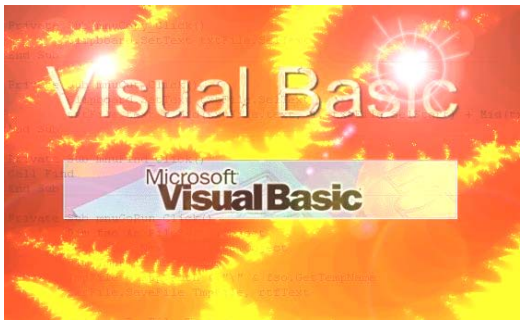
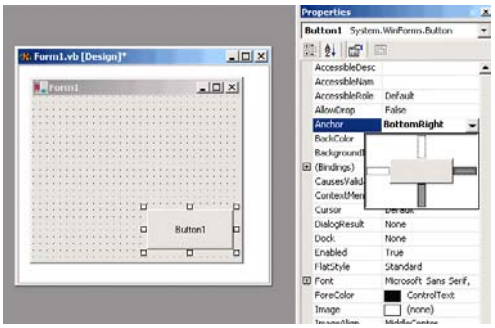
CURSO

Curso Completo de Visual Basic 6.0





Escuela Superior de Ingenieros Industriales
UNIVERSIDAD DE NAVARRA
Javier García de Jalón · José Ignacio Rodríguez
Alfonso Brazález · Patxi Funes · Eduardo Carrasco · Jesús Calleja

4.2 PROPIEDADES MÁS COMUNES

Hay algunas propiedades que son comunes a muchos controles. A continuación se hace una lista con las utilizadas más habitualmente:

- ***Appearance***: Establece si un objeto tiene un aspecto plano (valor 0) o tridimensional (valor 1).

- ***BackColor***: Establece el color de fondo de un objeto.

- ***Caption***: Establece el texto que aparece dentro o junto al objeto. Tiene el papel de un título.
- ***Enabled***: Establece si un objeto es accesible y modificable o no.

- ***Font***: Establece las características del tipo de letra del objeto.

- ***ForeColor***: Establece el color del texto y/o gráficos de un objeto.

- ***Height***: Establece la altura de un objeto.

- ***Left***: Establece la distancia horizontal entre el origen del control y el origen del objeto que lo contiene, que puede ser un formulario, un marco (frame), etc.

- ***MousePointer:*** Establece la forma que adoptará el puntero del ratón al posicionarse sobre el objeto. Esta forma puede elegirse dentro de una lista en las que aparecen las habituales del puntero del ratón o creando iconos propios.

- ***Name***: Nombre del objeto. Todos los objetos incluidos en un formulario deben tener un nombre con el que poder referirse a él a la hora de programar la forma en que debe actuar. Existen unas reglas para definir los nombre de los controles, que ya se vieron en el **Capítulo 1.**

- ***Top***: Establece la distancia vertical entre el origen del control y el origen del objeto que lo contiene.

- ***Visible***: Establece si el objeto es visible o invisible.

- ***Width***: Establece la anchura del objeto.

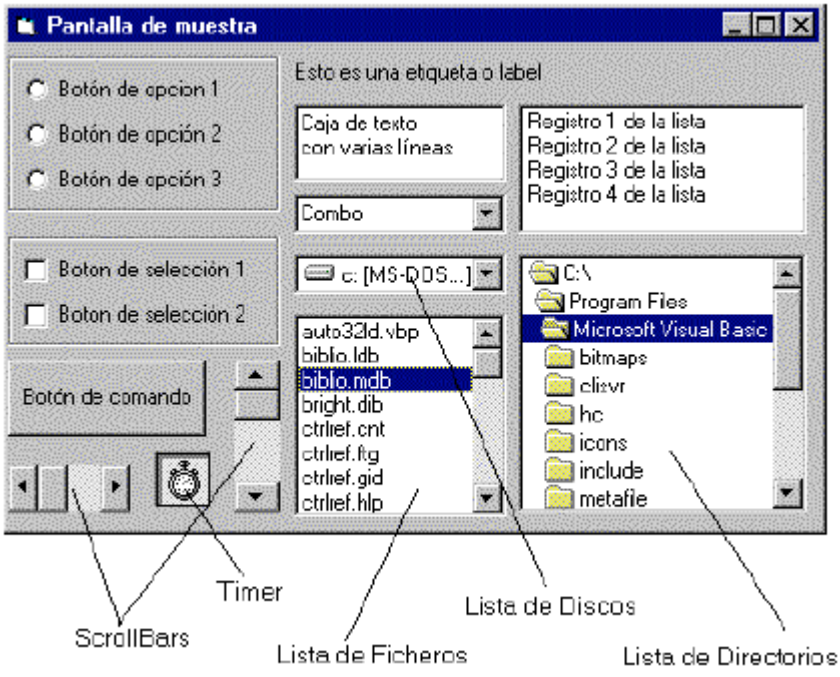


Figura 4.3. Algunos de los controles más habituales de ***Visual Basic***.

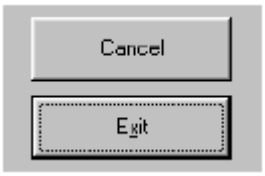
4.3 CONTROLES MÁS USUALES

En la Figura 4.3 se muestran algunos de los controles más habituales en ***Visual Basic 6.0***. Estos controles se explican a continuación con más detalle.

4.3.1 Botón de comando (Command Button)

La propiedades más importantes del botón de comando son su ***Caption***, que es lo que aparece escrito en él, las referentes a su posición (***Left*** y ***Top***) y apariencia externa (***Height***, ***Width*** y tipo de letra) y la propiedad ***Enabled***, que determina si en un momento dado puede ser pulsado o no.

No hay que confundir la propiedad ***Caption*** con la propiedad ***Name***. La primera define a un texto que aparecerá escrito en el control, mientras que las segunda define el nombre interno con el que se puede hacer referencia al citado objeto.

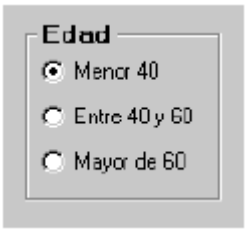


Si en la propiedad ***Caption*** se pone el carácter (&) antes de una de sus letras, dicha letra aparece subrayyada en el botón (como la "x" en el botón ***Exit*** de la figura anexa). Esto quiere decir que, como es habitual en ***Windows***, dicho botón puede activarse con el teclado por medio de la combinación ***Alt+letra subrayada***. Esta característica es común a muchos de los controles que tienen propiedad ***Caption***.

El evento que siempre suelen tener programado los botones de comandos es el evento ***Click***.

4.3.2 Botones de opción (Option Button)

Además de las mencionadas para el caso anterior estos botones tienen la propiedad ***Value*** que sólo puede ser ***True*** en uno de los botones del grupo.
Para agrupar botones se coloca primero un ***marco*** o ***frame*** en el formulario y, estando seleccionado, se colocan después cuantos botones de opción se desee. En un mismo formulario se pueden colocar cuantos grupos de botones de opción se quiera, cada uno de ellos agrupado dentro de su propio marco. Es muy importante colocar primero el ***frame*** y después los botones de opción. Con esto se consigue que los botones de opción estén agrupados, de modo que sólo uno de ellos pueda estar activado. Si no se coloca ningún ***frame*** todos los botones de opción de un mismo formulario forman un único grupo. Si los botones ya existen y se quieren introducir un un ***frame*** se seleccionan, se hace ***Cut*** y luego ***Paste*** dentro del ***frame***.



Sólo un grupo de botones de opción puede recibir el ***focus***, no cada botón por separado. Cuando el grupo tiene el ***focus***, con las flechas del teclado (↑ y ↓) se puede activar una u otra opción sin necesidad de usar el ratón. También se puede utilizar ***Alt+carácter*** introduciendo antes de dicho carácter un (&) en el ***Caption*** del botón de opción.

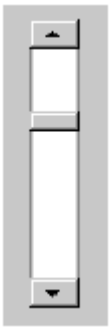
4.3.3 Botones de selección (Check Box)

La única diferencia entre estos botones y los anteriores es que en los botones de selección puede haber más de uno con la propiedad ***Value*** a ***True***. Estos botones no forman grupo aunque estén dentro de un ***frame***, y reciben el ***focus*** individualmente. Se puede también utilizar el carácter (&) en el ***Caption*** para activarlos con el teclado.

El usuario debe decidir qué tipo de botones se ajustan mejor a sus necesidades: en el caso de la edad, está claro que no se puede ser de dos edades diferentes; sí es posible sin embargo conocer varios lenguajes de programación.

4.3.4 Barras de desplazamiento (Scroll Bars)

En este tipo de control las propiedades más importantes son ***Max*** y ***Min***, que determinan el rango en el que está incluido su valor, ***LargeChange*** y ***SmallChange*** que determinan lo que se modifica su valor al clicar en la barra o en el botón con la flecha respectivamente y ***Value*** que determina el valor actual de la barra de desplazamiento. Las barras de desplazamiento no tienen propiedad ***Caption***.



El evento que se programa habitualmente es ***Change,*** que se activa cuando la barra de desplazamiento modifica su valor. Todo lo comentado en este apartado es común para las barras de desplazamiento verticales y horizontales.
Además de las ***Scroll Bars*** horizontal y vertical, ***Visual Basic 6.0*** dispone también del control ***Slider***, utilizado en los paneles de control de ***Windows***, que tiene una función similar.

4.3.5 Etiquetas (Labels)

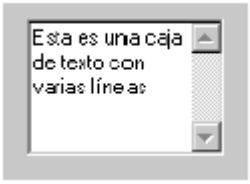
En las etiquetas o labels la propiedad más importante es ***Caption***, que contiene el texto que aparece sobre este control. Esta propiedad puede ser modificada desde programa, pero no interactivamente clicando sobre ella (a diferencia de las ***cajas de texto***, que se verán a continuación). Puede controlarse su tamaño, posición, color de fondo y una especie de borde 3-D. Habitualmente las ***labels*** no suelen recibir eventos ni contener código.

Las ***Labels*** tienen las propiedades ***AutoSize*** y ***WordWrap***. La primera, cuando está a ***True***, ajusta el tamaño del control al del texto en él contenido. La segunda hace que el texto se distribuya en varias líneas cuando no cabe en una sola.

4.3.6 Cajas de texto (Text Box)

La propiedad más importante de las cajas de texto es ***Text***, que almacena el texto contenido en ellas. También se suelen controlar las que hacen referencia a su tamaño, posición y apariencia. En algún momento se puede desear impedir el acceso a la caja de texto, por lo que se establecerá su propiedad ***Enabled*** como ***False***. La propiedad ***Locked*** como ***True*** hace que la caja de texto sea de sólo lectura. La propiedad ***MultiLine***, que sólo se aplica a las cajas de texto, determina si en una de ellas se pueden incluir más de una línea o si se ignoran los saltos de línea.

La justificación o centrado del texto se controla con la propiedad ***Alignment***. La propiedad ***ScrollBars*** permite controlar el que aparezca ninguna, una o las dos barras de desplazamiento de la caja.



En una caja de texto no se pueden introducir ***Intros*** con el teclado en modo de diseño. En modo de ejecución se deben introducir como caracteres ASCII (el 13 seguido del 10, *esto Carriage Return* y *Line Feed*). Afortunadamente ***Visual Basic 6.0*** dispone de la constante ***vbCrLf***, que realiza esta misión de modo automático.

Otras propiedades importantes hacen referencia a la selección de texto dentro de la caja, que sólo están disponibles en tiempo de ejecución. La propiedad ***SelStart*** sirve para posicionar el cursor al comienzo del texto que se desea seleccionar (el primer carácter es el cero); ***SelLength*** indica el número de caracteres o longitud de la selección; ***SelText*** es una cadena de caracteres que representa el texto seleccionado. Para hacer ***Paste*** con otro texto sustituyendo al seleccionado basta asignarle a esta propiedad ese otro texto (Si no hay ningún texto seleccionado, el texto de ***SelText*** se inserta en la posición del cursor); para entresacar el texto seleccionado basta utilizar esta propiedad en alguna expresión.

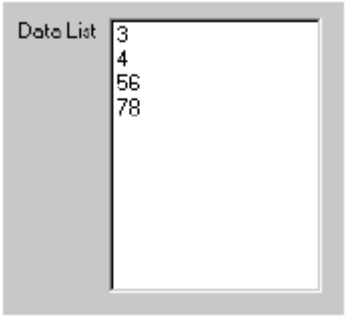
Los eventos que se programan son ***Change***, cuando se quiere realizar alguna acción al modificar el contenido de la caja, ***Click*** y ***DblClick*** y en algunos casos especiales ***KeyPress*** para controlar los caracteres que se introducen. Por ejemplo, se puede chequear la introducción del código ASCII 13 (***Intro***) para detectar que ya se finalizado con la introducción de datos. También se utiliza la propiedad ***MaxLength*** para determinar el número máximo de caracteres que pueden introducirse en la caja de texto.

En aquellos casos en los que se utilice una caja de texto como ***entrada de datos*** (es el control que se utiliza la mayoría de las veces con esta finalidad), puede ser interesante utilizar el método ***SetFocus*** para enviar el foco a la caja cuando se considere oportuno.

Otras propiedades de las cajas de texto hacen referencia a los tipos de letra y al estilo. Así la propiedad ***FontName*** es una cadena que contiene el nombre del *Font* (***Courier New***, ***Times New Roman***, etc.), ***FontSize*** es un tipo ***Short*** que contiene el tamaño de la letra, y ***FontBold***, ***FontItalic***, ***FontUnderline*** y ***FontStrikethrough*** son propiedades tipo ***Boolean*** que indican si el texto va a tener esa característica o no.

4.3.7 Listas (List Box)

Una ***lista*** es un control en el que se pueden mostrar varios registros o líneas, teniendo uno o varios de ellos seleccionado(s). Si en la lista hay más registros de los que se pueden mostrar al mismo tiempo, se añade automáticamente una ***scrollBar***.



Para añadir o eliminar registros de la lista se utilizan los métodos ***AddItem*** y ***RemoveItem*** respectivamente. Esto sólo puede hacerse en modo de ejecución, y no en modo de diseño. Se suelen inicializar desde el evento ***Form_Load***. La propiedad ***List*** es un array que permite definir el contenido de la lista en modo de diseño a través de la ventana de propiedades y también acceder a los elementos de la lista en tiempo de ejecución, para utilizar y para cambiar su valor. Para ello se pone en índice del elemento entre paréntesis (empezando a contar por cero) a continuación de ***List***, como se muestra a continuación por ejemplo, para cambiar el tercer elemento:

```
lstName.List(2) = "Tercero"
```

Para añadir un registro en tiempo de ejecución se utiliza **AddItem:**

```
lstName.AddItem Registro_Añadido, posicion
```

donde *posicion* es un argumento opcional que permite especificar la posición en que se debe añadir.
Si se omite el registro se añade al final de la lista. Lo anterior es válido si la propiedad ***Sorted*** está a ***False***; si está a ***True*** el nuevo registro se añade en la posición ordenada que le corresponde. Para eliminar un registro,

```
lstName.RemoveItem Posición_del_registro_en_la_lista
```

En el caso de que se quiera vaciar completamente el contenido de una lista se puede utilizar el método ***Clear***.

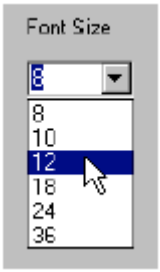
Dos propiedades interesantes de las listas son ***ListCount*** y ***ListIndex***. La primera contiene el número total de registros incluidos en la lista. La segunda permite acceder a una posición concreta de la lista para añadir un registro nuevo en esa posición, borrar uno ya existente, seleccionarlo, etc.

Hay que tener en cuenta que los elementos de la lista se empiezan a numerar por cero. El valor de propiedad ***ListIndex*** en cada momento coincide con el registro seleccionado y en el caso de no haber ninguno seleccionado esta propiedad vale -1.

Es interesante saber que al seleccionar uno de los registros de la lista se activa el evento ***Click*** de dicha lista.

4.3.8 Cajas combinadas (Combo Box)

Un ***combo*** tiene muchas cosas en común con una ***lista***. Por ejemplo los métodos ***AddItem***, ***RemoveItem*** o ***Clear*** y las propiedades ***List***, ***ListIndex*** o ***ListCount***.



La diferencia principal es que en un ***combo*** tiene una propiedad llamada ***Style***, que puede adoptar tres valores (***1,2*** ó ***3***) que corresponden con tres distintas formas de presentar una lista:

1. ***Style***=0 (***Dropdown Combo***), Éste es el valor más habitual y corresponde con el caso en el que sólo se muestra el registro seleccionado, que es editable por el usuario, permaneciendo el resto oculto hasta que el usuario despliega la lista completa clicando sobre el botón-flecha.

2. ***Style=1*** (***Simple Combo***). En este caso el registro seleccionado también es editable, y se muestra una lista no desplegable dotada si es necesario de una ***scrollbar***.

3. ***Style=2*** (***DropDown List***). En este último caso el registro seleccionado no es editable y la lista es desplegable.

En una ***caja combinada***, al igual que en una ***caja de texto*** sencilla, está permitido escribir con el teclado en tiempo de ejecución, si la propiedad ***Enabled*** vale ***True***. En una ***lista*** esto no es posible.

4.3.9 Controles relacionados con ficheros

Trabajando en un entorno ***Windows 95/98/NT/ XP/Vista*** es habitual tener que abrir y cerrar ficheros parar leer datos, guardar un documento, etc. Hay tres controles básicos que resultan de suma utilidad en esta tarea. Son la lista de unidades lógicas o discos (***Drive ListBox***), la lista de directorios (***Dir ListBox***) y la lista de ficheros (***File ListBox***). Estos controles se tratan con más detalle en el **Capítulo 7.**

4.3.10 Control Timer

Si se desea que una acción suceda con cierta periodicidad se puede utilizar un control ***Timer***.
Este control produce de modo automático un evento cada cierto número de milisegundos. La propiedad más importante de un objeto de este tipo es ***Interval***, que determina, precisamente, el intervalo en milisegundos entre eventos consecutivos. La acción que se desea activar debe programarse en el evento ***Timer*** de ese mismo control.

Si en algún momento se desea anular momentáneamente la acción periódica es suficiente con hacer ***False*** la propiedad ***Enabled*** del control ***Timer*** y para ejecutarla de nuevo volver a hacer ***True*** esa propiedad. Haciendo 0 la propiedad ***Interval*** también se consigue inhabilitar el ***Timer***.

Continuará.....

Nota de Radacción: El lector puede descargar este capítulo y capítulos anteriores del curso desde la sección ***"Artículos Técnicos"*** en el sitio web de **EduDevices (www.edudevices.com.ar)**