


```

        // list: [60,3,3]

1.Clear()           // Se eliminan todos los elementos de la lista.
                    // Vacía la lista.

                    // list: [0]
    }
    ,

```

- Las ventajas de una **LISTA** consiste en que puede contener tantos elementos como necesitemos en su interior, esto es, **no tienen límite de capacidad**. Son más sencillas y ágiles de trabajar que el resto de colecciones. Su principal uso es para tratar con datos que se almacenan y se destruyen de manera dinámica.

Arrays

- Son colecciones de elementos, **rígidas y estáticas**, del tipo que se indique en su definición. Tienen la característica particular que no permiten cambiar el tamaño de la colección, ni añadir o eliminar elementos posteriormente a su creación. De hecho, es obligatorio definir su tamaño en el momento que creamos el objeto.

```

,
public void CreateArrays()
{
    int [] array;           // Se crea la variable 'a' que es del tipo 'array de int'
    array = new int[4];     // Se crea un array en una posición de la memoria
                           // donde apunta el puntero 'a'
                           // Es imprescindible especificar el número de celdas
                           // que contiene el array.
                           // Este número no se puede cambiar.

                           // array: [0,0,0,0]

    array[3] = -10;
    array[0] = a[3];        // array: [-10,0,0,-10]

    int n = array.Length;   // n = 4

    int [] b = array;
                           // b = [-10,0,0,-10]
                           // b, del tipo 'array de int',
                           // apunta a la misma dirección que a.
}
,

```

Funciones de ejemplo

- **FUNCIÓN:** Se le pasa una lista de strings y devuelve el número de elementos que hay en su interior.

```
,  
public class ListExample  
{  
    public static int GetListItems(List<string> list)  
    {  
        return list.Count;  
    }  
}
```

- **'list'** es solo el puntero que apunta una dirección de la RAM donde se almacenan los datos particulares. Realmente es un número entero, por ejemplo 380.000. Esto quiere decir que hasta que no hacemos el 'NEW' no estamos creando ninguna lista
- **FUNCIÓN:** Se le pasa una lista de dobles y devuelve el número de elementos que son positivos.

```
,  
public static int GetPositiveListItems(List<double> list)  
{  
    int result = 0;  
    for(int i = 0; i < list.Count; i++)  
    {  
        if(list[i] > 0)  
            result++;  
    }  
    return result;  
}
```

- **FUNCIÓN:** Se le pasa un array de dobles y devuelve el número de elementos que son positivos.

```
,  
public static int GetPositiveArrayItems(double[] array)  
{  
    int result = 0;  
    for(int i = 0; i < array.Length; i++)  
    {  
        if(array[i] > 0)  
            result++;  
    }  
    return result;  
}
```

- La ventaja de un **ARRAY** es que permanecen inmutables a los cambios y definen una estructura muy rígida y estable de agrupación de datos. Son útiles para cálculos matemáticos y otras operaciones que mantengan los datos inmutables.