

Ejercicios básicos con LINQ

Iris Pérez Aparicio
2º DAM
Campus FP Emprende Humanes

1. Filtrar elementos

Dado el array:

```
int[] numeros = { 5, 12, 8, 23, 4, 16, 19, 7, 10, 3 };
```

- Filtra los números mayores que 10 y muestra el resultado.

CÓDIGO:

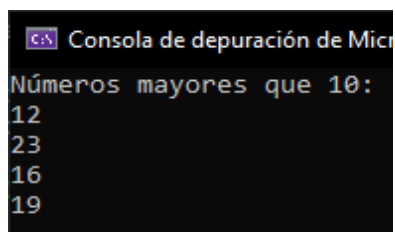
```
using System;
using System.Linq;

class Program
{
    static void Main()
    {
        int[] numeros = { 5, 12, 8, 23, 4, 16, 19, 7, 10, 3 };

        // Filtra los números mayores que 10 usando LINQ
        var mayoresQueDiez = numeros.Where(num => num > 10);

        // Imprimo los números mayores que 10
        Console.WriteLine("Números mayores que 10: ");
        foreach (int num in mayoresQueDiez)
        {
            Console.WriteLine(num);
        }
    }
}
```

EJECUCIÓN:



```
Consola de depuración de Micros...
Números mayores que 10:
12
23
16
19
```

2. Ordenar y transformar

- Ordena los números filtrados en el ejercicio anterior de forma descendente.
- Multiplica cada número resultante por 2 y muestra el resultado.

CÓDIGO:

```
using System;
using System.Linq;
using System.Collections.Generic;

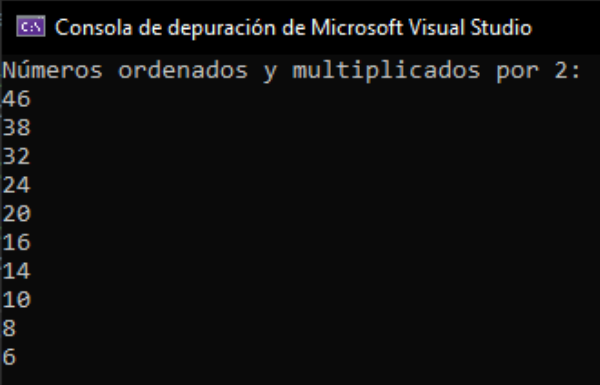
class Program
{
    static void Main()
    {
        int[] numeros = { 5, 12, 8, 23, 4, 16, 19, 7, 10, 3 };

        // Ordeno los números de mayor a menor
        var numerosOrdenados = numeros.OrderByDescending(num => num);

        // Multiplico por 2 cada número
        var numerosOrdenadosPor2 = numerosOrdenados.Select(n => n * 2);

        // Muestro los números ordenados de forma descendente y
        // multiplicados por 2
        Console.WriteLine("Números ordenados y multiplicados por 2:");
        foreach (int num in numerosOrdenadosPor2)
        {
            Console.WriteLine(num);
        }
    }
}
```

EJECUCIÓN:



Consola de depuración de Microsoft Visual Studio

Números ordenados y multiplicados por 2:

46
38
32
24
20
16
14
10
8
6

3. Eliminar duplicados y sumar

Dada la lista:

```
List<int> lista = new List<int> { 2, 4, 4, 6, 8, 2, 10 };
```

- Elimina los números duplicados y muestra la lista resultante.
- Suma todos los números únicos y muestra el resultado.

CÓDIGO:

```
using System;
using System.Linq;
using System.Collections.Generic;

class Program
{
    static void Main()
    {
        List<int> lista = new List<int> { 2, 4, 4, 6, 8, 2, 10 };

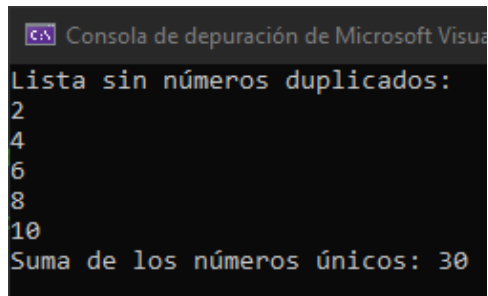
        // Elimino los números duplicados usando Distinct
        var listaSinDuplicados = lista.Distinct();

        // Muestro la lista resultante tras eliminar los duplicados
        Console.WriteLine("Lista sin números duplicados:");
        foreach (int num in listaSinDuplicados)
        {
            Console.WriteLine(num);
        }

        // Sumo todos los números únicos
        int sumaNumerosUnicos = listaSinDuplicados.Sum();

        // Muestro el resultado de la suma de todos los números únicos
        Console.WriteLine("Suma de los números únicos: " +
            sumaNumerosUnicos);
    }
}
```

EJECUCIÓN:



```
Consola de depuración de Microsoft Visual Studio
Lista sin números duplicados:
2
4
6
8
10
Suma de los números únicos: 30
```

4. Contar pares e impares

- Cuenta cuántos números pares e impares hay en la lista del ejercicio anterior y muestra ambos resultados en una sola sentencia LINQ.

CÓDIGO:

```
using System;
using System.Linq;
using System.Collections.Generic;

class Program
{
    static void Main()
    {
        List<int> lista = new List<int> { 2, 4, 4, 6, 8, 2, 10 };

        Console.WriteLine("Lista ejercicio anterior:");

        // Cuento cuantos números pares e impares hay y muestro ambos resultados
        lista.GroupBy(num => num % 2 == 0 ? "Pares" : "Impares") //
        Agrupo los números en pares e impares
            .ToList().ForEach(grupo => Console.WriteLine($"{grupo.Key}: {grupo.Count()}")); // Por cada grupo, muestro su nombre o clave (Key) y el número de elementos (Count())

        // (La lista del ejercicio anterior solo contiene números pares)
        // Voy a probar con una lista que contenga números pares e
        impares
        Console.WriteLine("\nLista inventada de prueba:");

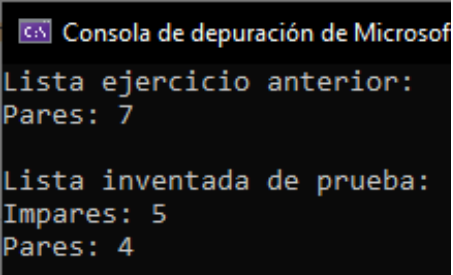
        List<int> listaInventada = new List<int> { 1, 3, 7, 4, 10, 11, 14, 15, 20 };
    }
}
```

```

        // Cuento cuantos números pares e impares hay y muestro ambos
resultados
        listaInventada.GroupBy(num => num % 2 == 0 ? "Pares" :
"Impares")
                        .ToList().ForEach(grupo => Console.WriteLine($"{grupo.Key}:
{grupo.Count()}"));
    }
}

```

EJECUCIÓN:



```

C:\> Consola de depuración de Microsoft
Lista ejercicio anterior:
Pares: 7

Lista inventada de prueba:
Impares: 5
Pares: 4

```

5. Agrupar y contar repeticiones

Dada la lista:

```
List<int> lista = new List<int> { 1, 2, 2, 3, 3, 3, 4 };
```

- Muestra cada número y la cantidad de veces que aparece en la lista (usa `GroupBy`).

CÓDIGO:

```

using System;
using System.Collections.Generic;
using System.Linq;

class Program
{
    static void Main()
    {
        List<int> lista = new List<int> { 1, 2, 2, 3, 3, 3, 4 };

        // Muestro cada número y la cantidad de veces que aparece en la
lista
        lista.GroupBy(num => num).ToList().ForEach(num =>
Console.WriteLine($"Número {num.Key} aparece {num.Count()} veces."));
    }
}

```

```
}  
}
```

EJECUCIÓN:

```
Consola de depuración de Micros  
Número 1 aparece 1 veces.  
Número 2 aparece 2 veces.  
Número 3 aparece 3 veces.  
Número 4 aparece 1 veces.
```

6. Promedio de números mayores a un valor

- Calcula el promedio de todos los números mayores a 1 en la lista del ejercicio anterior.

CÓDIGO:

```
using System;  
using System.Linq;  
using System.Collections.Generic;  
  
class Program  
{  
    static void Main()  
    {  
        List<int> lista = new List<int> { 1, 2, 2, 3, 3, 3, 4 };  
  
        // Filtro la lista dejando solo los números mayores a 1  
        var numerosMayoresQue1 = lista.Where(num => num > 1);  
        // Calculo el promedio con .Average()  
        double promedio = numerosMayoresQue1.Average();  
  
        Console.WriteLine("El promedio de todos los números mayores a 1  
en la lista es: " + promedio);  
    }  
}
```

EJECUCIÓN:

```
Consola de depuración de Microsoft Visual Studio  
El promedio de todos los números mayores a 1 en la lista es: 2,8333333333333335
```

7. Crear nuevos objetos

Dada la lista:

```
List<string> nombres = new List<string> { "Ana", "Luis", "Marta" };
```

- Usa LINQ para crear una nueva lista de objetos anónimos que tengan dos propiedades: **Nombre** y **Longitud** (la cantidad de letras del nombre).

CÓDIGO:

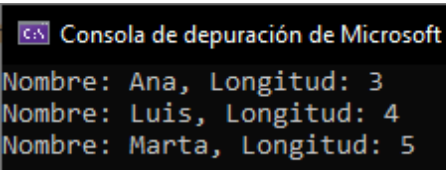
```
using System;
using System.Linq;
using System.Collections.Generic;

class Program
{
    static void Main()
    {
        List<string> nombres = new List<string> { "Ana", "Luis", "Marta"
    };

        // Creo una lista de objetos anónimos con las propiedades
        'Nombre' y 'Longitud' (cantidad de letras del nombre)
        var listaObjetosAnonimos = nombres
            .Select(nombre => new { Nombre = nombre, Longitud =
nombre.Length }) // Por cada nombre de la lista nombres, creo un objeto
con las propiedades Nombre y Longitud
            .ToList(); // Convierto el resultado en una lista

        // Muestro las propiedades de todos los objetos de la lista uno
a uno
        listaObjetosAnonimos.ForEach(objeto =>
Console.WriteLine($"Nombre: {objeto.Nombre}, Longitud:
{objeto.Longitud}"));
    }
}
```

EJECUCIÓN:



```
Consola de depuración de Microsoft
Nombre: Ana, Longitud: 3
Nombre: Luis, Longitud: 4
Nombre: Marta, Longitud: 5
```