

Contenido

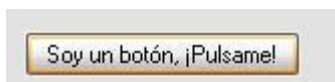
Los controles de Windows Forms	2
CommandButton	2
Command Button (System.Windows.Forms.Button)	2
Check Box	2
Check Box (System.Windows.Forms.CheckBox)	2
Label	3
Label (System.Windows.Forms.Label)	3
Link Label.....	3
Link Label (System.Windows.Forms.Label)	3
List Box	3
List Box (System.Windows.Forms.ListBox)	3
Text Box.....	4
Text Box (System.Windows.Forms.TextBox)	4
Rich Text Box	4
Rich Text Box (System.Windows.Forms.RichTextBox)	4
Progress Bar	4
Progress Bar (System.Windows.Forms.ProgressBar)	4
Radio Button	5
Radio Button (System.Windows.Forms.RadioButton)	5
GroupBox.....	5
Group Box (System.Windows.Forms.GroupBox)	5
Panel.....	6
Panel (System.Windows.Forms.Panel)	6
Tab Control.....	6
Tab Control (System.Windows.Forms.TabControl)	6
Tool Strip	6
Tool Strip (System.Windows.Form.ToolStrip) y StatusStrip (System.Windows.Form.StatusStrip)	6
Save File Dialog.....	7
Save File Dialog (System.Windows.Forms.SaveFileDialog) y OpenFileDialog (System.Windows.Forms.OpenFileDialog)	7

Los controles de Windows Forms

Representan toda la interfaz gráfica de nuestro programa. Son muy importantes, pues dotan a nuestras aplicaciones de una intuitividad y una facilidad de uso inmensa, es por ello que Windows triunfó en su día. No es lo mismo escribir comandos en una consola, que pulsar un botón y que haga todo eso por tí.

Los controles estandar los conocemos todos, pues los hemos estado usando durante toda la vida. El botón de pulsar, las cajas de texto, las cajas para marcar o los botones circulares que te permiten conmutar entre diferentes opciones, los deslizadores, la clásica barra de progreso, etc. Hay una completa infinidad de controles de todo tipo, y si no encontramos el que necesitamos, ya os explicaré como crear uno vosotros mismos. De momento, vamos a hechar un vistazo a ver como funcionan los controles más básicos.

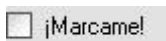
CommandButton



Command Button (System.Windows.Forms.Button)

Es un botón. Ni más, ni menos. Cuando lo pulsamos, ocurrirá algo. Podemos modificar su aspecto, incluyendo el texto e incluso añadir imagen de fondo. Su principal evento es Click, el cual se ejecuta al pulsar el botón.

Check Box

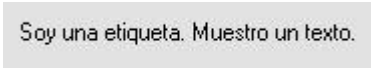


Check Box (System.Windows.Forms.CheckBox)

Es la típica casilla para marcar. Al pulsar sobre él, se cambiara su estado entre marcado, y desmarcado. Además del evento Click, incluye un evento CheckedChange, el cual se ejecuta al cambiar entre marcado y desmarcado, y CheckStatusChange, que se ejecuta mientras se está cambiando entre marcado y desmarcado. Incluye además una propiedad llamada "CheckState", que es un booleano que comprueba o establece si la casilla está marcada (true) o desmarcada (false).

```
bool System.Windows.Forms.CheckBox.CheckedState
```

Label




Soy una etiqueta. Muestro un texto.

Label (System.Windows.Forms.Label)

Es una etiqueta que muestra un texto. Podemos modificar la fuente (Font) y el texto (Text). Por lo general es puramente informativo, pero incluye los métodos típicos de todos los controles, incluido Click.

Link Label

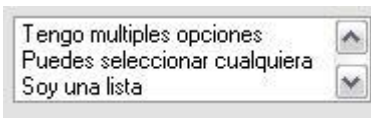


[Soy un enlace a Elektron Blog](#)

Link Label (System.Windows.Forms.LinkLabel)

Similar a Label, pero con la peculiaridad de tener el aspecto y comportamiento de un hipervínculo. Podemos generar un evento Click, en el cual probablemente queramos cargar una página web en el navegador.

List Box

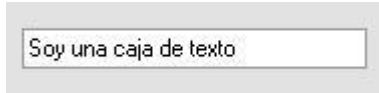


List Box (System.Windows.Forms.ListBox)

Muestra una lista con múltiples elementos. Los elementos se almacenan en una matriz de tipo `ObjectCollection` llamada “Items”. Dentro de Items tenemos una función `Add(object)` para añadir más objetos y `Remove(object)` para borrarlo. También podemos acceder al índice del objeto seleccionado mediante la propiedad `SelectedIndex`.

```
listBox.Items.Add(object);  
listBox.Items.Remove(object);  
int listBox.SelectedIndex;
```

Text Box



Text Box (System.Windows.Forms.TextBox)

Es una caja de texto, donde el usuario puede introducir un texto, recogido en la variable Text de tipo String.

Rich Text Box



Rich Text Box (System.Windows.Forms.RichTextBox)

A diferencia de la caja de texto, Rich Text Box es como un pequeño editor de texto de formato enriquecido, algo así como Wordpad. Lo que significa poder dar formato al texto, incluyendo escribir en diferentes fuentes, cambiar los colores e incluso incrustar objetos. Tiene una propiedad llamada Lines, que es una matriz de cadenas de texto string, que representa el texto. También tiene dos métodos llamados SaveFile y LoadFile, que permiten cargar y guardar automáticamente el texto, manteniendo su formato (Se debe guardar con la extensión RTF).

Progress Bar



Progress Bar (System.Windows.Forms.ProgressBar)

Todos conocemos la barra de progreso de Windows. Indica un progreso, o una capacidad. La propiedad Value indica la cantidad de progreso en números enteros (int). Max indica el valor máximo, entre 1 y 2,147,483,647. Step es otro entero que indica cada cuanto aparecerá una barra de color indicando progreso. Por lo general deberá ser una décima parte de Max.

Una propiedad muy interesante del Progress Bar es Style, la cual define el estilo del control. Usamos *Blocks* o *Continuous* para representar un progreso o capacidad, pero en ocasiones se usa *Marquee*, que indica al usuario que se está realizando una operación en segundo plano y que debe esperar. En caso de usar Marquee, usamos la propiedad MarqueeAnimationSpeed para indicar la velocidad de la animación.

Una buena idea sería empezar con un valor bajo en MarqueeAnimationSpeed, e ir acelerándolo a medida que vamos progresando.

Radio Button



Radio Button (System.Windows.Forms.RadioButton)

Son unos botones especiales, similares al CheckBox, pero con la peculiaridad de que *solo puede haber uno activo al mismo tiempo*. Así que lo lógico sería colocarlos sobre *paneles o cajas de grupo*, si vamos a colocar varios grupos de éstos. Tienen también una propiedad Checked para comprobar o establecer el estado de cada uno.

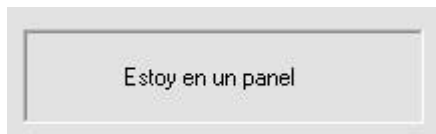
GroupBox



Group Box (System.Windows.Forms.GroupBox)

Seguro que hemos ido alguna vez a las opciones de algun programa y hemos visto unas cajitas con un texto. Éstas cajas son cajas de grupo. Son como contenedores de controles, pudiendo meter más controles dentro del mismo. Debido a que los controles que hay dentro están a un nivel distinto del formulario, podemos agrupar radio buttons dentro.

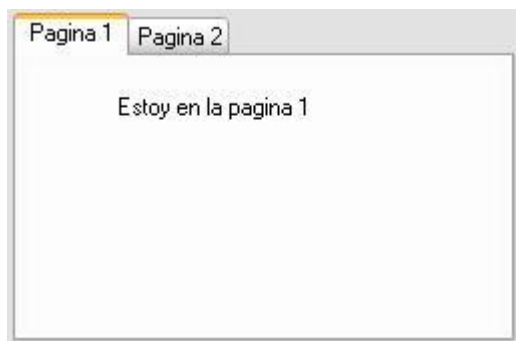
Panel



Panel (System.Windows.Forms.Panel)

Los paneles son lo mismo que los groupbox. Contenedores de controles. Lo único que cambia es la estética. Por defecto, son invisibles. Por eso son la mejor opción a la hora de crear grupos de radio buttons.

Tab Control



Tab Control (System.Windows.Forms.TabControl)

Otro control muy común que podemos ver en apartados de configuración. Los Tab Control son también contenedores, pero clasifican el contenido por páginas (TabPage). Dentro del TabControl hay una colección llamada TabPages que contiene todas las páginas. Cada página tiene además una colección de controles, y una propiedad Text que representa el texto de la página.

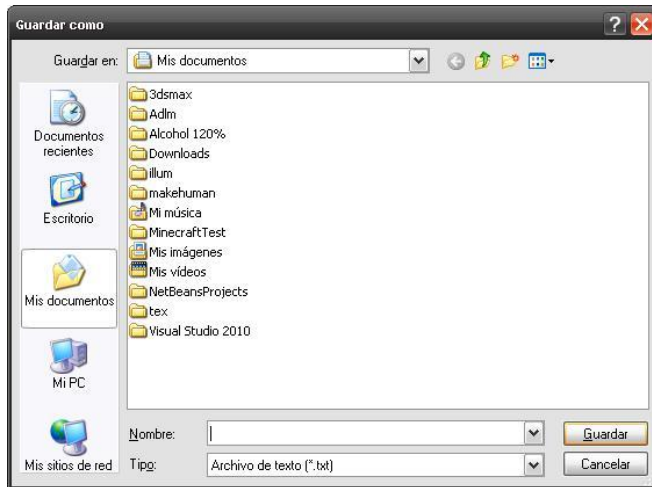
Tool Strip



Tool Strip (System.Windows.Forms.ToolStrip) y StatusStrip (System.Windows.Forms.StatusStrip)

Ambos son lo mismo, con la diferencia de que el ToolStrip va anclado justo en la parte de arriba del formulario, y el StatusStrip en la de abajo. Son menús que pueden contener submenús, botones o incluso barras de progreso o etiquetas Label.

Save File Dialog



Save File Dialog (System.Windows.Forms.SaveFileDialog) y OpenFileDialog (System.Windows.Forms.OpenFileDialog)

La típica pantalla que aparece al guardar o cargar un archivo. Son controles invisibles en los que se especifica una ruta a un archivo. La única diferencia es en el texto del botón Cargar o Guardar.

La propiedad Filter nos permite establecer un filtro, en formato string, que especifica la extensión, o extensiones, que mostrará la ventana del explorador. La ruta del archivo se encuentra en la propiedad FileName, que es una cadena de texto.

Para hacer aparecer el dialogo, tenemos que ejecutar la función ShowDialog().

```
openFileDialog1.ShowDialog();
```

Hay muchos más tipos de controles interesantes que ya iremos investigando en futuras entradas. Como selectores de color, gráficas, acceso a bases de datos e incluso controles no nativos de .net.

Por ahora éstos controles son los más básicos y los que más se van a usar. Sé que lo aquí explicado es muy escueto, pero de momento es todo lo que me puedo permitir poner en una entrada.