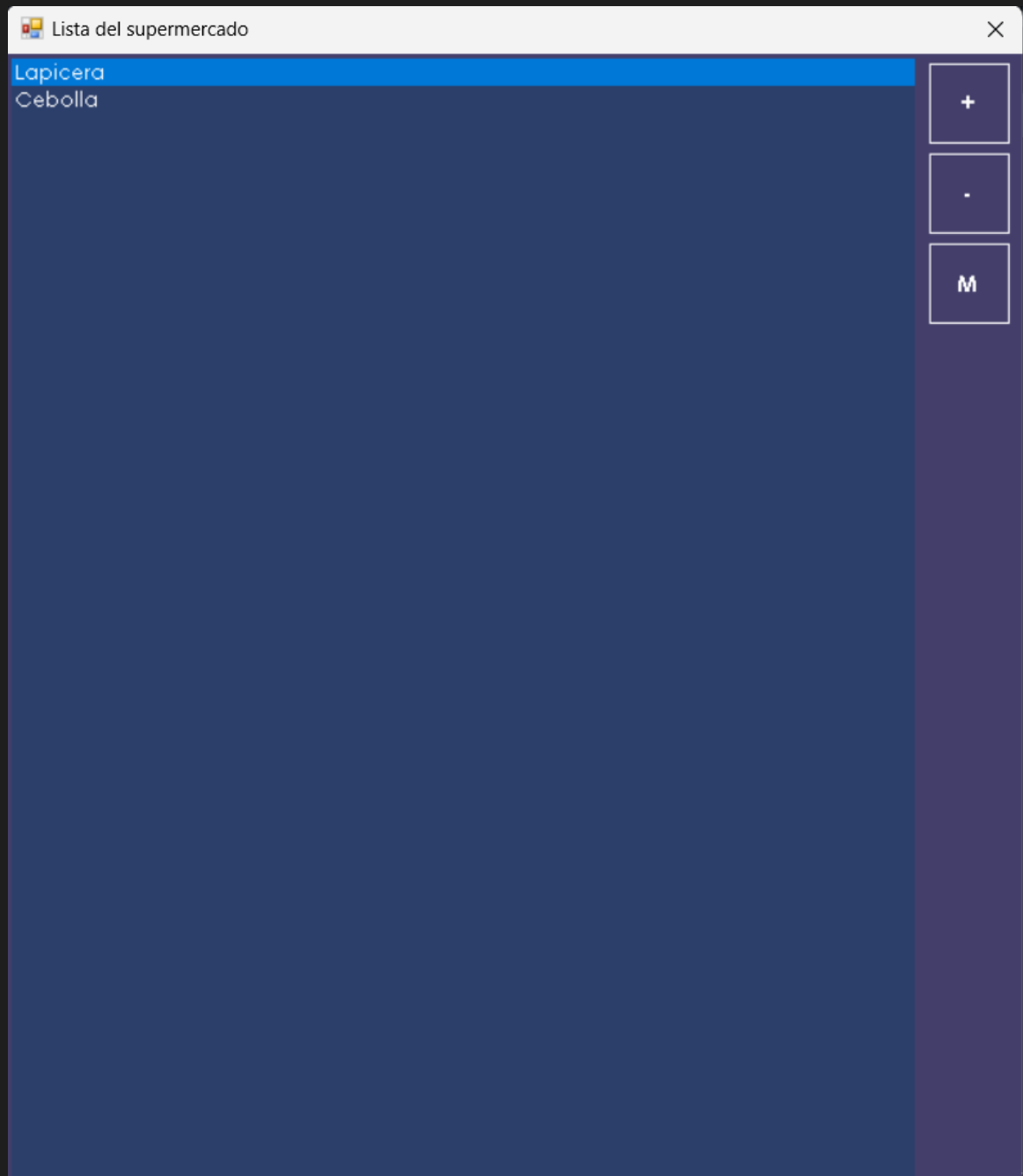


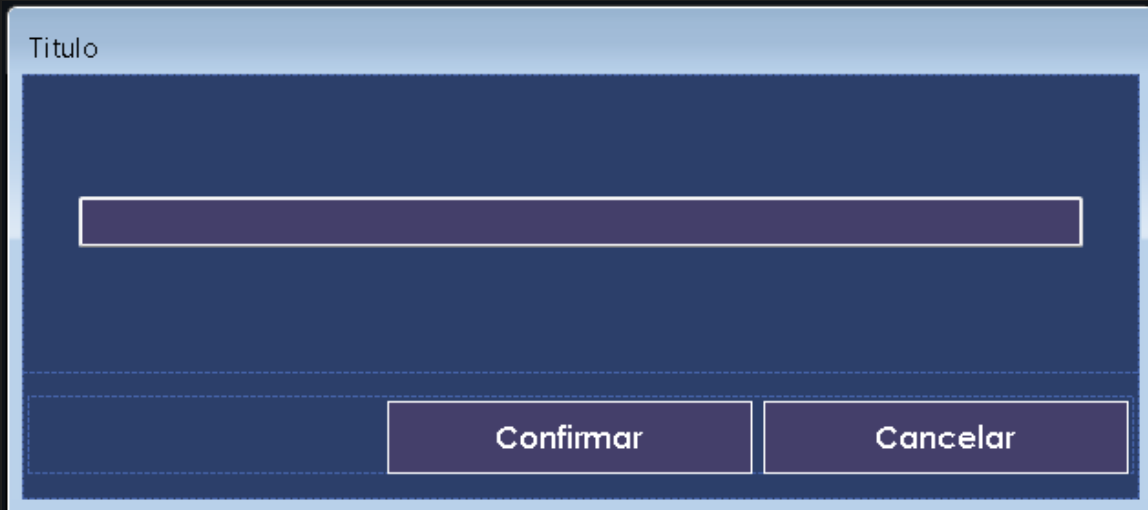
102. La lista del super

El objetivo será crear una aplicación que maneje una lista de supermercado.

Se podrá agregar, eliminar y modificar los elementos de la lista.



Crear un proyecto de Windows Forms con un formulario llamado FrmAltaModificacion que se vea como el siguiente:

The image shows a screenshot of a Windows Forms application window. The window has a title bar with the text "Titulo". Inside the window, there is a single-line text input field. Below the input field, there are two buttons: "Confirmar" and "Cancelar". The window has a standard Windows XP-style border.

- Deberá iniciar centrado en la pantalla (propiedad StartPosition).
- No deberá tener menú de control (propiedad ControlBox).
- No deberá poder cambiar de tamaño (propiedad FormBorderStyle).
- No tendrá ícono (propiedad ShowIcon).
- Tendrá un TextBox llamado txtObjeto, un Button llamado btnConfirmar y un Button llamado btnCancelar.
- El txtObjeto no deberá poder contener un texto de más de 50 caracteres (propiedad MaxLength).
- Tendrá una propiedad Objeto pública y de sólo lectura que retornará el contenido del txtObjeto.
- Al instanciarse deberá configurarse de acuerdo a la siguiente información que recibirá como argumentos en su constructor:
 - El título del formulario.
 - El texto que contendrá el txtObjeto.
 - El texto del btnConfirmar.
- Si se presiona el btnConfirmar o la tecla enter (posicionados dentro del TextBox):
 - Validar que el txtObjeto no se encuentre vacío.
 - Si se encuentra vacío mostrar una advertencia y no avanzar.
 - Si es válido, cargar la propiedad DialogResult con el valor DialogResult.OK y cerrar el formulario.

TIP

El evento `KeyPress` se acciona cuando el usuario está haciendo foco en el control que es dueño del evento y presiona una tecla.

Uno de los parámetros de entrada del manejador de dicho evento es de tipo `KeyPressEventArgs` y contiene una propiedad `KeyChar` que tendrá como valor el carácter correspondiente a la tecla presionada por el usuario.

Por ejemplo, para saber si el usuario presionó la tecla enter podemos hacer lo siguiente:

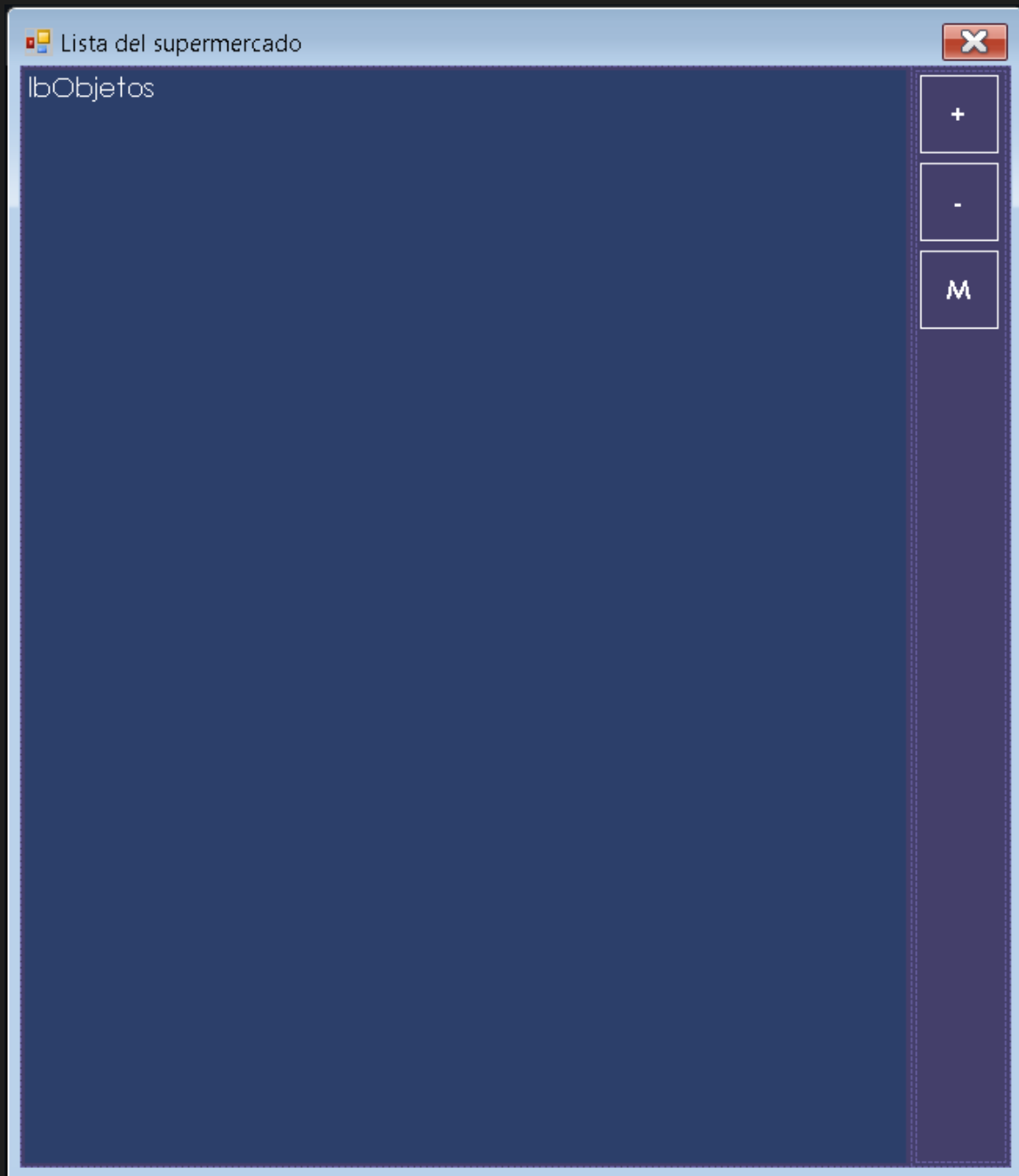
```
if (e.KeyChar == (char)13) // 13 es el código ASCII que representa a ENTER.
{
    // ...
}
```

Si queremos saber si el usuario presionó `s` o `S` (`shift + s`) podemos hacer lo siguiente:

```
if (e.KeyChar == 's' || e.KeyChar == 'S')
{
    // ...
}
```

- Si se presiona el `btnCancelar` o la tecla `escape` (posicionados dentro del `TextBox`), se deberá cargar la propiedad `DialogResult` con el valor `DialogResult.Cancel` y cerrar el formulario.

Agregar otro formulario llamado FrmListaSuper que se vea como el siguiente:



- Deberá ser el formulario que se muestra al iniciar la aplicación.
- Tendrá un atributo listaSupermercado de tipo List que contendrá los elementos de la lista del supermercado.
- Deberá iniciar centrado en la pantalla (propiedad StartPosition).

- Tendrá un ListBox llamado lstObjetos.
 - Mostrará la lista de objetos pendientes de comprar en el supermercado.
 - Estará anclado a todos los bordes del formulario (propiedad Anchor).
- Tendrá un Button llamado btnAgregar cuyo texto será "+" y al pasar por encima deberá mostrar un tooltip con el texto "Agregar objeto".
 - Deberá estar anclado arriba y a la derecha (propiedad Anchor).
 - Al accionarlo instanciará un FrmAltaModificacion y lo mostrará de forma modal.
 - El título será "Agregar objeto".
 - El contenido del txtObjeto será un texto vacío.
 - El texto del btnConfirmar será "Agregar".
 - Si la propiedad DialogResult de la instancia de FrmAltaModificacion vale DialogResult.OK, agregar el elemento retornado por la propiedad Objeto a la lista del supermercado. Caso contrario, no hacer nada.
- Tendrá un Button llamado btnEliminar cuyo texto será "-" y al pasar por encima deberá mostrar un tooltip con el texto "Eliminar objeto".
 - Deberá estar anclado arriba y a la derecha (propiedad Anchor).
 - Al accionarlo deberá borrar el objeto seleccionado en la lista. Si no hay nada seleccionado, no hacer nada y mostrar un cartel informando que se debe seleccionar un elemento de la lista.
- Tendrá un Button llamado btnModificar cuyo texto será "M" y al pasar por encima deberá mostrar un tooltip con el texto "Modificar objeto".
 - Deberá estar anclado arriba y a la derecha (propiedad Anchor).
 - Al accionarlo instanciará un FrmAltaModificacion y lo mostrará de forma modal.
 - El título será "Modificar objeto".
 - El contenido del txtObjeto será el elemento seleccionado en lstObjetos. Si no hay nada seleccionado, no hacer nada y mostrar un cartel informando que se debe seleccionar un elemento de la lista.
 - El texto del btnConfirmar será "Modificar".
 - Si la propiedad DialogResult de la instancia de FrmAltaModificacion vale DialogResult.OK, modificar el objeto en

la lista del supermercado asignándole el valor de la propiedad Objeto. Caso contrario, no hacer nada.

- En el manejador del evento Load se deberá buscar el archivo listaSupermercado.xml en la carpeta de datos de aplicaciones (Environment.SpecialFolder.ApplicationData) y, si existe, deserializarla desde formato XML como una lista de string.
 - Cargar lstObjetos con los elementos de la lista.
 - Si el archivo no existe, no hacer nada.
- Ante cualquier acción se deberá actualizar el contenido de lstObjetos y el archivo listaSupermercado.xml, que contendrá la lista de objetos serializada a formato XML y se encontrará en la ubicación antes nombrada.

TIP

Para actualizar el contenido de un ListBox se debe cargar la propiedad DataSource como null y luego asignarle a la misma la colección de elementos:

```
listBox.DataSource = null;
```

```
listBox.DataSource = listaSupermercado;
```

IMPORTANTE

Reutilizar código en todo el proyecto siguiendo el principio DRY.