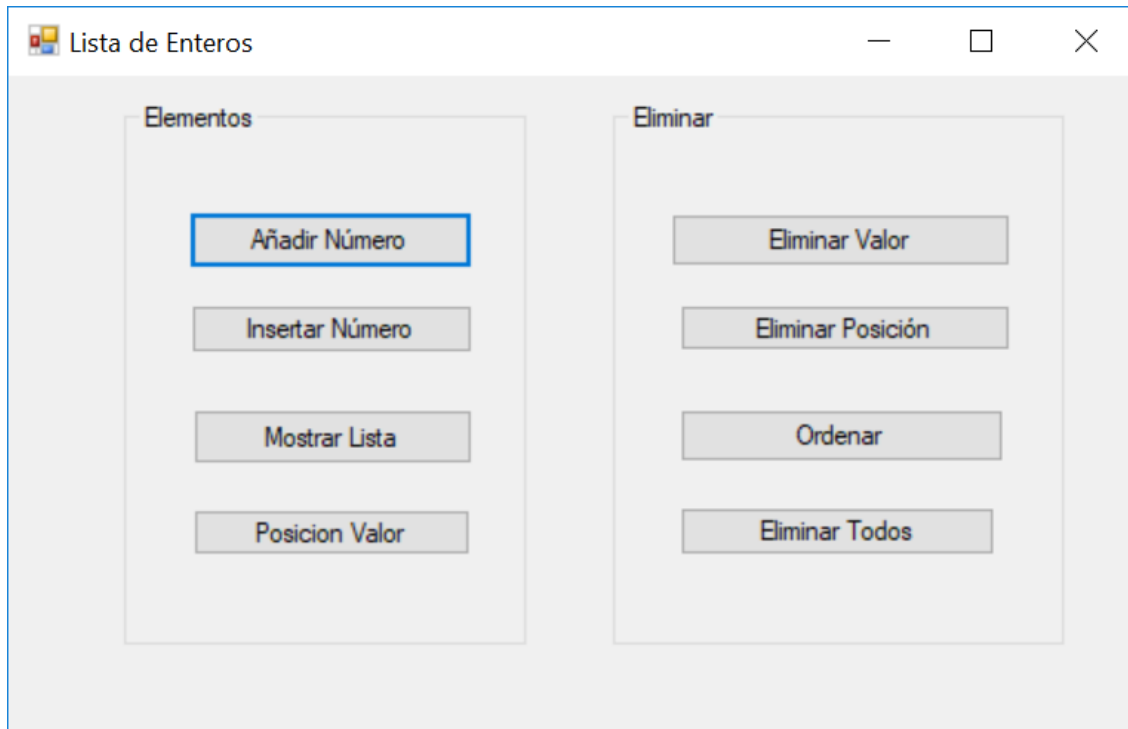


## Ejercicios

1. Realizar un ejercicio en el que trabajemos con una lista (List) de valores enteros. Tendremos las siguientes opciones en nuestro programa (**Ejercicio Resuelto**):



2. Realizar un programa en el que se lea una lista de números enteros. Iremos leyendo números y preguntando al usuario si quiere introducir más números. Cuando haya terminado de introducir los números haremos dos subprogramas:
  - Un subprograma en el que **copiaremos** en otra lista todos los números pares.
  - Otro subprograma en el que pasaremos (moveremos) los pares a otra lista de manera que en la primera lista se quedarán únicamente los impares y en la segunda los pares. (**Ejercicio Resuelto**).
3. Realizar un programa como el anterior con dos subprogramas en el que lo que copiemos o pasemos a la segunda lista sean los números **primos**. Os recomiendo hacer una función que nos devuelva si un número es primo.
4. Realizar un programa en el que leamos una lista de números Base y otra lista de números Exponente. Las dos listas se leerán exactamente con 10 números. Realizar un subprograma que obtenga una lista con la potencia **elemento a elemento** de las dos anteriores.

5. Se leen dos listas (List) de enteros. A continuación, ordenamos esas dos listas. Tendremos dos botones más: para crear una tercera lista mediante intercalación de las dos anteriores. (Meteremos, a partir de las listas ordenadas, elemento a elemento en una tercera lista).

En la primera opción lo haremos recorriendo las dos listas, **añadiendo** el elemento que corresponda en la tercera (**Resuelto**).

**Ejemplo:**

**Lista 1: (1, 4, 8, 8, 10, 20)**

**Lista 2: (2, 3, 4, 6, 11, 12, 13)**

**Lista 3: (1, 2, 3, 4, 4, 6, 8, 8, 10, 11, 12, 13, 20)**

En la segunda opción lo haremos **quitando** cada elemento de la lista correspondiente y añadiéndolo a la tercera, es decir, al terminar esta tercera opción las dos listas iniciales habrán quedado vacías (**Por resolver**).

6. Modificar el ejercicio anterior de manera que, en las dos opciones, **no se introduzcan elementos repetidos** en la **tercera** lista. (Recordad que los ArrayList tienen un método llamado Contains).

**Ejemplo:**

**Lista 1: (1, 4, 8, 8, 10, 20)**

**Lista 2: (2, 3, 4, 4, 4, 6, 10, 10, 11, 12, 13)**

**Lista 3: (1, 2, 3, 4, 6, 8, 10, 11, 12, 13, 20)**

7. Realizar un programa en el que tengamos un arrayList en el que se van introduciendo palabras, de manera que cada vez que se introduzca una palabra, esta se inserte en su posición por orden alfabético. La idea es que al introducir el usuario la palabra encontremos en qué **posición** le tocaría y **utilicemos lista.Insert** para insertarla en la posición que le corresponda.

Recordad que **String.Compare** es la función que nos permite comparar dos strings alfabéticamente.

8. Realizar un programa que simule el funcionamiento de la lotería primitiva.

Para ello utilizaremos dos List.

En un List se meterán los **números del sorteo**. Introduciremos 6 números. Estos números se calcularán de forma aleatoria mediante la función [Random](#). Tendremos en nuestra aplicación un botón para realizar el sorteo.

En la otra List el usuario introducirá los números con los que quiere jugar. Habrá que controlar que los números estén entre 1 y 49 y que no meta números repetidos. Se llamará desde otro botón.

Finalmente, en un último botón, debemos mostrar por pantalla la **cantidad de números** que el usuario ha acertado respecto a los del sorteo.