Executar a consulta

PONTOS IMPORTANTES

- A execução da consulta é diferente da propria consulta. Ao se criar uma variavel de consulta não se recupera nenhum dado, até o momento da execução dessa variável.

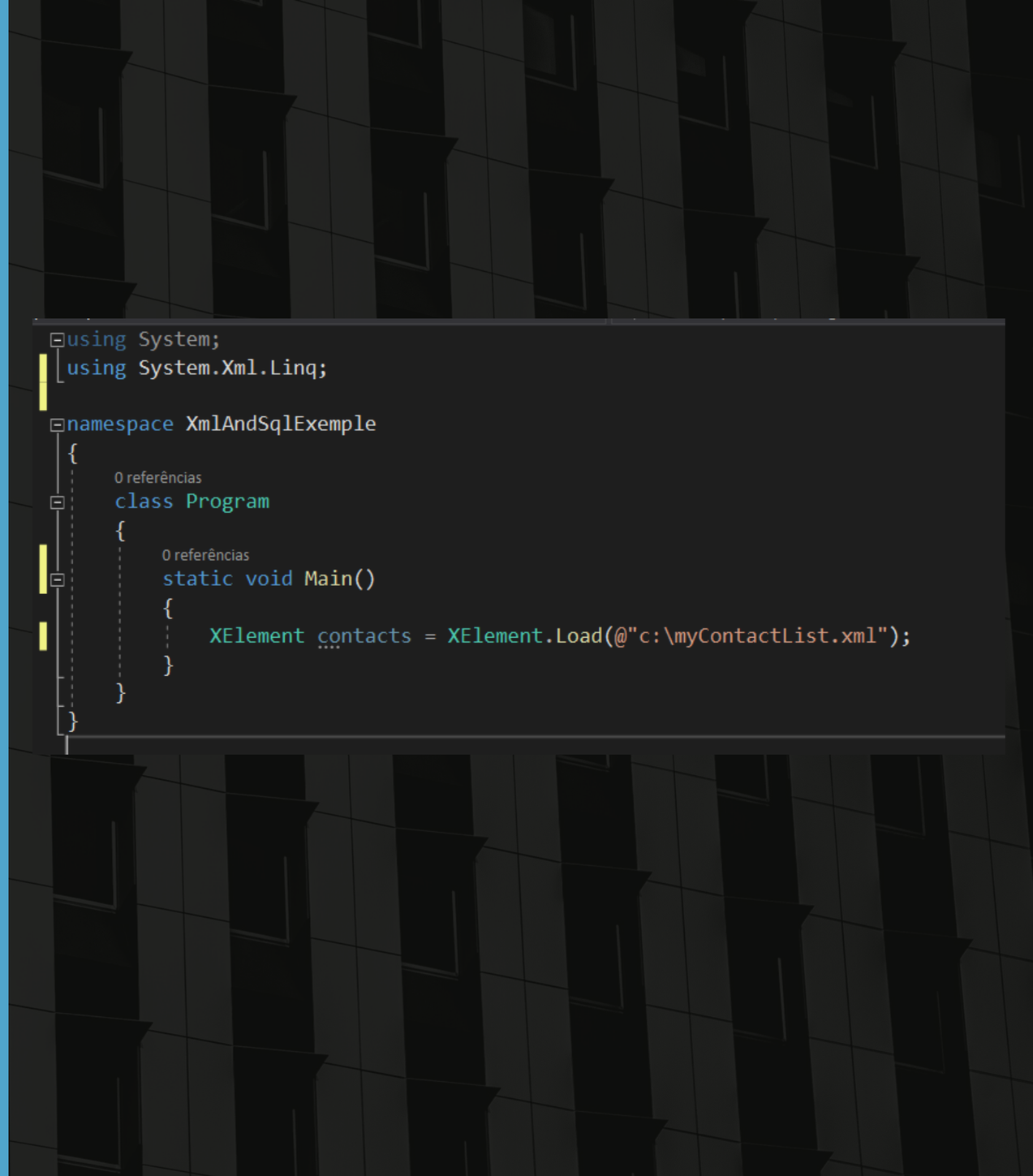
A Fonte de dados

- A IEnumerable<T> é a interface que permite que as classes de coleção genérica sejam enumeradas usando a instrução foreach.

- As vezes não se tem a declaração explicita desse tipo de interface , pois algumas fontes de dados são passiveis de consultas e dão o suporte a interface genérica IEnumerable<T> de forma implícita.

- Um tipo passível de consulta não requer nenhuma modificação e se os dados de origem não estiverem na memória como um tipo passível de consulta, o provedor LINQ se encarregará de fazer tal modificação.

- De forma geral, uma fonte de dados LINQ é qualquer objeto que de suporte a IEnumerable<T> interface genérica ou uma interface que herde dela.



```
using System;
using System.Xml.Linq;

namespace XmlAndSqlExemple
{
    0 referências
    class Program
    {
        0 referências
        static void Main()
        {
            XElement contacts = XElement.Load(@"c:\myContactList.xml");
        }
    }
}
```

A consulta

- A consulta especifica quais informações devem ser recuperadas da fonte (ou fontes) de dados.
- Pode-se também ser especificado como essas informações serão classificadas, agrupadas e moldadas antes de ser retornadas.
- Para fazer essas especificações, a expressão de consulta utiliza cláusulas de consulta.

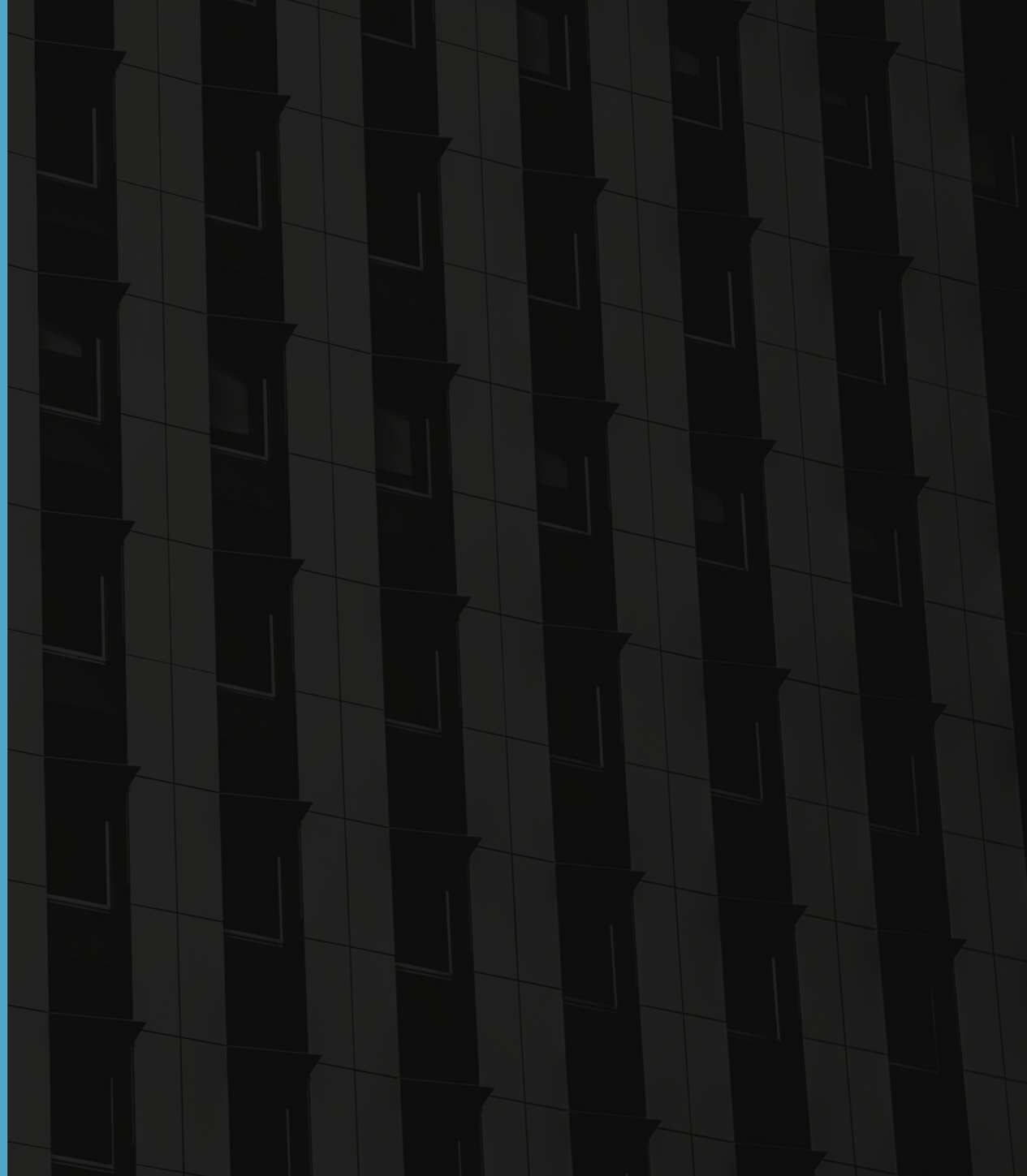
Cláusula	Descrição
from	Especifica uma fonte de dados e uma variável de intervalo (semelhante a uma variável de iteração).
where	Filtra elementos de origem baseados em uma ou mais expressões boolianas separadas por operadores AND e OR lógicos (`&&` ou `||`).
select	Especifica o tipo e a forma que os elementos na sequência retornada terão quando a consulta for executada.
grupo	Agrupa os resultados da consulta de acordo com um valor de chave especificado.
into	Fornece um identificador que pode funcionar como uma referência aos resultados de uma cláusula join, group ou select.
OrderBy	Classifica os resultados da consulta em ordem crescente ou decrescente com base no comparador padrão para o tipo de elemento.
join	Une duas fontes de dados com base em uma comparação de igualdade entre dois critérios de correspondência especificados.
let	Introduz uma variável de intervalo para armazenar os resultados de subexpressão em uma expressão de consulta.
Em	Palavra-chave contextual em uma cláusula join.
on	Palavra-chave contextual em uma cláusula join.
equals	Palavra-chave contextual em uma cláusula join.
by	Palavra-chave contextual em uma cláusula group.
ascending	Palavra-chave contextual em uma cláusula ordery.
descending	Palavra-chave contextual em uma cláusula ordery.

A execução

- **Execução adiada**: é a execução de uma variável de consulta. A consulta só é realizada quando existe iteração sobre a variável de consulta.

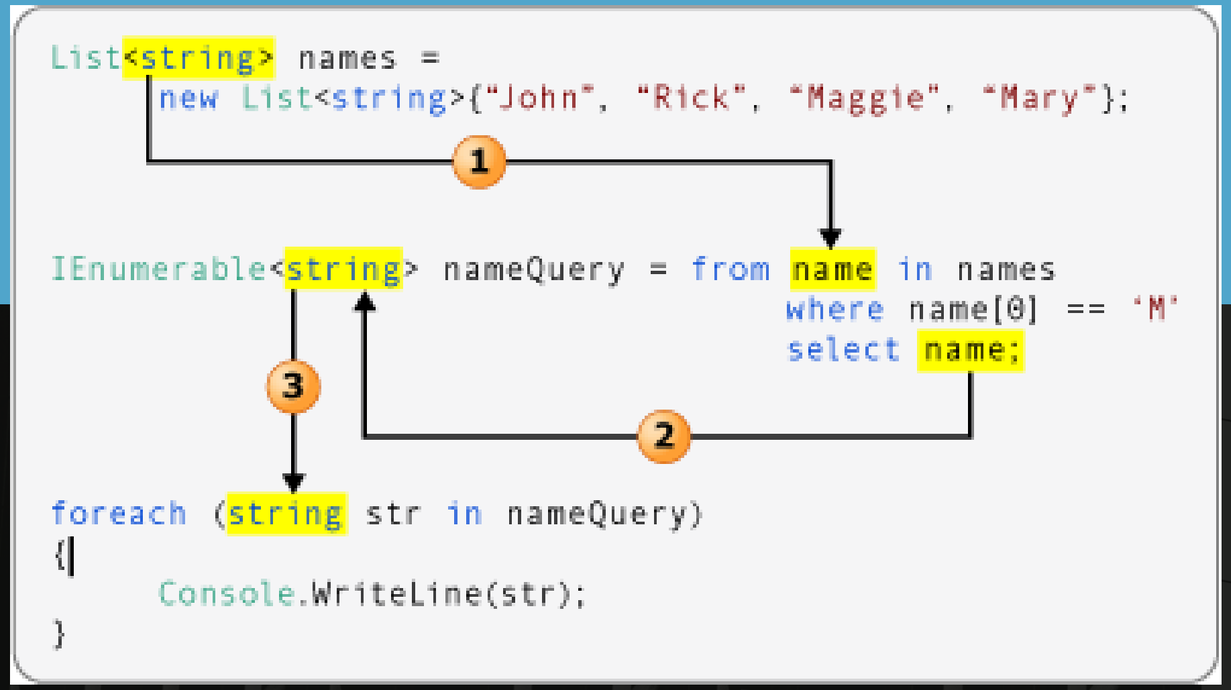
- **Execução imediata**: consultas que realizam funções de agregação sobre um intervalo de elementos deve iterar primeiramente sobre esses elementos. São utilizados em consultas com Count, Max, Average e First.

    Essas consultas retornam apenas um valor único e não uma coleção IEnumerable.

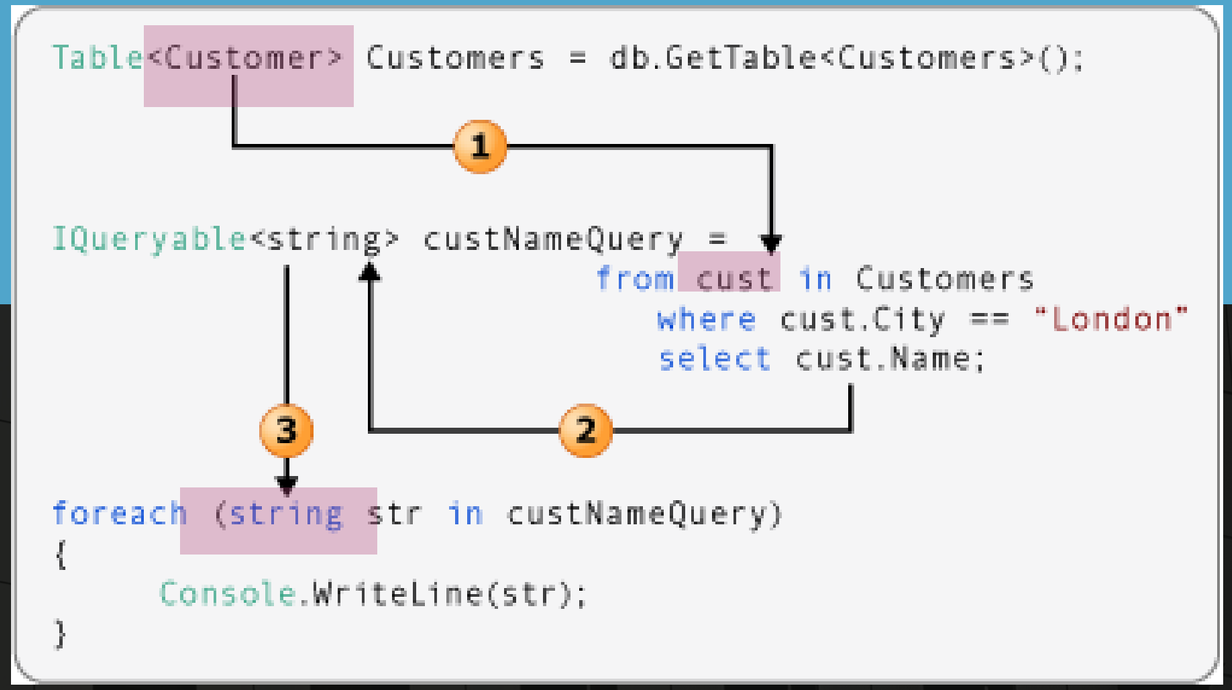


RELACIONAMENTO DE TIPOS EM CONSULTAS COM LINQ

Para escrever consultas com eficiencia, é preciso entender como os tipos de variaveis em uma operação de consulta completa se relacionam entre sí

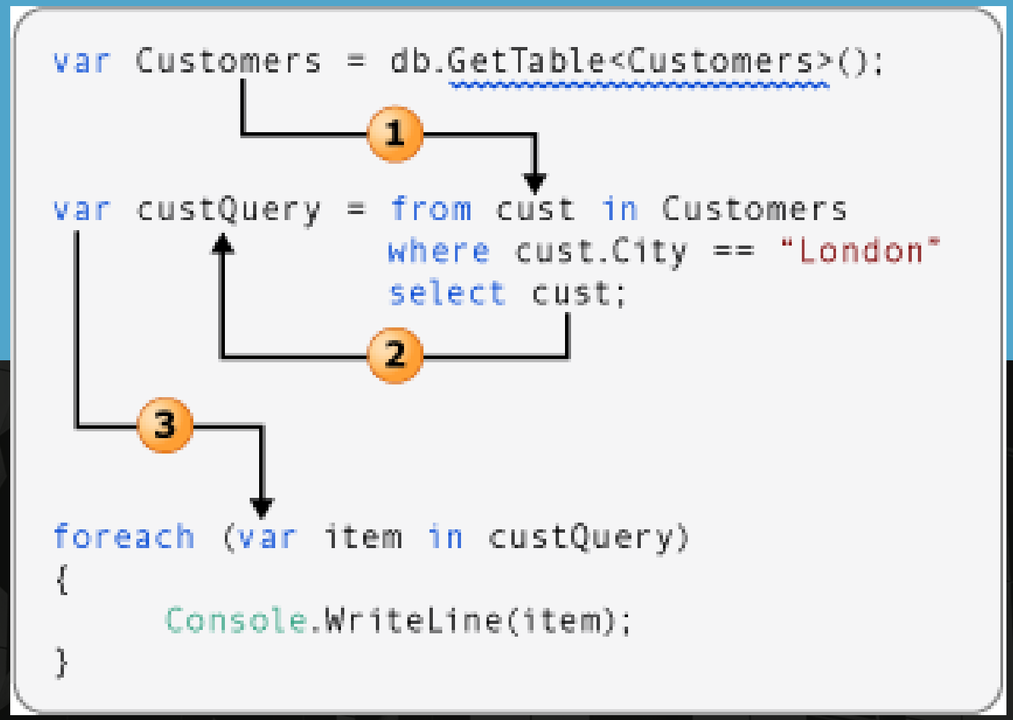


CONSULTAS QUE NÃO TRANSFORMAM OS DADOS DE ORIGEM



CONSULTAS QUE TRANSFORMAM OS DADOS DE ORIGEM

RELACIONAMENTO DE TIPOS EM CONSULTAS COM LINQ



PODE-SE UTILIZAR O "VAR" E DEIXAR QUE O COMPILADOR FAÇA A TIPAGEM FORTE, QUANDO NÃO SE SABE PARA QUAL TIPO OS DADOS SERÃO CONVERTIDOS

SINTAXE DE CONSULTA E SINTAXE DE MÉTODO

- As consultas podem ser escritas na sintaxe de consulta ou em sintaxe de método.
- Os operadores de consultas geram um novo tipo de método, os métodos de extensão.
- Métodos de extensão "estendem" um tipo existente, eles podem ser chamados como se fossem métodos de instância no tipo.
- Na sintaxe de método a cláusula Where está escrita como um método de instancia do objeto numbers. Isso é possível pois os operadores de consulta estendem o IEnumerable<T>.

```
//sintaxe de consulta
IEnumerable<int> numQuery1 =
from num in numbers
where num % 2 == 0
orderby num
select num;
```

```
//sintaxe de metodo
IEnumerable<int> numQuery2 = numbers.Where(num => num % 2 == 0).OrderBy(n => n);
```

Dúvidas?

ESTE ESTUDO FOI BASEADO NA DOCUMENTAÇÃO OFICIAL DA MICROSOFT E OS EXEMPLOS FORAM ADAPTADOS