

Introducción a C# y LINQ

LINQ es una poderosa herramienta de C# que simplifica la consulta y manipulación de datos en colecciones y bases de datos. LINQ proporciona una sintaxis unificada para trabajar con diferentes tipos de datos. Esta presentación explora algunos métodos LINQ comunes utilizados para trabajar con IEnumerable e IQueryable.



by Ariel Romero





Trabajando con IEnumerable e IQueryable

1 IEnumerable

IEnumerable representa una colección de elementos que se pueden iterar. Los métodos LINQ que funcionan con IEnumerable ejecutan consultas de forma diferida, es decir, la consulta no se ejecuta hasta que se itera sobre los resultados.

2 IQueryable

IQueryable representa una colección de elementos que se pueden consultar en una base de datos. Los métodos LINQ que funcionan con IQueryable traducen la consulta a lenguaje SQL y la ejecutan en la base de datos.

Método Where(para filtrar datos

Sintaxis

```
IEnumerable  
filteredCollection =  
collection.Where(predicate)  
;
```

Función

El método Where(filtra una colección de elementos basándose en un predicado que devuelve un valor booleano. Sólo los elementos que cumplen el predicado se incluyen en la colección filtrada.



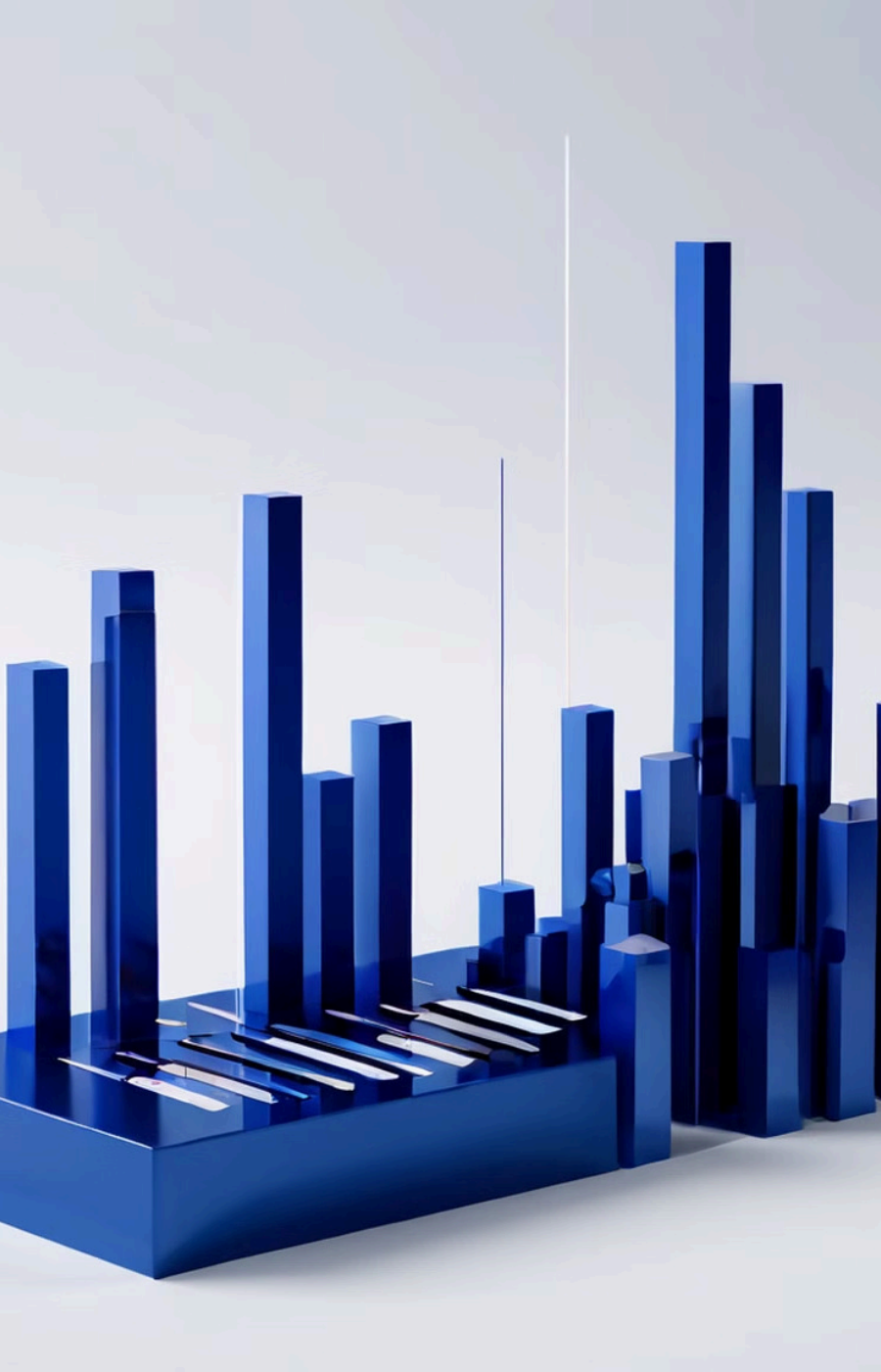
Método Select(para proyectar datos

Sintaxis

```
IEnumerable projectedCollection =  
collection.Select(projectionFunction);
```

Función

El método Select(transforma cada elemento de una colección en un nuevo tipo de objeto, proyectando los datos de acuerdo a una función de proyección. La función de proyección especifica cómo se transformará cada elemento.



Método OrderBy(y OrderByDescending(para ordenar datos

1

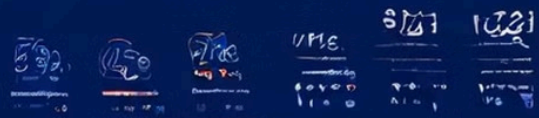
`OrderBy(`

Ordena una colección en orden ascendente. La función de ordenamiento especifica la propiedad o el criterio por el cual se ordenará la colección.

2

`OrderByDescending(`

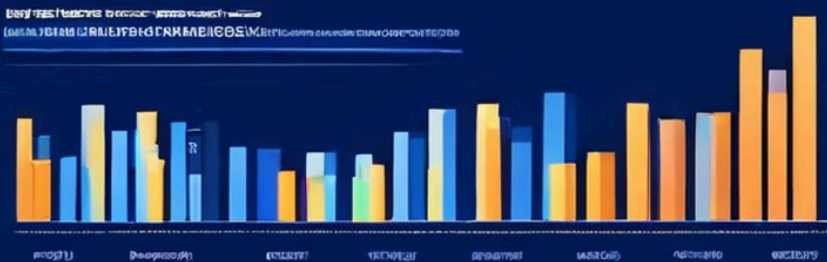
Ordena una colección en orden descendente. La función de ordenamiento especifica la propiedad o el criterio por el cual se ordenará la colección.



FOR YOUR
SITTING



COASTWATER NEWS



Método GroupBy(para agrupar datos

1

Sintaxis

```
IEnumerable> groupedCollection =  
collection.GroupBy(keySelector);
```

2

Función

El método `GroupBy()` agrupa los elementos de una colección basándose en un selector de clave. La clave del grupo se calcula aplicando el selector de clave a cada elemento.



Método Join(para combinar datos

Sintaxis

```
IEnumerable joinedCollection =  
collection1.Join(collection2, keySelector1,  
keySelector2, resultSelector);
```

Función

El método Join(combina los elementos de dos colecciones basándose en una coincidencia de clave. Los selectores de clave especifican la propiedad o el criterio por el cual se unirán los elementos de las colecciones.

Método FirstOrDefault(y SingleOrDefault(para obtener el primer o único elemento

FirstOrDefault(

Devuelve el primer elemento de la colección que satisface el predicado especificado, o el valor predeterminado del tipo si la colección está vacía.

SingleOrDefault(

Devuelve el único elemento de la colección que satisface el predicado especificado, o el valor predeterminado del tipo si la colección está vacía o si hay más de un elemento que cumple el predicado.

Método Count(, Sum(, Average(y otros métodos de agregación



Count(

Cuenta el número de elementos en la colección.



Sum(

Calcula la suma de los valores en la colección.



Average(

Calcula el promedio de los valores en la colección.



Min(

Devuelve el valor mínimo de la colección.



Max(

Devuelve el valor máximo de la colección.



Conclusión y recursos adicionales

LINQ es una herramienta invaluable para trabajar con datos en C#. Los métodos LINQ simplifican la consulta, manipulación y transformación de datos. Existen muchos otros métodos LINQ que no se cubrieron en esta presentación. Para más información, se recomienda consultar la documentación oficial de Microsoft.