



## List en c#

Una lista es una estructura dinámica en la que se puede añadir elementos sin tantas restricciones.

Su sintaxis es:

```
<list type="bullet|number|table">  
  <listheader>  
    <term>term</term>  
    <description>description</description>  
  </listheader>  
  <item>  
    <term>term</term>  
    <description>description</description>  
  </item>  
</list>
```

<listheader> : Se usa para definir la fila de encabezado de una tabla o de una lista de definiciones.

<ítem> : Se usa para especificar cada elemento de la lista.

### Sus parámetros son:

Term

Término que se define en description.

Description

Elemento de una lista numerada o con viñetas, o definición de un term.

En el caso de C#, tenemos dos variantes: ArrayList: una lista a cuyos elementos se puede acceder como a los de un array y SortedList : que es una lista ordenada.

## ArrayList

En un ArrayList, podemos añadir datos en la última posición con "Add", insertar en cualquier otra con "Insert", recuperar cualquier elemento usando corchetes, o incluso ordenar toda la lista con "Sort".

Ejemplo:

```
using System;
using System.Collections;

public class Ejemplo_11_04_01a
{
    public static void Main()
    {
        ArrayList miLista = new ArrayList();
        // Añadimos en orden
        miLista.Add("Hola,");
        miLista.Add("soy");
        miLista.Add("yo");

        // Mostramos lo que contiene
        Console.WriteLine( "Contenido actual:");
        foreach (string frase in miLista)
            Console.WriteLine( frase );

        // Accedemos a una posición
        Console.WriteLine( "La segunda palabra es: {0}",
            miLista[1] );

        // Insertamos en la segunda posición
        miLista.Insert(1, "Como estas?");

        // Mostramos de otra forma lo que contiene
```

```
Console.WriteLine( "Contenido tras insertar:");
for (int i=0; i<miLista.Count; i++)
    Console.WriteLine( miLista[i] );

// Buscamos un elemento
Console.WriteLine( "La palabra \"yo\" está en la posición {0}",
    miLista.IndexOf("yo") );

// Ordenamos
miLista.Sort();

// Mostramos lo que contiene
Console.WriteLine( "Contenido tras ordenar");
foreach (string frase in miLista)
    Console.WriteLine( frase );

// Buscamos con búsqueda binaria
Console.WriteLine( "Ahora \"yo\" está en la posición {0}",
    miLista.BinarySearch("yo") );

// Invertimos la lista
miLista.Reverse();

// Borramos el segundo dato y la palabra "yo"
miLista.RemoveAt(1);
miLista.Remove("yo");

// Mostramos nuevamente lo que contiene
Console.WriteLine( "Contenido dar la vuelta y tras eliminar dos:");
foreach (string frase in miLista)
    Console.WriteLine( frase );

// Ordenamos y vemos dónde iría un nuevo dato
miLista.Sort();
```

```

Console.WriteLine( "La frase \"Hasta Luego\"...");
int posicion = miLista.BinarySearch("Hasta Luego");
if (posicion >= 0)
    Console.WriteLine( "Está en la posición {0}", posicion );
else
    Console.WriteLine( "No está. El dato inmediatamente mayor "+
        "es el {0}: {1}",
        ~posicion, miLista[~posicion] );
}
}

```

Resultado:

Contenido actual:

Hola,

soy

yo

La segunda palabra es: soy

Contenido tras insertar:

Hola,

Como estas?

soy

yo

La palabra "yo" está en la posición 3

Contenido tras ordenar

Como estas?

Hola,

soy

yo

Ahora "yo" está en la posición 3

Contenido dar la vuelta y tras eliminar dos:

Hola,

Como estas?

La frase "Hasta Luego"...

No está. El dato inmediatamente mayor es el 1: Hola,

Información de <https://docs.microsoft.com/es-es/dotnet/csharp/programming-guide/xml/doc/list> y <http://www.nachocabanes.com/csharp/curso2015/csharp11d.php>