



## Ejemplo 1. Listas abiertas

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

namespace listas_csharp
{
    class Program
    {
        static void Main(string[] args)
        {
            List<nodo> alumnos = new List<nodo>();
            alumnos.Add(new nodo("Susy",20,9.1F));
            alumnos.Add(new nodo("Juan", 21, 8.7F));
            alumnos.Add(new nodo("Jorge", 22, 9.6F));

            //Recorrer lista modo 1
            Console.WriteLine();
            foreach (nodo N in alumnos)
                Console.WriteLine("{0},{1},{2}",N.nombre, N.edad,N.promedio);

            //Recorrer lista modo 2
            Console.WriteLine();
            for (int i = 0; i < alumnos.Count; i++)
                Console.WriteLine("{0},{1},{2}", alumnos[i].nombre, alumnos[i].edad,
alumnos[i].promedio);

            //delegate nos devuelve una referencia de un método encapsulado
            nodo pNodo = alumnos.Find(delegate(nodo P)
            {
                if(P.nombre.Equals("Pedro"))
                    return true;
                else
                    return false;
            });

            Console.WriteLine();
            if (pNodo != null)
                Console.WriteLine("{0},{1},{2}", pNodo.nombre, pNodo.edad, pNodo.promedio);
            else
                Console.WriteLine("No hay nodos con ese criterio");

            //Eliminar nodo
            alumnos.Remove(pNodo);

            pNodo=alumnos.Find(busca);
            Console.WriteLine();
            if (pNodo != null)
                Console.WriteLine("{0},{1},{2}", pNodo.nombre, pNodo.edad, pNodo.promedio);
            else
                Console.WriteLine("No hay nodos con ese criterio");
```

### Ejemplos Listas



```
//Eliminar por el indice del elemento
alumnos.RemoveAt(1);

//Recorrer lista modo 1
Console.WriteLine();
foreach (nodo N in alumnos)
    Console.WriteLine("{0},{1},{2}", N.nombre, N.edad, N.promedio);

//Cuantos elementos tengo
Console.WriteLine(alumnos.Count());

}
private static bool busca(nodo P)
{
    if(P.nombre.Equals("Juan"))
        return true;
    else
        return false;
}
}
class nodo
{
    public string nombre;
    public int edad;
    public float promedio;

    public nodo(string nom, int e, float prom)
    {
        this.nombre = nom;
        this.edad = e;
        this.promedio = prom;
    }
}
}
```



## Ejemplo 2. Listas abiertas

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

namespace listas_csharp2
{
    class Program
    {
        static void Main(string[] args)
        {
            List<int> MiLista=new List<int>();
            MiLista.Add(5);
            MiLista.Add(3);
            MiLista.Add(6);
            MiLista.Add(3);
            MiLista.Add(3);
            //Imprimir lista
            foreach (int N in MiLista)
                Console.WriteLine("{0}",N);
            //Indice de dato buscado
            Console.WriteLine(MiLista.IndexOf(3));
            Console.WriteLine(MiLista.IndexOf(9));

            //Reemplazar un valor buscado
            MiLista[MiLista.IndexOf(3)] = 7;

            //Mostramos nuevamente la lista
            foreach (int N in MiLista)
                Console.WriteLine("{0}", N);

            //Insertar un valor en la posicion indicada
            MiLista.Insert(3, 9);

            //Mostramos nuevamente la lista
            foreach (int N in MiLista)
                Console.WriteLine("{0}", N);

            //Ordenamos la lista
            MiLista.Sort();

            //Mostramos nuevamente la lista
            foreach (int N in MiLista)
                Console.WriteLine("{0}", N);
        }
    }
}
```



### Ejemplo 3. Listas abiertas

```
using System;  
using System.Collections.Generic; //Esta libreria es la que me permite trabajar con listas  
using System.Linq;  
using System.Text;
```

```
namespace listas_csharp3  
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            //Creamos la lista  
            List<nodo> LISTA = new List<nodo>();  
            Random r = new Random(); //Clase para generar números aleatorios  
            int j;  
            for (int i = 0; i < 10; i++)  
            {  
                j = r.Next(1, 500); //Creamos un número aleatorio entre 1 y 500  
                LISTA.Add(new nodo(i * j)); //Agregamos los nodos a la lista  
            }  
  
            //Forma 1: Recorremos la lista  
            Console.WriteLine("***** Recorriendo la lista por foreach *****");  
            foreach (nodo P in LISTA)  
                Console.WriteLine(P.dato);  
  
            //Forma 2: Recorremos la lista  
            Console.WriteLine("***** Recorriendo la lista por for *****");  
            for(int i=0; i<LISTA.Count; i++)  
                Console.WriteLine(LISTA[i].dato);  
  
            //Borramos un elemento de la lista de acuerdo al indice que ocupa  
            LISTA.RemoveAt(3);  
            Console.WriteLine("***** Lista sin elemento en el indice 3 *****");  
            foreach (nodo P in LISTA)  
                Console.WriteLine(P.dato);  
  
            //Insertamos un valor en el indice 3  
            LISTA.Insert(3, new nodo(-1000));  
            Console.WriteLine("***** Lista con nuevo elemento en el indice 3 *****");  
            foreach (nodo P in LISTA)  
                Console.WriteLine(P.dato);  
  
            //Ordenamos la lista  
            LISTA.Sort(delegate(nodo Primero, nodo Segundo) { return  
Primero.dato.CompareTo(Segundo.dato); });  
            Console.WriteLine("***** Lista ordenada *****");  
            foreach (nodo P in LISTA)  
                Console.WriteLine(P.dato);  
  
            //Buscar un elemento
```



TEMA3. ESTRUCTURAS DE DATOS COMPUESTAS

```
Console.WriteLine("Escribe el número que deseas buscar:");
int valor_buscar = Convert.ToInt32(Console.ReadLine());
nodo Q = LISTA.Find(delegate(nodo P) { return P.dato==valor_buscar; });
if(Q!=null)//Se encontro
    Console.WriteLine(Q.dato);
else
    Console.WriteLine("No se encontro");

//Buscar por indice
int indice = LISTA.FindIndex(delegate(nodo P) { return P.dato == valor_buscar; });
if(indice > 0) //Se encontro
    Console.WriteLine("Posicion en la lista: {0}", indice);
else
    Console.WriteLine("No se encontro");

//Borramos la lista
LISTA.Clear();
}
static bool encuentra(nodo P, int valor)
{
    return (P.dato == valor);
}
}
class nodo
{
    public int dato;

    //Método Constructor, nos sirve para inicializar las variables al crear el objeto
    public nodo(int dato)
    {
        this.dato = dato;
    }
}
}
```



## Ejemplo 4. Listas circulares

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

namespace lista_circular
{
    class Program
    {
        nodo LISTA = null;
        nodo ULTIMO = null;
        static void Main(string[] args)
        {
            Program objeto_lista = new Program();
            imprime("Vamos a crear una lista de 10 numeros");
            for (int i = 0; i < 10; i++)
            {
                Console.Write("{0}", i + 1);
                objeto_lista.insertar_nodo(Numeros.entero("Escriba un número.));
            }
            imprime("La lista circular es ");
            objeto_lista.recorrer_lista();

            //Si cambio lista al siguiente se imprimiria asi
            imprime("\nRecorriendo nuevamente lista ");
            objeto_lista.LISTA = objeto_lista.LISTA.siguiente;
            objeto_lista.recorrer_lista();
        }
        void insertar_nodo(int N)
        {
            nodo P = new nodo();
            P.n = N;
            P.siguiente = LISTA;
            LISTA = P;
            if (ULTIMO == null)
                ULTIMO = LISTA;
            else
                ULTIMO.siguiente = LISTA;
        }
        void recorrer_lista()
        {
            nodo Q = LISTA;
            if (Q != null)
            {
                do
                {
                    Console.Write(Q.n + "-->");
                    Q = Q.siguiente;
                } while (Q != LISTA);
            }
        }
    }
}
```



```
    }  
  }  
  
  static void imprime(string cadena)  
  {  
    Console.WriteLine(cadena);  
  }  
}  
class nodo  
{  
  public int n;  
  public nodo siguiente;  
}  
}
```



### Ejemplo 5. Pilas

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

namespace Pilas
{
    class Program
    {
        static void Main(string[] args)
        {
            Stack<int> Pila = new Stack<int>();
            Pila.Push(4);
            Pila.Push(5);
            Pila.Push(3);
            Pila.Push(11);
            Pila.Push(7);

            //Muestra el elemento que se encuentra hasta arriba de la pila
            Console.WriteLine("Elemento que se encuentra hasta arriba");
            Console.WriteLine(Pila.Peek());

            Console.WriteLine("Procesando la Pila");
            //Procesa el valor de la pila
            while (Pila.Count > 0)
            {
                Console.WriteLine(Pila.Pop());
            }
        }
    }
}
```



## Ejemplo 6. Colas

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

namespace Colas
{
    class Program
    {
        static void Main(string[] args)
        {
            Queue<int> Cola = new Queue<int>();
            //Insertando elementos
            Cola.Enqueue(5);
            Cola.Enqueue(3);
            Cola.Enqueue(1);
            Cola.Enqueue(2);
            Cola.Enqueue(4);

            //Primer elemento de la cola
            Console.WriteLine("Primer elemento de la cola");
            Console.WriteLine(Cola.Peek());

            //Procesando la cola
            Console.WriteLine("Procesando la cola");
            while (Cola.Count > 0)
            {
                Console.WriteLine(Cola.Dequeue());
            }
        }
    }
}
```