

**Controles de Windows Forms**  
**El control CheckedListBox**  
**y ListBox**

**Autor:**

**Jose Piniella Martinez**

---

# 1. Los controles **CheckedListBox** y **ListBox**

## 1.1. ¿En qué consisten?

Las funcionalidades de estos dos controles son muy parecidas y coinciden en muchas propiedades y eventos.

Un control **CheckedListBox** consiste en una lista que tiene un control checkbox en la parte izquierda de cada elemento, indicando si este está seleccionado o no, por su parte el **ListBox** consiste en presentar una lista que el usuario puede seleccionar pulsando con el ratón o el teclado, tiene dos modos de selección: simple o múltiple y una o varias columnas.

## 1.2. Constructores para **CheckedListBox** y **ListBox**

El constructor de **CheckedListBox** se presenta vacío:

```
//public CheckedListBox()
```

Tenemos tres posibles constructores que controlan **ListBox** son:

```
public ObjectCollection(
```

```
    ListBox owner // 1. ListBox que posee la colección
```

```
    Object[] value // 2. Matriz de objetos que se va a agregar a la colección
```

```
    ListBox.ObjectCollection value // 3. Colección cuyo contenido se copia en esta colección.
```

```
)
```

## 1.3. Selección de Propiedades de **CheckedListBox** y **ListBox**

Podemos señalar las siguientes propiedades más comunes de estos dos controles:

- a) **Item**. De tipo **int**, contiene todos los elementos de la lista, se utiliza para insertar, añadir o eliminar elementos de la lista.
- b) **Sorted**. De tipo **bool**, que devuelve si esta ordenada por orden alfabético.
- c) **AllowDrop**. De tipo **bool**, establece un valor que indica si el control puede aceptar los datos que el usuario arrastra al mismo.

---

### **CheckedListBox:**

- d) CheckedIndices.** Colección que contiene los elementos de la lista que están en estado Checked o indeterminado.
- e) CheckedItems.** Colección de los elementos de la lista que están en estado Checked o indeterminado.
- f) CheckedOnClick.** De tipo **bool**, indicara si un item cambia de estado cuando el usuario pulse el ratón sobre el.

### **ListBox:**

- g) MultiColumn.** De tipo **bool**, que define el número de columnas de la lista.
- h) DrawMode.** establece el modo de dibujo del control.
- i) ItemHeight.** De tipo **int** establece el alto de un elemento de ListBox.

## **1.4. Selección de Métodos de ChekedListBox y ListBox**

Los métodos más comunes de ambos controles son:

- a) Add(int n).** Agrega un elemento a la lista de elementos de la colección.
- b) Clear().** Quita todos los elementos de la colección.

### **CheckedListBox:**

- c) GetItemChecked(int n).** Devuelve un valor booleano indicando si el item n esta seleccionado o no.
- d) GetItemCheckState(int n).** Devuelve un valor indicando el estado del item n.

### **ListBox:**

- e) Findstring(string s).** Busca el primer elemento del control ListBox que comience por la cadena especificada.
- f) SetSelected(int n bool v).** Activa o desactiva la selección del elemento especificado de un control ListBox.

## **1.5. Selección de Eventos de ChekedListBox y ListBox**

Los eventos más comunes de y usados de ambos controles son:

- a) Click().** Se produce cuando el usuario hace clic en el control CheckedListBox o ListBox.

- 
- b) **MouseClicked()**. Se produce cuando el usuario hace clic en el control `CheckedListBox` o `ListBox` con el ratón.

**CheckedListBox:**

- c) **ItemCheck()**. Se produce cuando el estado chequeado de un ítem cambia.

**ListBox:**

- d) **DrawItem()**. Se produce cuando cambia la apariencia de un control `ListBox` dibujado por el propietario.

## 1.6. Observaciones a tener en cuenta sobre `CheckedListBox` y `ListBox`

**CheckedListBox:**

- Este control presenta una lista de elementos que el usuario pueda navegar usando el teclado o la barra de desplazamiento en el lado derecho del control. El usuario puede colocar una marca de verificación por uno o más elementos y los elementos seleccionados se puede navegar con el [CheckedListBox.CheckedItemCollection](#) y [CheckedListBox.CheckedIndexCollection](#).

**ListBox:**

- Hay varias maneras de agregar elementos a la colección. El método [Add](#) permite agregar un solo objeto a la colección. Para agregar varios objetos a la colección, cree una matriz de elementos y asígnela al método [AddRange](#).
- También se pueden manipular los elementos de un control `ListBox` mediante la propiedad [DataSource](#). Si utiliza la propiedad [DataSource](#) para agregar elementos a un control `ListBox`, podrá ver los elementos del control `ListBox` mediante la propiedad `Items` pero no podrá agregar ni quitar elementos de la lista mediante los métodos de [ListBox.ObjectCollection](#).
- El método [Contains](#) permite determinar si un objeto es un miembro de la colección. Cuando sepa que el elemento está ubicado en la colección, podrá utilizar el método [IndexOf](#) para determinar dónde está ubicado el elemento en la colección.

---

## 1.7. Documentación de referencia

- [https://msdn.microsoft.com/es-es/library/system.windows.forms.checkedlistbox\(v=vs.110\).aspx](https://msdn.microsoft.com/es-es/library/system.windows.forms.checkedlistbox(v=vs.110).aspx)  
(CheckedListBox)
- [https://msdn.microsoft.com/es-es/library/System.Windows.Forms.ListBox.ObjectCollection\(v=vs.110\).aspx](https://msdn.microsoft.com/es-es/library/System.Windows.Forms.ListBox.ObjectCollection(v=vs.110).aspx)  
(ListBox)