

Linq agrega funcionalidades de consulta a los lenguajes C# y Visual Basic.Net respecto cualquier almacén de datos, es decir, nos permite aplicar algo parecido a consultas SQL bien a una lista, a xml, o a otros orígenes de datos, incluso a una base de datos.

Una consulta linq requiere tres acciones:

1. Obtención del origen de datos
2. Creación de la consulta
3. Ejecución de la consulta

Ejemplo1: ordenar una lista de enteros con Linq

1. Obtención de origen de datos:

```
int[] arregloEnteros = { 1, 2, 3, 5, 3, 2, 8, 9};
```

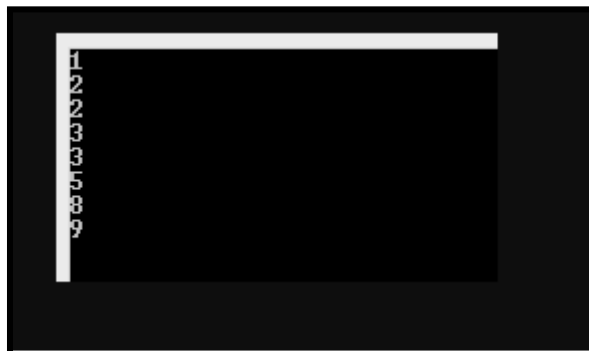
2. Creación de la consulta

```
IEnumerable<int> IstEnterosOrdenados = from nums in arregloEnteros
                                         orderby nums
                                         select nums;
```

En este punto aún **no se ejecuta la consulta** solo la guardamos en este caso en una colección de la instancia *IEnumerable*. Cualquier tipo de dato que admita implícitamente la interfaz *IEnumerable* o alguna de sus derivaciones, se le puede aplicar LINQ.

3. Ejecución de la consulta

```
foreach(int n in IstEnterosOrdenados){
    Console.WriteLine(n);
}
```



Ejemplo2: Filtrado de elementos

```
//Origen de datos
int[] arregloEnteros = { 1, 2, 3, 5, 3, 2, 8, 9 };

//Consulta, utilizamos where como en sql para obtener números menores a 5
IEnumerable<int> lstEnterosMenoresCinco = from nums in arregloEnteros
    where nums < 5
    select nums;

//Ejecución
foreach (int n in lstEnterosMenoresCinco) {
    Console.WriteLine(n);
}
```



Ejemplo3: Uso de de Linq en elementos complejos

```
//clase Complejo
public class Complejo
{
    public int numero{get;set;}
    public string cadena{get;set;}

    public Complejo(int numero, string cadena)
    {
        this.numero = numero;
        this.cadena = cadena;
    }

    public string imprime()
    {
        return numero + " " + cadena;
    }
}

//origen de datos
Complejo[] lstComplejos={
    new Complejo(1,"pato"),
    new Complejo(5,"perro"),
```

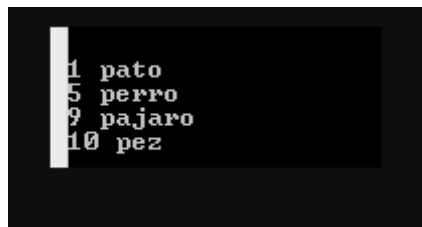
```
new Complejo(9,"pajaro"),
new Complejo(10,"pez"),
new Complejo(2,"ave"),
new Complejo(4,"gusano"),
};
```

//consulta

```
IEnumerable<Complejo> IstComplejosConP =from d in IstComplejos
                                         where d.cadena.StartsWith("p")
                                         select d;
```

//ejecución de consulta

```
foreach(Complejo c in IstComplejosConP){
    Console.WriteLine(c.imprime());
}
```



Ejemplo4: Linq en elementos complejos; obtener una colección reducida

//utilizamos la misma clase Complejo

//origen de datos

```
Complejo[] IstComplejos={
new Complejo(1,"pato"),
new Complejo(5,"perro"),
new Complejo(9,"pajaro"),
new Complejo(10,"pez"),
new Complejo(2,"ave"),
new Complejo(4,"gusano"),
};
```

//Consulta, reducimos la clase Complejo a una lista de cadenas

```
IEnumerable IstSoloCadenas = from d in IstComplejos
                              select d.cadena;
```

//ejecución de consulta

```
foreach (string s in IstSoloCadenas){
    Console.WriteLine(s);
}
```

```
pato  
perro  
pajaro  
pez  
ave  
gusano
```