





















**Controles comunes de Visual Basic**

Los controles comunes de Visual Basic se encuentran dentro del archivo ejecutable de Visual Basic. Los controles intrínsecos están siempre incluidos en el cuadro de herramientas, no como los controles ActiveX y los objetos insertables, que se pueden quitar o agregar al cuadro de herramientas.

En la tabla siguiente se enumeran los controles intrínsecos del cuadro de herramientas de Visual Basic.

<b>Icono</b>	<b>Nombre del control</b>	<b>Nombre de clase</b>	<b>Descripción</b>
	Casilla de verificación	CheckBox	Presenta una opción de tipo Verdadero o Falso, o Sí o No. Puede activar varias casillas de verificación al mismo tiempo.
	Cuadro combinado	ComboBox	Combina un cuadro de texto y un cuadro de lista. Permite que el usuario escriba una selección o seleccione un elemento de la lista desplegable.
	Botón de comando	CommandButton	Ejecuta un comando o una acción cuando un usuario hace clic en él.
	Datos	Data	Permite conectar con una base de datos existente y presentar información de ella en formularios.
	Cuadro de lista de directorios	DirListBox	Presenta directorios y rutas de acceso, y permite que el usuario los seleccione.
	Cuadro de lista de unidades	DriveListBox	Presenta unidades de disco válidas y permite que el usuario las seleccione.
	Cuadro de lista de archivos	FileListBox	Presenta una lista de archivos y permite que el usuario los seleccione.
	Marco	Frame	Proporciona un contenedor visual y funcional para otros controles.
	Barras de desplazamiento horizontal y vertical	HScrollBar y VScrollBar	Permite que un usuario agregue barras de desplazamiento a controles que no las tienen de forma automática. (No son las barras de desplazamiento incorporadas que se incluyen en muchos controles.)
	Imagen	Image	Presenta mapas de bits, iconos o metarchivos de Windows, archivos JPEG o GIF, y actúa como un botón de comando cuando se hace clic en él.
	Etiqueta	Label	Presenta texto con el que el usuario no puede interactuar ni modificar.
	Línea	Line	Agrega un segmento de línea resta a un formulario.
	Cuadro de lista	ListBox	Presenta una lista de elementos entre los que el usuario puede elegir.

Icono	Nombre del control	Nombre de clase	Descripción
	Contenedor OLE	OLE	Incrusta datos en una aplicación de Visual Basic. Se utiliza para presentar y manipular datos de otras aplicaciones de Windows, como Microsoft Excel y Microsoft Word.
	Botón de opción	OptionButton	El control OptionButton, como parte de un grupo de opciones con otros botones de opción, presenta varias opciones entre las que el usuario sólo puede elegir una.
	Cuadro de imagen	PictureBox	Presenta mapas de bits, iconos o metarchivos de Windows, archivos JPEG o GIF. También presenta texto o actúa como contenedor visual para otros controles.
	Forma	Shape	Agrega un rectángulo, un cuadrado, una elipse o un círculo a un formulario, marco o cuadro de imagen.
	Cuadro de texto	TextBox	Proporciona un área para escribir o presentar texto.
	Cronómetro	Timer	Ejecuta eventos periódicos a intervalos de tiempo especificados.

La herramienta puntero  proporciona una forma de mover y cambiar el tamaño de los formularios y controles. No es un control. Durante los capítulos anteriores se ha trabajado con algunos de ellos, por lo que, a continuación se detallarán brevemente algunos.

## Label

Un control *Label* es un control gráfico que puede usar para mostrar texto que el usuario no podrá cambiar directamente.

Puede escribir código que cambie el texto mostrado por un control *Label* como respuesta a eventos en tiempo de ejecución. Por ejemplo, si la aplicación tarda unos minutos en realizar un cambio, puede mostrar un mensaje de estado del proceso en un control *Label*. También puede usar *Label* para identificar un control, como un control *TextBox*, que no tenga su propia propiedad *Caption*.

Establezca las propiedades *AutoSize* y *WordWrap* si desea que *Label* muestre líneas de longitud variable o números variables de líneas.

Eventos a los que responde el control Label:

Eventos	Descripción
<i>Change</i>	Cambia el contenido del control Label. Ocurre cuando un vínculo DDE actualiza los datos o cuando usted cambia la configuración de la propiedad <i>Caption</i> mediante código
<i>Clic</i>	Ocurre cuando el usuario presiona y suelta un botón del mouse (ratón) en un objeto. También puede ocurrir cuando se cambia el valor de un control.
<i>DbClick</i>	Se produce cuando un usuario presiona y suelta un botón del mouse (ratón) dos veces sobre un objeto.
<i>DragDrop</i>	Ocurre cuando se completa una operación de arrastrar y colocar como resultado de arrastrar un control sobre un objeto y soltar el botón del mouse o usar el método <i>Drag</i> con su argumento acción establecido a 2 ( <i>Drop</i> ).
<i>DragOver</i>	Ocurre cuando una operación de arrastrar y colocar está en curso. Puede usar este evento para controlar el puntero del mouse a medida que entra, sale o descansa directamente sobre un destino válido. La posición del puntero del mouse determina el objeto de destino que recibe este evento.
<i>LinkClose</i>	Ocurre cuando termina una conversación DDE. Cualquier aplicación de una conversación DDE puede terminar una conversación en cualquier momento.
<i>LinkError</i>	Ocurre cuando se produce un error durante una conversación DDE. Este evento sólo se reconoce como el resultado de un error relacionado con DDE que ocurre cuando no se está ejecutando código de Visual Basic. El número de error se pasa como un argumento.
<i>LinkNotify</i>	Ocurre cuando el origen ha cambiado los datos definidos por el vínculo DDE si la propiedad <i>LinkMode</i> del control de destino está establecida a 3 ( <i>Notify</i> ).
<i>LinkOpen</i>	Ocurre cuando se está iniciando una conversación DDE.
<i>MouseDown</i> y <i>MouseUp</i>	Ocurren cuando el usuario presiona ( <i>MouseDown</i> ) o suelta ( <i>MouseUp</i> ) un botón del mouse.
<i>MouseMove</i>	Ocurre cuando el usuario mueve el mouse.
<i>OLECompleteDrag</i>	Se produce cuando un componente de origen se coloca en un componente de destino e informa al componente de origen de que se ha realizado o cancelado una operación de arrastre.

<b>Eventos</b>	<b>Descripción</b>
<i>OLEDragDrop</i>	Se produce cuando un componente de origen se coloca en un componente de destino, si el componente de origen determina que la colocación puede realizarse. Este evento sólo se produce cuando OLEDropMode está establecido a 1 (Manual).
<i>OLEDragOver</i>	Se produce cuando un componente se arrastra sobre otro.
<i>OLEGiveFeedback</i>	Se produce después de cada evento OLEDragOver. OLEGiveFeedback permite al componente de origen proporcionar al usuario una indicación visual, como cambiar el cursor del mouse para indicar lo que ocurrirá si coloca el objeto o señalar visualmente la selección (en el componente de origen) para indicar lo que va a ocurrir.
<i>OLESetData</i>	Se produce en un componente de origen cuando un componente de destino realiza el método GetData sobre el objeto DataObject del origen, pero antes de que se carguen los datos del formato especificado.
<i>OLEStartDrag</i>	Se produce cuando se ejecuta el método OLEDrag de un componente o cuando un componente inicia una operación OLE de arrastrar y colocar, y la propiedad OLEDragMode está establecida a Automatic. Este evento especifica los formatos de los datos y los efectos de colocación que admite el componente de origen. También puede usarlo para insertar datos en el objeto DataObject.

Métodos correspondientes al control Label:

<b>Métodos</b>	<b>Descripción</b>
<i>Drag</i>	Inicia, termina o cancela una operación de arrastre de cualquier control excepto los controles Line, Menu, Shape, Timer o CommonDialog. No acepta argumentos con nombre.
<i>LinkExecute</i>	Envía una cadena de comandos a la aplicación de origen de una conversación DDE. No acepta argumentos con nombre.
<i>LinkPoke</i>	Transfiere el contenido de un control Label, PictureBox o TextBox a la aplicación de origen de una conversación DDE.
<i>LinkRequest</i>	Pide a la aplicación de origen de una conversación DDE que actualice el contenido de un control Label, PictureBox o TextBox.
<i>LinkSend</i>	Transfiere el contenido de un control PictureBox a la aplicación de destino de una conversación DDE.
<i>Move</i>	Mueve objetos MDIForm o Form, o controles. No acepta argumentos con nombre.
<i>OLEDrag</i>	Hace que un componente inicie una operación OLE de arrastrar y colocar.
<i>Refresh</i>	Fuerza el volver a dibujar un formulario o un control completo.
<i>ShowWhatsThis</i>	Muestra un tema seleccionado de un archivo de Ayuda utilizando el menú emergente ¿Qué es esto? que ofrece la Ayuda de Windows 95.
<i>Zorder</i>	Coloca un objeto MDIForm o Form, o un control al principio o al final del orden z dentro de su nivel gráfico. No acepta argumentos con nombre.

Propiedades correspondientes al control Label:

<b>Propiedades</b>	<b>Descripción</b>
<i>Alignment</i>	Devuelve o establece un valor que determina la alineación del control.
<i>Appearance</i>	Devuelve o establece el estilo de dibujo de los controles de un objeto MDIForm o Form en tiempo de diseño. Es de sólo lectura en tiempo de ejecución.
<i>AutoSize</i>	Devuelve o establece un valor que determina si el tamaño de un control cambia automáticamente para presentar todo su contenido.
<i>BackColor</i> y <i>ForeColor</i>	<i>BackColor</i> : devuelve o establece el color de fondo de un objeto. <i>ForeColor</i> : devuelve o establece el color de primer plano utilizado para mostrar texto y gráficos en un objeto.
<i>BackStyle</i>	Devuelve o establece un valor que indica si un control Label o el fondo de un color Shape es transparente u opaco.
<i>BorderStyle</i>	Devuelve o establece el estilo del borde de un objeto. Para el objeto Form y el control TextBox es de sólo lectura en tiempo de ejecución.
<i>Caption</i>	Determina el texto que se muestra en el control.
<i>Container</i>	Devuelve o establece el contenedor de un control de un Form. No está disponible en tiempo de diseño.
<i>DataChanged</i>	Devuelve o establece un valor que indica que algún proceso distinto de la recuperación de datos del registro actual ha cambiado los datos del control enlazado. No está disponible en tiempo de diseño.
<i>DataField</i>	Devuelve o establece el nombre de un campo al que será enlazado un receptor de datos.
<i>DataFormat</i>	Devuelve o establece el objeto StdDataFormat al que está adjunto un objeto vinculado. Lectura y escritura en tiempo de diseño y en tiempo de ejecución.
<i>DataMember</i>	Devuelve o establece un miembro de datos especificado entre los varios que ofrece el proveedor de datos.
<i>DataSource</i>	Devuelve o establece el origen de datos mediante el cual un receptor de datos enlaza con una base de datos.
<i>DragIcon</i>	Devuelve o establece el icono que se presenta como puntero en una operación de arrastrar y colocar.
<i>DragMode</i>	Devuelve o establece un valor que determina si se usa el modo de arrastre manual o automático en una operación de arrastrar y colocar.
<i>Enabled</i>	Devuelve o establece un valor que determina si un formulario o un control puede responder a eventos generados por el usuario.
<i>Font</i>	Devuelve un objeto Font.
<i>FontBold</i> , <i>FontItalic</i> , <i>FontStrikethru</i> y <i>FontUnderline</i>	Devuelve o establece los estilos de fuente en los siguientes formatos: Negrita, Cursiva, <del>Tachado</del> y <u>Subrayado</u> .
<i>FontName</i>	Devuelve o establece la fuente utilizada para mostrar texto en un control o en una operación de dibujo o impresión en tiempo de ejecución.

<b>Propiedades</b>	<b>Descripción</b>
<i>FontSize</i>	Devuelve o establece el tamaño de la fuente que se va a usar para el texto mostrado en un control o en una operación de dibujo o impresión en tiempo de ejecución.
<i>Height y Width</i>	Devuelven o establecen las dimensiones de un objeto o el ancho del objeto Columns de un control DataGrid. Para los objetos Printer y Screen, no están disponibles en tiempo de diseño.
<i>Index (Matriz de controles)</i>	Devuelve o establece el número que identifica un control de forma exclusiva en una matriz de controles. Sólo está disponible si el control forma parte de una matriz de controles.
<i>Left y Top</i>	Left: devuelve o establece la distancia entre el borde interno izquierdo de un objeto y el borde izquierdo de su contenedor. Top: devuelve o establece la distancia entre el borde interno superior de un objeto y el borde superior de su contenedor.
<i>LinkItem</i>	Devuelve o establece los datos pasados a un control de destino en una conversación DDE con otra aplicación.
<i>LinkMode</i>	Devuelve o establece el tipo de vínculo usado en una conversación DDE y activa la conexión de la forma siguiente. Permite que un control de destino de un formulario de Visual Basic inicie una conversación, como se especifica en las propiedades LinkTopic y LinkItem.
<i>LinkTimeout</i>	Devuelve o establece la cantidad de tiempo que un control espera una respuesta a un mensaje DDE.
<i>LinkTopic</i>	Para un control de destino, devuelve o establece la aplicación de origen y el tema correspondiente (el agrupamiento de datos fundamental utilizado en esa aplicación). Utilice LinkTopic con la propiedad LinkItem para especificar el vínculo de datos completo.
<i>MouseIcon</i>	Devuelve o establece un icono personalizado de mouse (ratón).
<i>MousePointer</i>	Devuelve o establece un valor que indica el tipo de puntero de mouse que aparece cuando se sitúa el mouse sobre un área determinada de un objeto en tiempo de ejecución.
<i>Name</i>	Devuelve el nombre utilizado en el código para identificar un formulario, un control o un objeto de acceso a datos. Es de sólo lectura en tiempo de ejecución.
<i>OLEDrag</i>	Hace que un componente inicie una operación OLE de arrastrar y colocar.
<i>OLEDropMode</i>	Devuelve o establece el modo en que un componente de destino realiza las operaciones de colocación.
<i>Parent</i>	Devuelve el formulario, objeto o colección que contiene un control u otro objeto o colección.
<i>RightToLeft</i>	El control Label permite configurar la dirección de alineación y el orden de lectura del texto de la etiqueta. Los controles Label se crean inicialmente con una posición determinada por el valor predeterminado de la propiedad RightToLeft del formulario primario.
<i>TabIndex</i>	Devuelve o establece el orden de tabulación de la mayoría de los objetos dentro de su formulario primario.
<i>Tag</i>	Devuelve o establece una expresión que almacena cualquier dato adicional que necesite el programa. Al contrario de lo que sucede con

Propiedades	Descripción
	otras propiedades, Visual Basic no utiliza el valor de Tag. Puede usar esta propiedad para identificar objetos.
<i>ToolTipText</i>	Devuelve o establece Información sobre herramientas.
<i>UseMnemonic</i>	Devuelve o establece un valor que indica si al incluir el signo & en el texto de la propiedad Caption del control Label se define una tecla de acceso
<i>Visible</i>	Devuelve o establece un valor que indica si un objeto es visible o está oculto.
<i>WhatsThisHelpID</i>	Devuelve o establece un número de contexto asociado a un objeto. Se utiliza para dotar a las aplicaciones de Ayuda interactiva con el menú emergente ¿Qué es esto? de la Ayuda de Windows 95.
<i>WordWrap</i>	Devuelve o establece un valor que indica si un control Label con el valor True en su propiedad AutoSize se expande vertical u horizontalmente para adaptarse al texto especificado en su propiedad Caption.

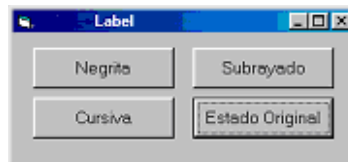
A continuación se desarrollará una aplicación que consiste en utilizar las propiedades *FontItalic*, *FontBold* y *FontUnderline* de una etiqueta llamada Label1 de acuerdo al botón pulsado.

En el menú Archivo, seleccione la opción Nuevo proyecto y pulse Aceptar para crear un nuevo archivo ejecutable de tipo estándar.

Ajuste el tamaño del formulario hasta que alcance las dimensiones de un cuadro de diálogo de tamaño medio.

Pulse el control *Label* y cree una etiqueta alargada en la parte superior del formulario.

Cree cuatro botones de pulsación. Básese en la figura siguiente para determinar su emplazamiento.



Asigne las siguientes propiedades para los objetos mencionados.

Objeto	Propiedad	Valor
<i>Label1</i>	Caption	UTN
<i>Command1</i>	Nombre	cmdNegrita
	Caption	Negrita
<i>Command2</i>	Nombre	cmdSubrayado
	Caption	Subrayado
<i>Command3</i>	Nombre	cmdCursiva
	Caption	Cursiva
<i>Command4</i>	Nombre	cmdEstadoOriginal
	Caption	EstadoOriginal

A continuación, deberá introducir el código asociado con los cuatro procedimientos de evento clic.

Realice una doble pulsación sobre el objeto *cmdNegrita*. El procedimiento de evento *cmdNegrita\_Click* aparecerá en la ventana Código. Escriba lo siguiente:

```
Private Sub cmdNegrita_Click()  
    Label1.FontBold = True
```

End Sub

Esto hará que la etiqueta se torne negrita.

Ahora escriba el código correspondiente al evento clic del botón de pulsación *cmdSubrayado*:

```
Private Sub cmdSubrayado_Click()  
    Label1.FontUnderline = True  
End Sub
```

Introduzca las siguientes sentencias de programa en el procedimiento de evento *cmdCursiva\_Clic*:

```
Private Sub cmdCursiva_Click()  
    Label1.FontItalic = True  
End Sub
```

Para que la etiqueta vuelva a su fuente original escriba el siguiente código en el evento clic del botón de pulsación *cmdEstadoOriginal*.

```
Private Sub cmdEstadoOriginal_Click()  
    Label1.FontItalic = False  
    Label1.FontBold = False  
    Label1.FontUnderline = False  
End Sub
```

Al ejecutar la aplicación, por ejemplo al pulsar el botón *Negrita* y luego *Cursiva*, la ventana se mostrará como en la imagen que sigue:





### TextBox (Caja de texto)

Un control *TextBox*, también denominado control de campo de edición o control de edición, muestra información introducida en tiempo de diseño, introducida por el usuario o asignada al control en código en tiempo de ejecución.

Para mostrar múltiples líneas de texto en un control *TextBox*, establezca la propiedad *MultiLine* a *True*. Si un *TextBox* de múltiples líneas no tiene una barra de desplazamiento horizontal, el texto se ajustará automáticamente aunque se ajuste el tamaño de *TextBox*. Para personalizar la combinación de barra de desplazamiento en un *TextBox*, establezca la propiedad *ScrollBars*.

Las barras de desplazamiento siempre aparecerán en el *TextBox* cuando su propiedad *MultiLine* se haya establecido a *True* y su propiedad *ScrollBars* a cualquier cosa excepto *None* (0).

Si establece la propiedad *MultiLine* a *True*, podrá emplear la propiedad *Alignment* para establecer la alineación del texto dentro del *TextBox*. El texto está justificado a la izquierda de forma predeterminada. Si la propiedad *MultiLine* es *False*, el valor de la propiedad *Alignment* no tiene ningún efecto.

Eventos a los que responde el control *TextBox*:

Eventos	Descripción
<i>Change</i>	Indica que el contenido de una caja de texto ha cambiado. Ocurre cuando un vínculo DDE actualiza los datos, cuando un usuario cambia el texto o cuando cambia la configuración de la propiedad <i>Text</i> mediante código.
<i>Click</i>	Ocurre cuando el usuario presiona y suelta un botón del mouse (ratón) en un objeto. También puede ocurrir cuando se cambia el valor de un control.
<i>DbClick</i>	Se produce cuando un usuario presiona y suelta un botón del mouse (ratón) dos veces sobre un objeto.
<i>DragDrop</i>	Ocurre cuando se completa una operación de arrastrar y colocar como resultado de arrastrar un control sobre un objeto y soltar el botón del mouse o usar el método <i>Drag</i> con su argumento acción establecido a 2 ( <i>Drop</i> ).
<i>DragOver</i>	Ocurre cuando una operación de arrastrar y colocar está en curso. Puede usar este evento para controlar el puntero del mouse a medida que entra, sale o descansa directamente sobre un destino válido. La posición del puntero del mouse determina el objeto de destino que recibe este evento.
<i>GotFocus</i>	Ocurre cuando un objeto recibe el enfoque, ya sea mediante una acción del usuario, como tabular o hacer clic en el objeto, o cambiando el enfoque en el código mediante el método <i>SetFocus</i> .
<i>KeyDown</i> y <i>KeyUp</i>	Ocurren cuando el usuario presiona ( <i>KeyDown</i> ) o suelta ( <i>KeyUp</i> ) una tecla mientras un objeto tiene el enfoque.
<i>KeyPress</i>	Ocurre cuando el usuario presiona y suelta una tecla ANSI.
<i>LinkClose</i>	Ocurre cuando termina una conversación DDE. Cualquier aplicación de una conversación DDE puede terminar una conversación en cualquier momento.
<i>LinkError</i>	Ocurre cuando se produce un error durante una conversación DDE. Este evento sólo se reconoce como el resultado de un error relacionado con DDE que ocurre cuando no se está ejecutando código de Visual Basic. El número de error se pasa como un argumento.
<i>LinkNotify</i>	Ocurre cuando el origen ha cambiado los datos definidos por el vínculo DDE si la propiedad <i>LinkMode</i> del control de destino está establecida a 3 ( <i>Notify</i> ).
<i>LinkOpen</i>	Ocurre cuando se está iniciando una conversación DDE.

Eventos	Descripción
<i>LostFocus</i>	Ocurre cuando un objeto pierde el enfoque, ya sea por una acción del usuario, como tabular o hacer clic en otro objeto, o bien mediante un cambio del enfoque en el código con el método <i>SetFocus</i> .
<i>MouseDown</i> y <i>MouseUp</i>	Ocurren cuando el usuario presiona ( <i>MouseDown</i> ) o suelta ( <i>MouseUp</i> ) un botón del mouse.
<i>MouseMove</i>	Ocurre cuando el usuario mueve el mouse.
<i>OLECompleteDrag</i>	Se produce cuando un componente de origen se coloca en un componente de destino e informa al componente de origen de que se ha realizado o cancelado una operación de arrastre.
<i>OLECompleteDrag</i> (Controles <i>ActiveX</i> )	Se produce cuando un componente de origen se coloca en un componente de destino e informa al componente de origen de que se ha realizado o cancelado una operación de arrastre.
<i>OLEDragDrop</i>	Se produce cuando un componente de origen se coloca en un componente de destino, si el componente de origen determina que la colocación puede realizarse. Nota: Este evento sólo se produce cuando <i>OLEDropMode</i> está establecido a 1 (Manual).
<i>OLEDragOver</i>	Se produce cuando un componente se arrastra sobre otro.
<i>OLEDragOver</i> (Controles <i>ActiveX</i> )	Se produce cuando un componente se arrastra sobre otro.
<i>OLEGiveFeedback</i>	Se produce después de cada evento <i>OLEDragOver</i> . <i>OLEGiveFeedback</i> permite al componente de origen proporcionar al usuario una indicación visual, como cambiar el cursor del mouse para indicar lo que ocurrirá si coloca el objeto o señalar visualmente la selección (en el componente de origen) para indicar lo que va a ocurrir.
<i>OLESetData</i>	Se produce en un componente de origen cuando un componente de destino realiza el método <i>GetData</i> sobre el objeto <i>DataObject</i> del origen, pero antes de que se carguen los datos del formato especificado.
<i>OLEStartDrag</i>	Se produce cuando se ejecuta el método <i>OLEDrag</i> de un componente o cuando un componente inicia una operación OLE de arrastrar y colocar, y la propiedad <i>OLEDragMode</i> está establecida a <i>Automatic</i> .
<i>Validate</i>	Ocurre cuando el foco cambia a un (segundo) control que tiene su propiedad <i>CausesValidation</i> establecida a <i>True</i> .

Métodos correspondientes al control *TextBox*:

Métodos	Descripción
<i>Drag</i>	Inicia, termina o cancela una operación de arrastre del control.
<i>LinkExecute</i>	Envía una cadena de comandos a la aplicación de origen de una conversación DDE. No acepta argumentos con nombre.
<i>LinkPoke</i>	Transfiere el contenido del control a la aplicación de origen de una conversación DDE.
<i>LinkRequest</i>	Pide a la aplicación de origen de una conversación DDE que actualice el contenido de un control.
<i>LinkSend</i>	Transfiere el contenido de un control <i>PictureBox</i> a la aplicación de destino de una conversación DDE.

<b>Métodos</b>	<b>Descripción</b>
<i>Move</i>	Mueve controles. No acepta argumentos con nombre.
<i>OLEDrag</i>	Hace que un componente inicie una operación OLE de arrastrar y colocar.
<i>Refresh</i>	Fuerza el volver a dibujar un formulario o un control completo.
<i>SetFocus</i>	Mueve el enfoque al control o formulario especificado.
<i>ShowWhatsThis</i>	Muestra un tema seleccionado de un archivo de Ayuda utilizando el menú emergente ¿Qué es esto? que ofrece la Ayuda de Windows 95.
<i>ZOrder</i>	Coloca un control al principio o al final del orden z dentro de su nivel gráfico. No acepta argumentos con nombre.

Propiedades correspondientes al control TextBox:

<b>Propiedades</b>	<b>Descripción</b>
<i>Alignment</i>	Devuelve o establece un valor que determina la alineación del texto de el control. Es de sólo lectura en tiempo de ejecución para este control.
<i>Appearance</i>	Devuelve o establece el estilo de dibujo de los controles de un objeto MDIForm o Form en tiempo de diseño. Es de sólo lectura en tiempo de ejecución.
<i>BackColor</i> <i>ForeColor</i>	y BackColor: devuelve o establece el color de fondo de un objeto. ForeColor: devuelve o establece el color de primer plano utilizado para mostrar texto y gráficos en un objeto.
<i>BorderStyle</i>	Devuelve o establece el estilo del borde de un objeto. Para este control es de sólo lectura en tiempo de ejecución.
<i>Container</i>	Devuelve o establece el contenedor de un control de un Form. No está disponible en tiempo de diseño
<i>DataChanged</i>	Devuelve o establece un valor que indica que algún proceso distinto de la recuperación de datos del registro actual ha cambiado los datos del control enlazado. No está disponible en tiempo de diseño.
<i>DataField</i>	Devuelve o establece el nombre de un campo al que será enlazado un receptor de datos.
<i>DataFormat</i>	Devuelve o establece el objeto StdDataFormat al que está adjunto un objeto vinculado. Lectura y escritura en tiempo de diseño y en tiempo de ejecución.
<i>DataMember</i>	Devuelve o establece un miembro de datos especificado entre los varios que ofrece el proveedor de datos.
<i>DataSource</i>	Devuelve o establece el origen de datos mediante el cual un receptor de datos enlaza con una base de datos.
<i>DragIcon</i>	Devuelve o establece el icono que se presenta como puntero en una operación de arrastrar y colocar.
<i>DragMode</i>	Devuelve o establece un valor que determina si se usa el modo de arrastre manual o automático en una operación de arrastrar y colocar.
<i>Enabled</i>	Devuelve o establece un valor que determina si un la caja de texto puede responder a eventos generados por el usuario.
<i>Font</i>	Devuelve un objeto Font.

<b>Propiedades</b>	<b>Descripción</b>
<i>FontBold, FontItalic, FontStrikethru y FontUnderline</i>	Devuelve o establece los estilos de fuente en los siguientes formatos: Negrita, Cursiva, <del>Tachado</del> y <u>Subrayado</u> .
<i>FontName</i>	Devuelve o establece la fuente utilizada para mostrar texto en el control.
<i>FontSize</i>	Devuelve o establece el tamaño de la fuente que se va a usar para el texto mostrado en el control.
<i>Height y Width</i>	Devuelven o establecen las dimensiones del control.
<i>HelpContextID</i>	Devuelve o establece un número de contexto asociado a un objeto. Se utiliza para proporcionar Ayuda interactiva para la aplicación.
<i>HideSelection</i>	Devuelve un valor que determina si el texto seleccionado aparece resaltado cuando un control pierde el enfoque.
<i>hWnd</i>	Devuelve el controlador de un formulario o un control. Nota: Esta propiedad no es compatible con el control contenedor OLE.
<i>Index (Control Array)</i>	Devuelve o establece el número que identifica un control de forma exclusiva en una matriz de controles. Sólo está disponible si el control forma parte de una matriz de controles.
<i>Left y Top</i>	Left: devuelve o establece la distancia entre el borde interno izquierdo de un objeto y el borde izquierdo de su contenedor. Top: devuelve o establece la distancia entre el borde interno superior de un objeto y el borde superior de su contenedor.
<i>LinkItem</i>	Devuelve o establece los datos pasados a un control de destino en una conversación DDE con otra aplicación.
<i>LinkMode</i>	Devuelve o establece el tipo de vínculo usado en una conversación DDE y activa la conexión de la forma siguiente: permite que un control de destino de un formulario de Visual Basic inicie una conversación, como se especifica en las propiedades LinkTopic y LinkItem.
<i>LinkTimeout</i>	Devuelve o establece la cantidad de tiempo que un control espera una respuesta a un mensaje DDE.
<i>LinkTopic</i>	Para un control de destino, devuelve o establece la aplicación de origen y el tema correspondiente (el agrupamiento de datos fundamental utilizado en esa aplicación). Utilice LinkTopic con la propiedad LinkItem para especificar el vínculo de datos completo.
<i>Locked</i>	Devuelve o establece un valor que indica si un control se puede modificar.
<i>MaxLength</i>	Devuelve o establece un valor que indica si existe un número máximo de caracteres que se pueden introducir en un control TextBox y, en caso afirmativo, especifica dicho valor máximo.
<i>MouseIcon</i>	Devuelve o establece un icono personalizado de mouse (ratón).
<i>MousePointer</i>	Devuelve o establece un valor que indica el tipo de puntero de mouse que aparece cuando se sitúa el mouse sobre un área determinada de un objeto en tiempo de ejecución.
<i>MultiLine</i>	Devuelve o establece un valor que indica si un control TextBox acepta y muestra múltiples líneas de texto. Es de sólo lectura en tiempo de ejecución.
<i>Name</i>	Devuelve el nombre utilizado en el código para identificar al control. Es de sólo lectura en tiempo de ejecución.
<i>OLEDragMode</i>	Devuelve o establece un valor que indica si las operaciones OLE de

Propiedades	Descripción
	arrastrar y colocar las controla el componente o el programador.
<i>OLEDropMode</i>	Devuelve o establece el modo en que un componente de destino realiza las operaciones de colocación.
<i>OLEDropMode (Controles ActiveX)</i>	Devuelve o establece el modo en que un componente de destino realiza las operaciones de colocación.
<i>Parent</i>	Devuelve el formulario, objeto o colección que contiene al control u otro objeto o colección.
<i>PasswordChar</i>	Devuelve o establece un valor que indica si en un control TextBox aparecen los caracteres que escribe el usuario o un carácter marcador de posición. Devuelve o establece el carácter utilizado como marcador de posición.
<i>ScrollBars</i>	Devuelve o establece un valor que indica si un objeto tiene barras de desplazamiento horizontal o vertical. Es de sólo lectura en tiempo de ejecución.
<i>SelLength, SelStart y SelText</i>	SelLength: devuelve o establece el número de caracteres seleccionados. SelStart: devuelve o establece el punto inicial del texto seleccionado, indicando la posición del punto de inserción si no se ha seleccionado texto. SelText: devuelve o establece una cadena con el texto seleccionado actualmente o es una cadena de longitud cero ("") si no hay caracteres seleccionados. Estas propiedades no están disponibles en tiempo de diseño.
<i>TabIndex</i>	Devuelve o establece el orden de tabulación de la mayoría de los objetos dentro de su formulario primario.
<i>TabStop</i>	Devuelve o establece un valor que indica si el usuario puede usar la tecla TAB para llevar el enfoque a un objeto.
<i>Tag</i>	Devuelve o establece una expresión que almacena cualquier dato adicional que necesite el programa. Al contrario de lo que sucede con otras propiedades, Visual Basic no utiliza el valor de Tag. Puede usar esta propiedad para identificar objetos.
<i>Text</i>	Devuelve o establece el texto contenido en el área de edición.
<i>ToolTipText</i>	Devuelve o establece Información sobre herramientas.
<i>Visible</i>	Devuelve o establece un valor que indica si un objeto es visible o está oculto.
<i>WhatsThisHelpID</i>	Devuelve o establece un número de contexto asociado a un objeto. Se utiliza para dotar a las aplicaciones de Ayuda interactiva con el menú emergente ¿Qué es esto? de la Ayuda de Windows 95.

A continuación se desarrollará una aplicación que consiste en un formulario en donde se ingresarán, por medio de cajas de texto, el código, nombre y fecha de nacimiento de odontólogos que trabaja en clínica Smile. Luego, al hacer clic sobre el botón de pulsación Validar, se verificará si los datos solicitados han sido ingresado y que los mismos sean correctos. En el menú Archivo, seleccione la opción Nuevo proyecto y pulse Aceptar para crear un nuevo archivo ejecutable de tipo estándar.

Pulse el control *TextBox* y cree tres cajas de texto en el formulario.

Cree un botón de pulsación. Basese en la figura siguiente para determinar su emplazamiento.

En el evento clic del botón de pulsación *Validar*, se realizarán cinco validaciones. Comencemos a describirlas.

La primer validación será la verificación de que el código del odontólogo haya sido ingresado:

```
If txtCodigo.Text = "" Then
    MsgBox "Ingrese el código del odontólogo.", vbCritical, _
        "Atención"
    txtCodigo.SetFocus
    Exit Sub
End If
```

Esta sección de código, mediante una sentencia *If ... Then*, verifica si la caja de texto es igual a una cadena vacía. Si lo es, muestra un mensaje de advertencia, coloca el cursor en dicha caja y sale del procedimiento.

Luego se valida si en la misma caja de texto se ha ingresado un valor numérico.

```
If Not IsNumeric(txtCodigo.Text) Then
    MsgBox "Ingrese el código del odontólogo correctamente", _
        vbCritical, "Atención"
    txtCodigo.SetFocus
    Exit Sub
End If
```

La función *IsNumeric* devuelve un valor de tipo Boolean que indica si una expresión se puede evaluar como un número. La función *IsNumeric* devuelve True si la expresión completa se reconoce como un número; en otro caso, devuelve False. La función *IsNumeric* devuelve False si expresión es una expresión de fecha.

Además se evalúa si los datos correspondiente al nombre y fecha de nacimiento del odontólogo han sido ingresados de la misma forma que en la primera validación.

```
If txtNombre.Text = "" Then
    MsgBox "Ingrese el nombre del odontólogo.", vbCritical, _
        "Atención"
    txtNombre.SetFocus
    Exit Sub
End If
```

```
If txtFecha.Text = "" Then
    MsgBox "Ingrese la fecha de nacimiento del odontólogo.", _
        vbCritical, "Atención"
    txtFecha.SetFocus
    Exit Sub
End If
```

También se validará si la fecha de nacimiento es del tipo Date:

```
If Not IsDate(txtFecha.Text) Then
    MsgBox "Ingrese la fecha de nacimiento del " & _
        "odontólogo correctamente.", vbCritical, "Atención"
    txtFecha.SetFocus
    Exit Sub
```

End If

La función *IsDate* devuelve un valor de tipo Boolean que indica si una expresión se puede convertir en una fecha. El argumento *expresión* requerido, es un tipo de datos Variant que contiene una expresión de fecha o una expresión de cadena reconocible como una fecha o una hora.

La función *IsDate* devuelve True si la expresión es una fecha o se puede reconocer como una fecha válida; en caso contrario, devuelve False. En Microsoft Windows, el intervalo de fechas válidas va desde el 1 de enero de 100 D. de C. hasta el 31 de diciembre de 9999 D. de C.; los intervalos varían de un sistema operativo a otro.

Por último, si se han ingresado los datos correctamente en las cajas de texto, se mostrará un mensaje informando tal situación.

```
MsgBox "LOS DATOS HAN SIDO INGRESADOS CORRECTAMENTE",_  
vbInformation, "INFORMACIÓN"
```

Este tipo de validaciones es la que se utilizará en todos las aplicaciones en las cuales sea necesario controlar los datos ingresados por el usuario.

**CommandButton (Botón de comando)**

Utilice un control *CommandButton* para comenzar, interrumpir o terminar un proceso. Cuando está activado, un *CommandButton* aparece presionado y a veces se denomina *botón de comando*.

Para mostrar texto en un control *CommandButton*, establezca su propiedad *Caption*. Un usuario siempre puede elegir un *CommandButton* si hace clic en él. Para permitir que el usuario lo elija presionando la tecla *ENTRAR*, establezca la propiedad *Default* a *True*.

Eventos a los que responde el control *CommandButton*:

<b>Eventos</b>	<b>Descripción</b>
<i>Click</i>	Ocurre cuando el usuario presiona y suelta un botón del mouse (ratón) en un objeto. Para un control, este evento ocurre cuando el usuario: Presiona <i>ENTRAR</i> cuando un formulario tiene un control <i>CommandButton</i> con su propiedad <i>Default</i> establecida a <i>True</i> .
<i>DragDrop</i>	Ocurre cuando se completa una operación de arrastrar y colocar como resultado de arrastrar un control sobre un objeto y soltar el botón del mouse o usar el método <i>Drag</i> con su argumento acción establecido a 2 ( <i>Drop</i> ).
<i>DragOver</i>	Ocurre cuando una operación de arrastrar y colocar está en curso. Puede usar este evento para controlar el puntero del mouse a medida que entra, sale o descansa directamente sobre un destino válido. La posición del puntero del mouse determina el objeto de destino que recibe este evento.
<i>GotFocus</i>	Ocurre cuando un objeto recibe el enfoque, ya sea mediante una acción del usuario, como tabular o hacer clic en el objeto, o cambiando el enfoque en el código mediante el método <i>SetFocus</i> . Un formulario recibe el enfoque sólo cuando todos los controles visibles están desactivados.
<i>KeyDown</i> y <i>KeyUp</i>	Ocurren cuando el usuario presiona ( <i>KeyDown</i> ) o suelta ( <i>KeyUp</i> ) una tecla mientras un objeto tiene el enfoque. (Para interpretar los caracteres ANSI, utilice el evento <i>KeyPress</i> .)
<i>KeyPress</i>	Ocurre cuando el usuario presiona y suelta una tecla ANSI.
<i>LostFocus</i>	Ocurre cuando un objeto pierde el enfoque, ya sea por una acción del usuario, como tabular o hacer clic en otro objeto, o bien mediante un cambio del enfoque en el código con el método <i>SetFocus</i> .
<i>MouseDown</i> y <i>MouseUp</i>	Ocurren cuando el usuario presiona ( <i>MouseDown</i> ) o suelta ( <i>MouseUp</i> ) un botón del mouse.
<i>MouseMove</i>	Ocurre cuando el usuario mueve el mouse.
<i>OLECompleteDrag</i>	Se produce cuando un componente de origen se coloca en un componente de destino e informa al componente de origen de que se ha realizado o cancelado una operación de arrastre.
<i>OLEDragDrop</i>	Se produce cuando un componente de origen se coloca en un componente de destino, si el componente de origen determina que la colocación puede realizarse. Nota: Este evento sólo se produce cuando <i>OLEDropMode</i> está establecido a 1 ( <i>Manual</i> ).
<i>OLEDragOver</i>	Se produce cuando un componente se arrastra sobre otro.



Eventos	Descripción
<i>OLEGiveFeedback</i>	Se produce después de cada evento <i>OLEDragOver</i> . <i>OLEGiveFeedback</i> permite al componente de origen proporcionar al usuario una indicación visual, como cambiar el cursor del mouse para indicar lo que ocurrirá si coloca el objeto o señalar visualmente la selección (en el componente de origen) para indicar lo que va a ocurrir.
<i>OLESetData</i>	Se produce en un componente de origen cuando un componente de destino realiza el método <i>GetData</i> sobre el objeto <i>DataObject</i> del origen, pero antes de que se carguen los datos del formato especificado.
<i>OLEStartDrag</i>	Se produce cuando se ejecuta el método <i>OLEDrag</i> de un componente o cuando un componente inicia una operación OLE de arrastrar y colocar, y la propiedad <i>OLEDragMode</i> está establecida a <i>Automatic</i> . Este evento especifica los formatos de los datos y los efectos de colocación que admite el componente de origen. También puede usarlo para insertar datos en el objeto <i>DataObject</i> .
<i>Validate</i>	Ocurre cuando el foco cambia a un (segundo) control que tiene su propiedad <i>CausesValidation</i> establecida a <i>True</i> .

Métodos correspondientes al control *CommandButton*:

Métodos	Descripción
<i>Drag</i>	Inicia, termina o cancela una operación de arrastre del control. No acepta argumentos con nombre.
<i>Move</i>	Mueve el control. No acepta argumentos con nombre.
<i>OLEDrag</i>	Hace que un componente inicie una operación OLE de arrastrar y colocar.
<i>Refresh</i>	Fuerza el volver a dibujar un formulario o un control completo.
<i>SetFocus</i>	Mueve el enfoque al control o formulario especificado.
<i>ShowWhatsThis</i>	Muestra un tema seleccionado de un archivo de Ayuda utilizando el menú emergente ¿Qué es esto? Que ofrece la Ayuda de Windows 95.
<i>ZOrder</i>	Coloca al control al principio o al final del orden z dentro de su nivel gráfico. No acepta argumentos con nombre.

Propiedades correspondientes al control *CommandButton*:

Propiedades	Descripción
<i>Appearance</i>	Devuelve o establece el estilo de dibujo de los controles de un objeto <i>MDIForm</i> o <i>Form</i> en tiempo de diseño. Es de sólo lectura en tiempo de ejecución.
<i>BackColor</i> <i>ForeColor</i>	y <i>BackColor</i> : devuelve o establece el color de fondo de un objeto. <i>ForeColor</i> : devuelve o establece el color de primer plano utilizado para mostrar texto y gráficos en un objeto.
<i>Cancel</i>	Devuelve o establece un valor que indica si un botón de comando es el botón Cancelar de un formulario. Este botón de comando puede ser un control <i>CommandButton</i> o cualquier otro objeto contenido en un control contenedor OLE que se comporte como un botón de comando.
<i>Caption</i>	Determina el texto que se muestra en un control.
<i>Container</i>	Devuelve o establece el contenedor de un control de un <i>Form</i> . No está disponible en tiempo de diseño.

<b>Propiedades</b>	<b>Descripción</b>
<i>Default</i>	Devuelve o establece un valor que determina qué control CommandButton es el botón de comando predeterminado de un formulario.
<i>DisabledPicture</i>	Devuelve o establece una referencia a una imagen que se va a mostrar en un control cuando está inactivo. (Es decir, cuando su propiedad Enabled se establece a False).
<i>DownPicture</i>	Devuelve o establece una referencia a una imagen que se muestra en un control cuando se hace clic en él y está presionado.
<i>DragIcon</i>	Devuelve o establece el icono que se presenta como puntero en una operación de arrastrar y colocar.
<i>DragMode</i>	Devuelve o establece un valor que determina si se usa el modo de arrastre manual o automático en una operación de arrastrar y colocar.
<i>Enabled</i>	Devuelve o establece un valor que determina si un formulario o un control puede responder a eventos generados por el usuario.
<i>Font</i>	Devuelve un objeto Font.
<i>FontBold, FontItalic, FontStrikethru</i> y <i>FontUnderline</i>	Devuelve o establece los estilos de fuente en los siguientes formatos: Negrita, Cursiva, <del>Tachado</del> y <u>Subrayado</u> .
<i>FontName</i>	Devuelve o establece la fuente utilizada para mostrar texto en el control.
<i>FontSize</i>	Devuelve o establece el tamaño de la fuente que se va a usar para el texto mostrado en el control.
<i>Height y Width</i>	Devuelven o establecen las dimensiones del control.
<i>HelpContextID</i>	Devuelve o establece un número de contexto asociado a un objeto. Se utiliza para proporcionar Ayuda interactiva para la aplicación.
<i>hWnd</i>	Devuelve el controlador de un formulario o un control. Nota: Esta propiedad no es compatible con el control contenedor OLE.
<i>Index(Control Array)</i>	Devuelve o establece el número que identifica un control de forma exclusiva en una matriz de controles. Sólo está disponible si el control forma parte de una matriz de controles.
<i>Left y Top</i>	Left: devuelve o establece la distancia entre el borde interno izquierdo de un objeto y el borde izquierdo de su contenedor. Top: devuelve o establece la distancia entre el borde interno superior de un objeto y el borde superior de su contenedor.
<i>MaskColor</i>	Devuelve o establece un color de la imagen del botón que actuará de "máscara" (es decir, será transparente).
<i>MouseIcon</i>	Devuelve o establece un icono personalizado de mouse (ratón).
<i>MousePointer</i>	Devuelve o establece un valor que indica el tipo de puntero de mouse que aparece cuando se sitúa el mouse sobre un área determinada de un objeto en tiempo de ejecución.
<i>Name</i>	Devuelve el nombre utilizado en el código para identificar un formulario, un control o un objeto de acceso a datos. Es de sólo lectura en tiempo de ejecución.
<i>OLEDropMode</i>	Devuelve o establece el modo en que un componente de destino realiza las operaciones de colocación.
<i>Parent</i>	Devuelve el formulario, objeto o colección que contiene al control.
<i>Picture</i>	Devuelve el formulario, objeto o colección que contiene al control u otro

Propiedades	Descripción
	objeto o colección.
<i>RightToLeft</i>	El control <i>CommandButton</i> permite configurar el orden de lectura del título y proporciona compatibilidad bidireccional de idioma para el título del botón de comando. El orden predeterminado de lectura se hereda del valor de la propiedad <i>RightToLeft</i> del formulario primario; no obstante, es posible cambiar esta dirección mediante la propiedad <i>RightToLeft</i> del propio control <i>CommandButton</i> .
<i>Style</i>	Devuelve o establece un valor que indica el tipo de muestra y el comportamiento del control. Es de sólo lectura en tiempo de ejecución.
<i>TabIndex</i>	Devuelve o establece el orden de tabulación de la mayoría de los objetos dentro de su formulario primario.
<i>TabStop</i>	Devuelve o establece un valor que indica si el usuario puede usar la tecla TAB para llevar el enfoque a un objeto.
<i>ToolTipText</i>	Devuelve o establece Información sobre herramientas.
<i>UseMaskColor</i>	Devuelve o establece un valor que determina si el color asignado en la propiedad <i>MaskColor</i> se usa como "máscara", es decir, se usa para crear zonas transparentes.
<i>Value</i>	Devuelve o establece un valor que indica si se ha elegido el botón. No está disponible en tiempo de diseño.
<i>Visible</i>	Devuelve o establece un valor que indica si un objeto es visible o está oculto.
<i>WhatsThisHelpID</i>	Devuelve o establece un número de contexto asociado a un objeto. Se utiliza para dotar a las aplicaciones de Ayuda interactiva con el menú emergente ¿Qué es esto? de la Ayuda de Windows 95.

A continuación desarrollaremos una aplicación utilizando varias propiedades y eventos del control *CommandButton*. La interfaz será similar a la siguiente:



El objetivo de esta aplicación es que al pasar el mouse sobre uno de los tres botones de pulsación superiores, la fuente del botón *Reestablecer* cambiará dependiendo del botón de pulsación por el que se pasó el mouse y dicho botón será deshabilitado. Al pulsar *Reestablecer*, los botones superiores que se encontraban deshabilitados se habilitarán. A continuación se muestra una tabla con los controles y el valor de sus propiedades en tiempo de diseño:

Objeto	Propiedad	Valor
<i>CommandButton1</i>	Nombre	<i>cmdNegrita</i>
	Caption	Negrita
	Style	1 – Graphical
	Picture	Negrita.bmp

Objeto	Propiedad	Valor
<i>CommandButton2</i>	Nombre	cmdCursiva
	Caption	Cursiva
	Style	1 – Graphical
	Picture	Cursiva.bmp
<i>CommandButton3</i>	Nombre	cmdSubrayado
	Caption	Subrayado
	Style	1 – Graphical
	Picture	Subrayado.bmp
<i>CommandButton4</i>	Nombre	cmdReestablecer
	Caption	Reestablecer
	Style	1 – Graphical
	BackColor	Amarillo

Para cada uno de los botones de pulsación superiores, en el evento *MouseMove* escribiremos:

```
Private Sub cmdNegrita_MouseMove(Button As Integer, _
    Shift As Integer, X As Single, Y As Single)
    cmdReestablecer.FontBold = True
    cmdNegrita.Enabled = False
End Sub
```

```
Private Sub cmdCursiva_MouseMove(Button As Integer, _
    Shift As Integer, X As Single, Y As Single)
    cmdReestablecer.FontItalic = True
    cmdCursiva.Enabled = False
End Sub
```

```
Private Sub cmdSubrayado_MouseMove(Button As Integer, _
    Shift As Integer, X As Single, Y As Single)
    cmdReestablecer.FontUnderline = True
    cmdSubrayado.Enabled = False
End Sub
```

El efecto que causa cada uno de estos procedimientos de suceso es modificar el título del botón de pulsación *Reestablecer* a negrita, cursiva o subrayado y luego se deshabilitará el botón el cual se pasó el mouse. El evento *MouseMove* ocurre cada vez que el usuario mueve el puntero del ratón a nueva posición dentro del control.

El argumento *Button* indica qué botón se ha pulsado, y el argumento *Shift* si simultáneamente se ha pulsado una tecla *Shift*, *Ctrl* o *Alt*. Para comprobar los argumentos *Button* o *Shift*, puede utilizar las constantes que se listan en la biblioteca de objetos del examinador de objetos de Visual Basic.

Argumento *Button*:

Constante	Valor	Descripción
<i>vbLeftButton</i>	1	Se ha pulsado el botón izquierdo
<i>vbRightButton</i>	2	Se ha pulsado el botón derecho
<i>vbMiddleButton</i>	4	Se ha pulsado el botón central

Argumento *Shift*:

Constante	Valor	Descripción
<i>vbShiftMask</i>	1	Está presionada la tecla <i>Mayús</i> ( <i>Shift</i> )

Constante	Valor	Descripción
<i>vbCtrlMask</i>	2	Está presionada la tecla <i>Ctrl</i>
<i>vbAltMask</i>	4	Está presionada la tecla <i>Alt</i>

Para que los botones de pulsación vuelvan a su estado original, al hacer clic sobre *cmdReestablecer*, escriba el siguiente código:

```
Private Sub cmdReestablecer_Click()  
    cmdNegrita.Enabled = True  
    cmdCursiva.Enabled = True  
    cmdSubrayado.Enabled = True  
    cmdReestablecer.FontBold = False  
    cmdReestablecer.FontItalic = False  
    cmdReestablecer.FontUnderline = False  
End Sub
```

Ejecute la aplicación y observe los resultados.



### Picture Box (imagen)

Un control *PictureBox* puede mostrar un gráfico a partir de un mapa de bits, un icono o un metarchivo, así como un metarchivo mejorado, un archivo JPEG o un archivo GIF. Recorta el gráfico si el control no es lo bastante grande para mostrar la imagen completa.

También puede utilizar un control *PictureBox* para agrupar controles *OptionButton* y mostrar el resultado de métodos gráficos y texto escrito con el método *Print*.

Para hacer que un control *PictureBox* ajuste automáticamente su tamaño para mostrar un gráfico completo, establezca su propiedad *AutoSize* a *True*.

Eventos a los que responde el control *PictureBox*:

Eventos	Descripción
<i>Change</i>	Indica que el contenido de un control ha cambiado. Ocurre cuando un vínculo DDE actualiza los datos o cuando usted cambia la configuración de la propiedad <i>Picture</i> mediante código
<i>Clic</i>	Ocurre cuando el usuario presiona y suelta un botón del mouse (ratón) en un objeto. También puede ocurrir cuando se cambia el valor de un control.
<i>DbIClick</i>	Se produce cuando un usuario presiona y suelta un botón del mouse (ratón) dos veces sobre un objeto.
<i>DragDrop</i>	Ocurre cuando se completa una operación de arrastrar y colocar como resultado de arrastrar un control sobre un objeto y soltar el botón del mouse o usar el método <i>Drag</i> con su argumento acción establecido a 2 ( <i>Drop</i> ).
<i>DragOver</i>	Ocurre cuando una operación de arrastrar y colocar está en curso. Puede usar este evento para controlar el puntero del mouse a medida que entra, sale o descansa directamente sobre un destino válido. La posición del puntero del mouse determina el objeto de destino que recibe este evento.
<i>GotFocus</i>	Ocurre cuando un objeto recibe el enfoque, ya sea mediante una acción del usuario, como tabular o hacer clic en el objeto, o cambiando el enfoque en el código mediante el método <i>SetFocus</i> . Un formulario recibe el enfoque sólo cuando todos los controles visibles están desactivados.
<i>KeyDown</i> y <i>KeyUp</i>	Ocurren cuando el usuario presiona ( <i>KeyDown</i> ) o suelta ( <i>KeyUp</i> ) una tecla mientras un objeto tiene el enfoque. (Para interpretar los caracteres ANSI, utilice el evento <i>KeyPress</i> .)
<i>KeyPress</i>	Ocurre cuando el usuario presiona y suelta una tecla ANSI.
<i>LinkClose</i>	Ocurre cuando termina una conversación DDE. Cualquier aplicación de una conversación DDE puede terminar una conversación en cualquier momento.
<i>LinkError</i>	Ocurre cuando se produce un error durante una conversación DDE. Este evento sólo se reconoce como el resultado de un error relacionado con DDE que ocurre cuando no se está ejecutando código de Visual Basic. El número de error se pasa como un argumento.
<i>LinkNotify</i>	Ocurre cuando el origen ha cambiado los datos definidos por el vínculo DDE si la propiedad <i>LinkMode</i> del control de destino está establecida a 3 ( <i>Notify</i> ).
<i>LinkOpen</i>	Ocurre cuando se está iniciando una conversación DDE.

Eventos	Descripción
<i>LostFocus</i>	Ocurre cuando un objeto pierde el enfoque, ya sea por una acción del usuario, como tabular o hacer clic en otro objeto, o bien mediante un cambio del enfoque en el código con el método <i>SetFocus</i> .
<i>MouseDown</i> y <i>MouseUp</i>	Ocurren cuando el usuario presiona ( <i>MouseDown</i> ) o suelta ( <i>MouseUp</i> ) un botón del mouse.
<i>MouseMove</i>	Ocurre cuando el usuario mueve el mouse.
<i>OLECompleteDrag</i>	Se produce cuando un componente de origen se coloca en un componente de destino e informa al componente de origen de que se ha realizado o cancelado una operación de arrastre.
<i>OLEDragDrop</i>	Se produce cuando un componente de origen se coloca en un componente de destino, si el componente de origen determina que la colocación puede realizarse. Este evento sólo se produce cuando <i>OLEDropMode</i> está establecido a 1 ( <i>Manual</i> ).
<i>OLEDragOver</i>	Se produce cuando un componente se arrastra sobre otro.
<i>OLEGiveFeedback</i>	Se produce después de cada evento <i>OLEDragOver</i> . <i>OLEGiveFeedback</i> permite al componente de origen proporcionar al usuario una indicación visual, como cambiar el cursor del mouse para indicar lo que ocurrirá si coloca el objeto o señalar visualmente la selección (en el componente de origen) para indicar lo que va a ocurrir.
<i>OLESetData</i>	Se produce en un componente de origen cuando un componente de destino realiza el método <i>GetData</i> sobre el objeto <i>DataObject</i> del origen, pero antes de que se carguen los datos del formato especificado.
<i>OLEStartDrag</i>	Se produce cuando se ejecuta el método <i>OLEDrag</i> de un componente o cuando un componente inicia una operación OLE de arrastrar y colocar, y la propiedad <i>OLEDragMode</i> está establecida a <i>Automatic</i> . Este evento especifica los formatos de los datos y los efectos de colocación que admite el componente de origen. También puede usarlo para insertar datos en el objeto <i>DataObject</i> .
<i>Paint</i>	Ocurre cuando un objeto entero o una parte del mismo se expone después de haberse movido o ampliado, o después de haberse movido una ventana que lo estaba cubriendo.
<i>Resize</i>	Ocurre cuando un objeto se muestra primero o cuando cambia el estado de una ventana. (Por ejemplo, cuando se maximiza, minimiza o restaura un formulario.)
<i>Validate</i>	Ocurre cuando el foco cambia a un (segundo) control que tiene su propiedad <i>CausesValidation</i> establecida a <i>True</i> .

Métodos correspondientes al control *PictureBox*:

Métodos	Descripción
<i>Circle</i>	Dibuja un círculo, una elipse o un arco en un objeto.
<i>Cls</i>	Borra los gráficos y el texto generados en tiempo de ejecución de los controles <i>Form</i> o <i>PictureBox</i>
<i>Drag</i>	Inicia, termina o cancela una operación de arrastre de cualquier control excepto los controles <i>Line</i> , <i>Menu</i> , <i>Shape</i> , <i>Timer</i> o <i>CommonDialog</i> . No acepta argumentos con nombre.

<b>Métodos</b>	<b>Descripción</b>
<i>Line</i>	Dibuja líneas y rectángulos en un objeto
<i>LinkExecute</i>	Envía una cadena de comandos a la aplicación de origen de una conversación DDE. No acepta argumentos con nombre.
<i>LinkPoke</i>	Transfiere el contenido de un control Label, PictureBox o TextBox a la aplicación de origen de una conversación DDE.
<i>LinkRequest</i>	Pide a la aplicación de origen de una conversación DDE que actualice el contenido de un control Label, PictureBox o TextBox.
<i>LinkSend</i>	Transfiere el contenido de un control PictureBox a la aplicación de destino de una conversación DDE.
<i>Move</i>	Mueve objetos MDIForm o Form, o controles. No acepta argumentos con nombre.
<i>OleDrag</i>	Hace que un componente inicie una operación OLE de arrastrar y colocar.
<i>PaintPicture</i>	Dibuja el contenido de un archivo gráfico (.bmp, .wmf, .emf, .cur, .ico o .dib) en un objeto Form, PictureBox o Printer. No acepta argumentos con nombre.
<i>Point</i>	Devuelve, como un valor de tipo Integer Long, el color rojo-verde-azul (RGB) del punto especificado de un objeto Form o un control PictureBox. No acepta argumentos con nombre.
<i>PSet</i>	Establece un punto de un objeto a un color especificado.
<i>Refresh</i>	Fuerza el volver a dibujar un formulario o un control completo.
<i>Scale</i>	Define el sistema de coordenadas para un control Form, PictureBox o Printer. No acepta argumentos con nombre.
<i>ScaleX, ScaleY</i>	Convierten el valor del ancho o el alto de un control Form, PictureBox o Printer de una unidad de medida de la propiedad ScaleMode a otra. No aceptan argumentos con nombre.
<i>SetFocus</i>	Mueve el enfoque al control o formulario especificado.
<i>ShowWhatsThis</i>	Muestra un tema seleccionado de un archivo de Ayuda utilizando el menú emergente ¿Qué es esto? que ofrece la Ayuda de Windows 95.
<i>TextHeight</i>	Devuelve el alto de una cadena de texto tal y como se imprimiría con la fuente actual de un objeto Form, PictureBox o Printer. No acepta argumentos con nombre.
<i>TextWidth</i>	Devuelve el ancho de una cadena de texto, tal como se imprimiría con la fuente actual de un objeto Form, PictureBox o Printer. No acepta argumentos con nombre.
<i>ZOrder</i>	Coloca al control al principio o al final del orden z dentro de su nivel gráfico. No acepta argumentos con nombre.



Propiedades correspondientes al control *PictureBox*

Propiedades	Descripción
<i>Align</i>	Devuelve o establece un valor que determina si un objeto se presenta en cualquier tamaño y posición dentro de un formulario o si se presenta en la parte superior, inferior, izquierda o derecha del mismo y si su tamaño se ajusta automáticamente al ancho del formulario.
<i>Appearance</i>	Devuelve o establece el estilo de dibujo de los controles de un objeto MDIForm o Form en tiempo de diseño. Es de sólo lectura en tiempo de ejecución.
<i>AutoRedraw</i>	Devuelve o establece el resultado de un método gráfico como gráfico persistente.
<i>AutoSize</i>	Devuelve o establece un valor que determina si el tamaño de un control cambia automáticamente para presentar todo su contenido
<i>BackColor</i> <i>ForeColor</i>	y BackColor: devuelve o establece el color de fondo de un objeto. ForeColor: devuelve o establece el color de primer plano utilizado para mostrar texto y gráficos en un objeto.
<i>BorderStyle</i>	Devuelve o establece el estilo del borde de un objeto.
<i>ClipControls</i>	Devuelve o establece un valor que determina si los métodos gráficos en eventos Paint vuelven a dibujar el objeto entero o sólo las áreas recién expuestas. También determina si el entorno operativo Microsoft Windows crea un área de recorte que excluya los controles no gráficos contenidos en el objeto. Es de sólo lectura en tiempo de ejecución.
<i>Container</i>	Devuelve o establece el contenedor de un control de un Form. No está disponible en tiempo de diseño.
<i>CurrentX, CurrentY</i>	Devuelven o establecen las coordenadas horizontal (CurrentX) o vertical (CurrentY) para el siguiente método gráfico o de impresión. No están disponibles en tiempo de diseño.
<i>DataChanged</i>	Devuelve o establece un valor que indica que algún proceso distinto de la recuperación de datos del registro actual ha cambiado los datos del control enlazado. No está disponible en tiempo de diseño.
<i>DataField</i>	Devuelve o establece el nombre de un campo al que será enlazado un receptor de datos.
<i>DataFormat</i>	Devuelve o establece el objeto StdDataFormat al que está adjunto un objeto vinculado. Lectura y escritura en tiempo de diseño y en tiempo de ejecución.
<i>DataMember</i>	Devuelve o establece un miembro de datos especificado entre los varios que ofrece el proveedor de datos.
<i>DataSource</i>	Devuelve o establece el origen de datos mediante el cual un receptor de datos enlaza con una base de datos.
<i>DragIcon</i>	Devuelve o establece el icono que se presenta como puntero en una operación de arrastrar y colocar.
<i>DragMode</i>	Devuelve o establece un valor que determina si se usa el modo de arrastre manual o automático en una operación de arrastrar y colocar.
<i>DrawMode</i>	Devuelve o establece un valor que determina la apariencia del resultado de un método gráfico o la apariencia de un control Shape o Line.
<i>DrawStyle</i>	Devuelve o establece un valor que determina el estilo de línea del

Propiedades	Descripción
	resultado de métodos gráficos.
<i>DrawWidth</i>	Devuelve o establece el ancho de línea para el resultado de los métodos gráficos.
<i>Enabled</i>	Devuelve o establece un valor que determina si un formulario o un control puede responder a eventos generados por el usuario.
<i>FillColor</i>	Devuelve o establece el color usado para llenar formas; <i>FillColor</i> también se usa para llenar círculos y cuadros creados con los métodos gráficos <i>Circle</i> y <i>Line</i> .
<i>FillStyle</i>	Devuelve o establece el patrón usado para llenar controles <i>Shape</i> , así como los círculos y los cuadros creados con los métodos gráficos <i>Circle</i> y <i>Line</i> .
<i>Font</i>	Devuelve un objeto <i>Font</i> .
<i>FontBold</i> , <i>FontItalic</i> , <i>FontStrikethru</i> y <i>FontUnderline</i>	Devuelve o establece los estilos de fuente en los siguientes formatos: <b>Negrita</b> , <i>Cursiva</i> , <del>Tachado</del> y <u>Subrayado</u> .
<i>FontName</i>	Devuelve o establece la fuente utilizada para mostrar texto en el control.
<i>FontSize</i>	Devuelve o establece el tamaño de la fuente que se va a usar para el texto mostrado en el control.
<i>FontTransparent</i>	Devuelve o establece un valor que determina si el texto y los gráficos de fondo de un objeto <i>Form</i> o <i>Printer</i> , o de un control <i>PictureBox</i> , se muestran en el espacio situado entre los caracteres.
<i>hDC</i>	Devuelve un controlador proporcionado por el entorno operativo Microsoft Windows para el contexto de dispositivo de un objeto.
<i>Height</i> y <i>Width</i>	Devuelven o establecen las dimensiones del control.
<i>HelpContextID</i>	Devuelve o establece un número de contexto asociado a un objeto. Se utiliza para proporcionar Ayuda interactiva para la aplicación.
<i>hWnd</i>	Devuelve el controlador de un formulario o un control. Nota: Esta propiedad no es compatible con el control contenedor OLE.
<i>Image</i>	Devuelve un controlador de un gráfico persistente; el controlador lo proporciona el entorno operativo Microsoft Windows.
<i>Index(Control Array)</i>	Devuelve o establece el número que identifica un control de forma exclusiva en una matriz de controles. Sólo está disponible si el control forma parte de una matriz de controles.
<i>Left</i> y <i>Top</i>	<i>Left</i> : devuelve o establece la distancia entre el borde interno izquierdo de un objeto y el borde izquierdo de su contenedor. <i>Top</i> : devuelve o establece la distancia entre el borde interno superior de un objeto y el borde superior de su contenedor.
<i>LinkItem</i>	Devuelve o establece los datos pasados a un control de destino en una conversación DDE con otra aplicación.
<i>LinkMode</i>	Devuelve o establece el tipo de vínculo usado en una conversación DDE y activa la conexión de la forma siguiente: Control: permite que un control de destino de un formulario de Visual Basic inicie una conversación, como se especifica en las propiedades <i>LinkTopic</i> y <i>LinkItem</i> .
<i>LinkTimeOut</i>	Devuelve o establece la cantidad de tiempo que un control espera una respuesta a un mensaje DDE.
<i>LinkTopic</i>	Para un control de destino, devuelve o establece la aplicación de origen y

Propiedades	Descripción
	el tema correspondiente (el agrupamiento de datos fundamental utilizado en esa aplicación). Utilice LinkTopic con la propiedad LinkItem para especificar el vínculo de datos completo. Para un formulario de origen, devuelve o establece el tema al que el formulario de origen responde en una conversación DDE.
<i>MouseIcon</i>	Devuelve o establece un icono personalizado de mouse (ratón).
<i>MousePointer</i>	Devuelve o establece un valor que indica el tipo de puntero de mouse que aparece cuando se sitúa el mouse sobre un área determinada de un objeto en tiempo de ejecución.
<i>Name</i>	Devuelve el nombre utilizado en el código para identificar un formulario, un control o un objeto de acceso a datos. Es de sólo lectura en tiempo de ejecución.
<i>Negotiate</i>	Establece un valor que determina si un control que puede alinearse se muestra cuando un objeto activo del formulario muestra una o más barras de herramientas. No está disponible en tiempo de ejecución.
<i>OLEDragMode</i>	Devuelve o establece un valor que indica si las operaciones OLE de arrastrar y colocar las controla el componente o el programador.
<i>OLEDropMode</i>	Devuelve o establece el modo en que un componente de destino realiza las operaciones de colocación.
<i>Parent</i>	Devuelve el formulario, objeto o colección que contiene al control.
<i>Picture</i>	Devuelve el formulario, objeto o colección que contiene al control u otro objeto o colección.
<i>RightToLeft</i>	El control CommandButton permite configurar el orden de lectura del título y proporciona compatibilidad bidireccional de idioma para el título del botón de comando. El orden predeterminado de lectura se hereda del valor de la propiedad RightToLeft del formulario primario; no obstante, es posible cambiar esta dirección mediante la propiedad RightToLeft del propio control CommandButton..
<i>ScaleHeight, ScaleWidth</i>	Devuelven o establecen el número de unidades de medida horizontal (ScaleWidth) y vertical (ScaleHeight) del interior de un objeto al usar métodos gráficos o al colocar controles. Para los objetos MDIForm, no está disponible en tiempo de diseño y es de sólo lectura en tiempo de ejecución.
<i>ScaleLeft, ScaleTop</i>	Devuelven o establecen las coordenadas horizontal (ScaleLeft) y vertical (ScaleTop) de los bordes izquierdo y superior de un objeto al usar métodos gráficos o al situar controles.
<i>ScaleMode</i>	Devuelve o establece un valor que indica la unidad de medida de las coordenadas de un objeto al usar métodos gráficos o al situar controles.
<i>TabIndex</i>	Devuelve o establece el orden de tabulación de la mayoría de los objetos dentro de su formulario primario.
<i>TabStop</i>	Devuelve o establece un valor que indica si el usuario puede usar la tecla TAB para llevar el enfoque a un objeto.
<i>Tag</i>	Devuelve o establece una expresión que almacena cualquier dato adicional que necesite el programa. Al contrario de lo que sucede con otras propiedades, Visual Basic no utiliza el valor de Tag. Puede usar esta propiedad para identificar objetos.
<i>ToolTipText</i>	Devuelve o establece Información sobre herramientas.

Propiedades	Descripción
<i>Visible</i>	Devuelve o establece un valor que indica si un objeto es visible o está oculto.
<i>WhatsThisHelpID</i>	Devuelve o establece un número de contexto asociado a un objeto. Se utiliza para dotar a las aplicaciones de Ayuda interactiva con el menú emergente ¿Qué es esto? De la Ayuda de Windows 95.

Ahora creamos otra aplicación un tanto más compleja: Consta de un formulario con dos cajas de texto, un botón de pulsación y siete cajas de imágenes. El diseño del formulario es el siguiente:



A continuación se muestran los controles y sus propiedades incluidos en el formulario:

Objeto	Propiedad	Valor
<i>Text1</i>	Name	txtTexto
<i>Text2</i>	Name	txtTamano
<i>CommandButton1</i>	Caption Name	Mostrar cmdMostrar
<i>PictureBox1</i>	Name BackColor	picRojo &H000000FF&
<i>PictureBox2</i>	Name BackColor	picNaranja &H000080FF&
<i>PictureBox3</i>	Name BackColor	picAmarillo &H0000FFFF&
<i>PictureBox4</i>	Name BackColor	picVerde &H0000FF00&
<i>PictureBox5</i>	Name BackColor	picAzul &H00FF0000&
<i>PictureBox6</i>	Name BackColor	picElegido &H00000000&
<i>PictureBox7</i>	Name	picSalida

El objeto de esta aplicación es escribir en la caja de texto *txtTexto*, elegir el estilo de la fuente y al hacer clic en *Imprimir*, lo ingresado en la caja de texto será copiado al control *pictureBox picSalida* en el formato elegido. Además, al seleccionar un color de los cinco disponibles se deberá cambiar el color al control *picElegido* por el seleccionado. Esto requiere que en el evento clic de cada uno de los controles *pictureBox* que muestran los colores disponibles se transfiera el color al *picElegida*.

```
Private Sub picRojo_Click()
```

```
PicElegido.BackColor = picRojo.BackColor  
End Sub
```

El código correspondiente al evento clic del botón de pulsación es el siguiente:

```
Private Sub cmdMostrar_Click()  
    picSalida.FontSize = Val(txtTamaño.Text)  
    picSalida.ForeColor = picElegido.BackColor  
    picSalida.Print txtTexto.Text  
End Sub
```

Por ejemplo si escribe Visual Basic en la caja de texto, selecciona un color y asigna un tamaño de texto, al pulsar mostrar la ventana se mostrará como en la figura siguiente:



## Frame (Marco)

Un marco, también denominado *caja de grupo*, permite identificar fácilmente un grupo de controles. Un marco tiene propiedades propias (por ejemplo, título, color, etc.), característica que aprovechamos cuando queremos realzar el aspecto de un formulario.

Para crear un marco, proceda como se indica a continuación:

1. Diríjase a la caja de herramientas y haga un doble clic sobre la herramienta marco, o bien seleccione esta herramienta utilizando la tecla *Tab* y pulse *Entrar*.
2. Ajuste el tamaño del marco y muévelo a la posición deseada. Puede hacerlo con el ratón o modificando las propiedades correspondientes.
3. Si lo desea, ponga un título al marco. Seleccione su propiedad *Caption* y escriba el título que desee.

Una vez creado el marco, puede añadir los controles que van a estar agrupados en el mismo. Para añadir un control, diríjase a la caja de herramientas, haga un clic sobre él, desplace cursor del ratón (+) al interior del marco y con el botón izquierdo del ratón pulsado, arrastre hasta dibujarlo.

Para garantizar que los controles pertenezcan al marco, usted no puede dibujar un control fuera del marco y después arrastrarlo dentro de él. Si hace esto, el control estaría encima del marco, no dentro de él.

Para seleccionar varios controles pertenecientes a un marco, mantenga pulsada la tecla *Mayús* o *Ctrl* y haga clic en cada uno de los controles, o bien arrastre el ratón con el botón izquierdo pulsado hasta rodear los controles deseados (para selecciones múltiples combine con las teclas *Mayús* o *Ctrl*).

## Agrupando botones

Un botón de opción siempre trabaja como parte de un grupo; esto es, cuando se selecciona uno, automáticamente quedan los demás sin seleccionar.

Todos los botones de opción que coloquemos directamente en un formulario forman un único grupo. Para crear diferentes grupos de botones de opción, hay que encerrar cada uno de ellos en un marco.

Eventos a los que responde el control *Frame*:

Eventos	Descripción
<i>Clic</i>	Ocurre cuando el usuario presiona y suelta un botón del mouse (ratón) en un objeto.
<i>DragDrop</i>	Ocurre cuando se completa una operación de arrastrar y colocar como resultado de arrastrar un control sobre un objeto y soltar el botón del mouse o usar el método <i>Drag</i> con su argumento acción establecido a 2 ( <i>Drop</i> ).
<i>DragOver</i>	Ocurre cuando una operación de arrastrar y colocar está en curso. Puede usar este evento para controlar el puntero del mouse a medida que entra, sale o descansa directamente sobre un destino válido. La posición del puntero del mouse determina el objeto de destino que recibe este evento.
<i>DbClick</i>	Se produce cuando un usuario presiona y suelta un botón del mouse (ratón) dos veces sobre un objeto.
<i>MouseDown</i> <i>MouseUp</i> <i>MouseMove</i>	Ocurren cuando el usuario presiona ( <i>MouseDown</i> ) o suelta ( <i>MouseUp</i> ) un botón del mouse. Ocurre cuando el usuario mueve el mouse.
<i>OLECompleteDrag</i>	Se produce cuando un componente de origen se coloca en un componente de destino e informa al componente de origen de que se ha realizado o cancelado una operación de arrastre.

Eventos	Descripción
<i>OLEDragDrop</i>	Se produce cuando un componente de origen se coloca en un componente de destino, si el componente de origen determina que la colocación puede realizarse. Nota: Este evento sólo se produce cuando <i>OLEDropMode</i> está establecido a 1 (Manual).
<i>OLEDragOver</i>	Se produce cuando un componente se arrastra sobre otro.
<i>OLEGiveFeedback</i>	Se produce después de cada evento <i>OLEDragOver</i> . <i>OLEGiveFeedback</i> permite al componente de origen proporcionar al usuario una indicación visual, como cambiar el cursor del mouse para indicar lo que ocurrirá si coloca el objeto o señalar visualmente la selección (en el componente de origen) para indicar lo que va a ocurrir.
<i>OLESetData</i>	Se produce en un componente de origen cuando un componente de destino realiza el método <i>GetData</i> sobre el objeto <i>DataObject</i> del origen, pero antes de que se carguen los datos del formato especificado.
<i>OLEStartDrag</i>	Se produce cuando se ejecuta el método <i>OLEDrag</i> de un componente o cuando un componente inicia una operación OLE de arrastrar y colocar, y la propiedad <i>OLEDragMode</i> está establecida a <i>Automatic</i> . Este evento especifica los formatos de los datos y los efectos de colocación que admite el componente de origen. También puede usarlo para insertar datos en el objeto <i>DataObject</i> .

Métodos correspondientes al control *Frame*:

Métodos	Descripción
<i>Drag</i>	Inicia, termina o cancela una operación de arrastre del control. No acepta argumentos con nombre.
<i>Move</i>	Mueve el control. No acepta argumentos con nombre.
<i>OLEDrag</i>	Hace que un componente inicie una operación OLE de arrastrar y colocar.
<i>Refresh</i>	Fuerza el volver a dibujar un formulario o un control completo.
<i>ShowWhatsThis</i>	Muestra un tema seleccionado de un archivo de Ayuda utilizando el menú emergente ¿Qué es esto? que ofrece la Ayuda de Windows 95.
<i>ZOrder</i>	Coloca al control al principio o al final del orden z dentro de su nivel gráfico. No acepta argumentos con nombre.

Propiedades correspondientes al control *Frame*:

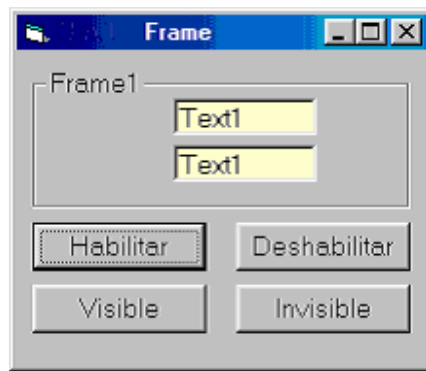
Propiedades	Descripción
<i>Appearance</i>	Devuelve o establece el estilo de dibujo de los controles de un objeto <i>MDIForm</i> o <i>Form</i> en tiempo de diseño. Es de sólo lectura en tiempo de ejecución.
<i>BackColor</i> y <i>ForeColor</i>	<i>BackColor</i> : devuelve o establece el color de fondo de un objeto. <i>ForeColor</i> : devuelve o establece el color de primer plano utilizado para mostrar texto y gráficos en un objeto.
<i>BorderStyle</i>	Devuelve o establece el estilo del borde de un objeto.
<i>Caption</i>	Determina el texto que se muestra en un control.

<b>Propiedades</b>	<b>Descripción</b>
<i>ClipControls</i>	Devuelve o establece un valor que determina si los métodos gráficos en eventos Paint vuelven a dibujar el objeto entero o sólo las áreas recién expuestas. También determina si el entorno operativo Microsoft Windows crea un área de recorte que excluya los controles no gráficos contenidos en el objeto. Es de sólo lectura en tiempo de ejecución.
<i>Container</i>	Devuelve o establece el contenedor de un control de un Form. No está disponible en tiempo de diseño.
<i>DragIcon</i>	Devuelve o establece el icono que se presenta como puntero en una operación de arrastrar y colocar.
<i>DragMode</i>	Devuelve o establece un valor que determina si se usa el modo de arrastre manual o automático en una operación de arrastrar y colocar.
<i>Enabled</i>	Devuelve o establece un valor que determina si un formulario o un control puede responder a eventos generados por el usuario.
<i>Font</i>	Devuelve un objeto Font.
<i>FontBold</i> , <i>FontItalic</i> , <i>FontStrikethru</i> y <i>FontUnderline</i>	Devuelve o establece los estilos de fuente en los siguientes formatos: Negrita, Cursiva, Tachado y <u>Subrayado</u> .
<i>FontName</i>	Devuelve o establece la fuente utilizada para mostrar texto en el control.
<i>FontSize</i>	Devuelve o establece el tamaño de la fuente que se va a usar para el texto mostrado en el control.
<i>Height</i> y <i>Width</i>	Devuelven o establecen las dimensiones del control.
<i>HelpContextID</i>	Devuelve o establece un número de contexto asociado a un objeto. Se utiliza para proporcionar Ayuda interactiva para la aplicación.
<i>HWND</i>	Devuelve el controlador de un formulario o un control. Nota: Esta propiedad no es compatible con el control contenedor OLE.
<i>Index(Control Array)</i>	Devuelve o establece el número que identifica un control de forma exclusiva en una matriz de controles. Sólo está disponible si el control forma parte de una matriz de controles.
<i>Left</i> y <i>Top</i>	Left: devuelve o establece la distancia entre el borde interno izquierdo de un objeto y el borde izquierdo de su contenedor. Top: devuelve o establece la distancia entre el borde interno superior de un objeto y el borde superior de su contenedor.
<i>MouseIcon</i>	Devuelve o establece un icono personalizado de mouse (ratón).
<i>MousePointer</i>	Devuelve o establece un valor que indica el tipo de puntero de mouse que aparece cuando se sitúa el mouse sobre un área determinada de un objeto en tiempo de ejecución.
<i>Name</i>	Devuelve el nombre utilizado en el código para identificar un formulario, un control o un objeto de acceso a datos. Es de sólo lectura en tiempo de ejecución.
<i>OLEDropMode</i>	Devuelve o establece el modo en que un componente de destino realiza las operaciones de colocación.
<i>Parent</i>	Devuelve el formulario, objeto o colección que contiene al control.



Propiedades	Descripción
<i>RightToLeft</i>	Devuelve un valor booleano que indica el sentido de presentación del texto y controla la apariencia visual en un sistema bidireccional.
<i>TabIndex</i>	Devuelve o establece el orden de tabulación de la mayoría de los objetos dentro de su formulario primario.
<i>Tag</i>	Devuelve o establece una expresión que almacena cualquier dato adicional que necesite el programa. Al contrario de lo que sucede con otras propiedades, Visual Basic no utiliza el valor de Tag. Puede usar esta propiedad para identificar objetos.
<i>ToolTipText</i>	Devuelve o establece Información sobre herramientas.
<i>Visible</i>	Devuelve o establece un valor que indica si un objeto es visible o está oculto.
<i>WhatsThisHelpID</i>	Devuelve o establece un número de contexto asociado a un objeto. Se utiliza para dotar a las aplicaciones de Ayuda interactiva con el menú emergente ¿Qué es esto? de la Ayuda de Windows 95.

Creemos una nueva aplicación para aplicar estos conocimientos. La aplicación estará compuesta de un formulario el cual contendrá un *frame*, dos cajas de texto dentro de él y cuatro botones de pulsación. El diseño del formulario es el siguiente.



Por medio de este ejemplo observaremos cómo funciona un marco como agrupador de controles. Los botones de pulsación habilitarán, deshabilitarán, harán visible e invisible al marco *Frame1*. Es código correspondiente a cada botón de pulsación se muestra a continuación.

```
Private Sub cmdDeshabilitar_Click()
    Frame1.Enabled = False
End Sub

Private Sub cmdHabilitar_Click()
    Frame1.Enabled = True
End Sub

Private Sub cmdInvisible_Click()
    Frame1.Visible = False
End Sub

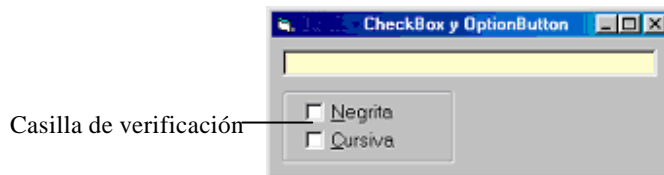
Private Sub cmdVisible_Click()
    Frame1.Visible = True
End Sub
```

Con esto podrá observar que al deshabilitar el marco, las cajas de texto incluidas en él también son deshabilitadas, es decir que los controles agrupados en un marco dependen de dicho marco y cuando sus propiedades son alteradas tienen efecto sobre los controles contenidos.

### Casillas de verificación (CheckBox)

Una *casilla de verificación* es un control que indica si una opción particular está activada o desactivada. Cada casilla de verificación es independiente de las demás, ya que cada una de ellas tiene su propio nombre (propiedad *Name*). El número de opciones representadas de esta forma puede ser cualquiera y el usuario puede seleccionar todas las que desee cada vez.

Si en tiempo de ejecución se hace clic en una casilla de verificación, la opción queda seleccionada (). Una opción ya seleccionada puede pasar a no estarlo haciendo clic de nuevo sobre la casilla de verificación correspondiente (). Todo esto suponiendo que el convenio adoptado es: , opción seleccionada, y , opción no seleccionada.



Para saber si una determinada opción está seleccionada, hay que verificar el valor de su propiedad *Value*. Este valor puede ser: 0, la casilla aparece vacía; 1, la casilla aparece con una , y 2, la casilla aparece en gris .

También se puede inhabilitar una casilla de verificación (aparece en gris) poniendo su propiedad *Enabled* a valor *False*.

Por ejemplo, diseñe una caja de diálogo como la de la figura anterior, de tal forma que cuando el usuario seleccione la opción "Negrita", todo el texto que haya escrito en la caja de texto *txtTexto* aparezca en negrita y cuando seleccione "Cursiva" se le aplique dicho formato. Inserte un & antes del carácter que desea que aparezca subrayado en el título de la casilla de verificación, por ejemplo antes de la N, y podrá activar o desactivar la opción pulsando *Alt + N*.

Asigne a la caja de texto el nombre *txtTexto*, a la primera casilla de verificación *chkNegrita* y a la segunda *chkCursiva*.

Cuando el usuario seleccione la opción "Negrita", haciendo clic, pulsando *Alt* más el carácter que aparece subrayado o eligiéndola con la tecla *Tab* y pulsando espacio, la casilla de verificación reconoce el evento *Click*. Por lo tanto, será el procedimiento conducido por este evento el que convierta a negrita y a cursiva el contenido de la caja de texto.

```
Private Sub chkNegrita_Click()
    If chkNegrita.Value = 1 Then
        txtTexto.FontBold = True
    Else
        txtTexto.FontBold = False
    End If
End Sub

Private Sub chkCursiva_Click()
    If chkCursiva.Value = 1 Then
        txtTexto.FontItalic = True
    Else
        txtTexto.FontItalic = False
    End If
End Sub
```

Cuando el usuario haga clic sobre la opción "Negrita" y la propiedad *Value* valga 1, el texto de la caja *txtTexto* se convierte a negrita utilizando la función *txtTexto.FontBold = True*.

Un razonamiento análogo corresponde para la opción "Cursiva".

Eventos a los que responde el control *CheckBox*:

<b>Eventos</b>	<b>Descripción</b>
<i>Clic</i>	Ocurre cuando el usuario presiona y suelta un botón del mouse (ratón) en un objeto. También puede ocurrir cuando se cambia el valor de un control. Para un <i>CheckBox</i> , este evento ocurre cuando el usuario: Presiona la BARRA ESPACIADORA cuando un control <i>CheckBox</i> tiene el enfoque.
<i>DragDrop</i>	Ocurre cuando se completa una operación de arrastrar y colocar como resultado de arrastrar un control sobre un objeto y soltar el botón del mouse o usar el método <i>Drag</i> con su argumento acción establecido a 2 ( <i>Drop</i> ).
<i>DragOver</i>	Ocurre cuando una operación de arrastrar y colocar está en curso. Puede usar este evento para controlar el puntero del mouse a medida que entra, sale o descansa directamente sobre un destino válido. La posición del puntero del mouse determina el objeto de destino que recibe este evento.
<i>GotFocus</i>	Ocurre cuando un objeto recibe el enfoque, ya sea mediante una acción del usuario, como tabular o hacer clic en el objeto, o cambiando el enfoque en el código mediante el método <i>SetFocus</i> . Un formulario recibe el enfoque sólo cuando todos los controles visibles están desactivados.
<i>KeyDown</i> y <i>KeyUp</i>	Ocurren cuando el usuario presiona ( <i>KeyDown</i> ) o suelta ( <i>KeyUp</i> ) una tecla mientras un objeto tiene el enfoque. (Para interpretar los caracteres ANSI, utilice el evento <i>KeyPress</i> .)
<i>KeyPress</i>	Ocurre cuando el usuario presiona y suelta una tecla ANSI.
<i>LostFocus</i>	Ocurre cuando un objeto pierde el enfoque, ya sea por una acción del usuario, como tabular o hacer clic en otro objeto, o bien mediante un cambio del enfoque en el código con el método <i>SetFocus</i> .
<i>MouseDown</i> y <i>MouseUp</i>	Ocurren cuando el usuario presiona ( <i>MouseDown</i> ) o suelta ( <i>MouseUp</i> ) un botón del mouse.
<i>MouseMove</i>	Ocurre cuando el usuario mueve el mouse.
<i>OLECompleteDrag</i>	Se produce cuando un componente de origen se coloca en un componente de destino e informa al componente de origen de que se ha realizado o cancelado una operación de arrastre.
<i>OLEDragDrop</i>	Se produce cuando un componente de origen se coloca en un componente de destino, si el componente de origen determina que la colocación puede realizarse. Nota: Este evento sólo se produce cuando <i>OLEDropMode</i> está establecido a 1 ( <i>Manual</i> ).
<i>OLEDragOver</i>	Se produce cuando un componente se arrastra sobre otro.
<i>OLEGiveFeedback</i>	Se produce después de cada evento <i>OLEDragOver</i> . <i>OLEGiveFeedback</i> permite al componente de origen proporcionar al usuario una indicación visual, como cambiar el cursor del mouse para indicar lo que ocurrirá si coloca el objeto o señalar visualmente la selección (en el componente de origen) para indicar lo que va a ocurrir.
<i>OLESetData</i>	Se produce en un componente de origen cuando un componente de destino realiza el método <i>GetData</i> sobre el objeto <i>DataObject</i> del origen, pero antes de que se carguen los datos del formato especificado.

Eventos	Descripción
<i>OLEStartDrag</i>	Se produce cuando se ejecuta el método <i>OLEDrag</i> de un componente o cuando un componente inicia una operación OLE de arrastrar y colocar, y la propiedad <i>OLEDragMode</i> está establecida a <i>Automatic</i> . Este evento especifica los formatos de los datos y los efectos de colocación que admite el componente de origen. También puede usarlo para insertar datos en el objeto <i>DataObject</i> .
<i>Validate</i>	Ocurre cuando el foco cambia a un (segundo) control que tiene su propiedad <i>CausesValidation</i> establecida a <i>True</i> .

Métodos correspondientes al control *CheckBox*:

Métodos	Descripción
<i>Drag</i>	Inicia, termina o cancela una operación de arrastre del control. No acepta argumentos con nombre.
<i>Move</i>	Mueve el control. No acepta argumentos con nombre.
<i>OLEDrag</i>	Hace que un componente inicie una operación OLE de arrastrar y colocar.
<i>Refresh</i>	Fuerza el volver a dibujar un formulario o un control completo.
<i>SetFocus</i>	Mueve el enfoque al control o formulario especificado.
<i>ShowWhatsThis</i>	Muestra un tema seleccionado de un archivo de Ayuda utilizando el menú emergente ¿Qué es esto? que ofrece la Ayuda de Windows 95.
<i>ZOrder</i>	Coloca al control al principio o al final del orden z dentro de su nivel gráfico. No acepta argumentos con nombre.

Propiedades correspondientes al control *CheckBox*:

Propiedades	Descripción
<i>Alignment</i>	Devuelve o establece un valor que determina la alineación de un control <i>CheckBox</i> u <i>OptionButton</i> , del texto de un control. Es de sólo lectura en tiempo de ejecución para los controles <i>CheckBox</i> , <i>OptionButton</i> y <i>TextBox</i> .
<i>Appearance</i>	Devuelve o establece el estilo de dibujo de los controles de un objeto <i>MDIForm</i> o <i>Form</i> en tiempo de diseño. Es de sólo lectura en tiempo de ejecución.
<i>BackColor</i> y <i>ForeColor</i>	<i>BackColor</i> : devuelve o establece el color de fondo de un objeto. <i>ForeColor</i> : devuelve o establece el color de primer plano utilizado para mostrar texto y gráficos en un objeto.
<i>Caption</i>	Determina el texto que se muestra en un control.
<i>Container</i>	Devuelve o establece el contenedor de un control de un <i>Form</i> . No está disponible en tiempo de diseño.
<i>DataChanded</i>	Devuelve o establece un valor que indica que algún proceso distinto de la recuperación de datos del registro actual ha cambiado los datos del control enlazado. No está disponible en tiempo de diseño.
<i>DataField</i>	Devuelve o establece el nombre de un campo al que será enlazado un receptor de datos.

<b>Propiedades</b>	<b>Descripción</b>
<i>DataFormat</i>	Devuelve o establece el objeto StdDataFormat al que está adjunto un objeto vinculado. Lectura y escritura en tiempo de diseño y en tiempo de ejecución.
<i>DataMember</i>	Devuelve o establece un miembro de datos especificado entre los varios que ofrece el proveedor de datos.
<i>DisabledPicture</i>	Devuelve o establece una referencia a una imagen que se va a mostrar en un control cuando está inactivo. (Es decir, cuando su propiedad Enabled se establece a False).
<i>DownPicture</i>	Devuelve o establece una referencia a una imagen que se muestra en un control cuando se hace clic en él y está presionado.
<i>DragIcon</i>	Devuelve o establece el icono que se presenta como puntero en una operación de arrastrar y colocar.
<i>DragMode</i>	Devuelve o establece un valor que determina si se usa el modo de arrastre manual o automático en una operación de arrastrar y colocar.
<i>Enabled</i>	Devuelve o establece un valor que determina si un formulario o un control puede responder a eventos generados por el usuario.
<i>Font</i>	Devuelve un objeto Font.
<i>FontBold</i> , <i>FontItalic</i> , <i>FontStrikethru</i> y <i>FontUnderline</i>	Devuelve o establece los estilos de fuente en los siguientes formatos: Negrita, Cursiva, <del>Tachado</del> y <u>Subrayado</u> .
<i>FontName</i>	Devuelve o establece la fuente utilizada para mostrar texto en el control.
<i>FontSize</i>	Devuelve o establece el tamaño de la fuente que se va a usar para el texto mostrado en el control.
<i>Height</i> y <i>Width</i>	Devuelven o establecen las dimensiones del control.
<i>HelpContextID</i>	Devuelve o establece un número de contexto asociado a un objeto. Se utiliza para proporcionar Ayuda interactiva para la aplicación.
<i>HWND</i>	Devuelve el controlador de un formulario o un control. Nota: Esta propiedad no es compatible con el control contenedor OLE.
<i>Index(Control Array)</i>	Devuelve o establece el número que identifica un control de forma exclusiva en una matriz de controles. Sólo está disponible si el control forma parte de una matriz de controles.
<i>Left</i> y <i>Top</i>	Left: devuelve o establece la distancia entre el borde interno izquierdo de un objeto y el borde izquierdo de su contenedor. Top: devuelve o establece la distancia entre el borde interno superior de un objeto y el borde superior de su contenedor.
<i>MaskColor</i>	Devuelve o establece un color de la imagen del botón que actuará de "máscara" (es decir, será transparente).
<i>MouseIcon</i>	Devuelve o establece un icono personalizado de mouse (ratón).
<i>MousePointer</i>	Devuelve o establece un valor que indica el tipo de puntero de mouse que aparece cuando se sitúa el mouse sobre un área determinada de un objeto en tiempo de ejecución.

<b>Propiedades</b>	<b>Descripción</b>
<i>Name</i>	Devuelve el nombre utilizado en el código para identificar un formulario, un control o un objeto de acceso a datos. Es de sólo lectura en tiempo de ejecución.
<i>OLEDropMode</i>	Devuelve o establece el modo en que un componente de destino realiza las operaciones de colocación.
<i>Parent</i>	Devuelve el formulario, objeto o colección que contiene al control.
<i>Picture</i>	Devuelve el formulario, objeto o colección que contiene al control u otro objeto o colección.
<i>RightToLeft</i>	Devuelve un valor booleano que indica el sentido de presentación del texto y controla la apariencia visual en un sistema bidireccional.
<i>Style</i>	Devuelve o establece un valor que indica el tipo de muestra y el comportamiento del control. Es de sólo lectura en tiempo de ejecución.
<i>TabIndex</i>	Devuelve o establece el orden de tabulación de la mayoría de los objetos dentro de su formulario primario.
<i>TabStop</i>	Devuelve o establece un valor que indica si el usuario puede usar la tecla TAB para llevar el enfoque a un objeto.
<i>Tag</i>	Devuelve o establece una expresión que almacena cualquier dato adicional que necesite el programa. Al contrario de lo que sucede con otras propiedades, Visual Basic no utiliza el valor de Tag. Puede usar esta propiedad para identificar objetos.
<i>ToolTipText</i>	Devuelve o establece Información sobre herramientas.
<i>UseMaskColor</i>	Devuelve o establece un valor que determina si el color asignado en la propiedad MaskColor se usa como "máscara", es decir, se usa para crear zonas transparentes.
<i>Value</i>	Devuelve o establece el estado del control.
<i>Visible</i>	Devuelve o establece un valor que indica si un objeto es visible o está oculto.
<i>WhatsThisHelpID</i>	Devuelve o establece un número de contexto asociado a un objeto. Se utiliza para dotar a las aplicaciones de Ayuda interactiva con el menú emergente ¿Qué es esto? de la Ayuda de Windows 95.

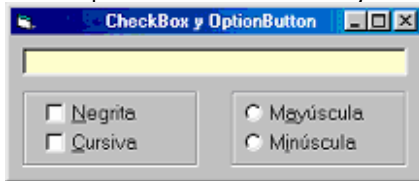
**Botón de opción (OptionButton)**

Un botón de opción es un control que indica si una determinada opción está activada o desactivada. Cada botón de opción es independiente de los demás, ya que cada uno de ellos tiene su propio nombre (propiedad *Name*). El número de opciones representadas de esta forma puede ser cualquiera, y de ellas el usuario sólo puede seleccionar una cada vez.

Si durante la ejecución se hace clic sobre un botón de opción, la opción queda seleccionada (☑). La selección de una opción de este tipo provoca que si hay otra opción del mismo grupo actualmente seleccionada pase a no estarlo.

Para saber si una determinada opción está seleccionada, hay que verificar el valor de su propiedad *Value*. Este valor puede ser falso (*False*), el botón aparece vacío, o verdadero (*True*), el botón aparece con un ☑. Cuando se hace clic sobre un botón de opción, se da el evento *Click*. Cuando una de estas opciones está deshabilitada, la etiqueta asociada aparece en gris. Esto se consigue poniendo su propiedad *Enabled* al valor *False*.

Por ejemplo, retomando la aplicación anterior, agregue al formulario dos botones de opción de forma que el diseño se asemeje a la siguiente figura.

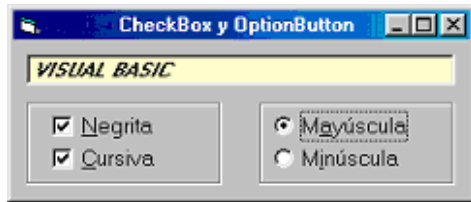


Cuando seleccione uno de los botones de opción "Mayúscula" o "Minúscula", el contenido de la caja de texto *txtTexto* se convertirá según la selección realizada. Asigne a la propiedad *Name* el valor de *optMayuscula* para el primer botón de opción y *optMinuscula* para el segundo. El código correspondiente al evento Clic de cada botón de opción es el siguiente:

```
Private Sub optMayuscula_Click()
    If optMayuscula.Value = True Then
        txtTexto.Text = UCase(txtTexto.Text)
    End If
End Sub

Private Sub optMinuscula_Click()
    If optMinuscula.Value = True Then
        txtTexto.Text = LCase(txtTexto.Text)
    End If
End Sub
```

Ejecute la aplicación y observe los resultados.



Eventos a los que responde el control *OptionButton*:

Eventos	Descripción
<i>Clic</i>	Ocurre cuando el usuario presiona y suelta un botón del mouse (ratón) en un objeto. También puede ocurrir cuando se cambia el valor de un control. Para un <i>OptionButton</i> , este evento ocurre cuando el usuario: Presiona la BARRA ESPACIADORA cuando un control <i>CheckBox</i> tiene el enfoque.
<i>DragDrop</i>	Ocurre cuando se completa una operación de arrastrar y colocar como resultado de arrastrar un control sobre un objeto y soltar el botón del mouse o usar el método <i>Drag</i> con su argumento acción establecido a 2 ( <i>Drop</i> ).

<b>Eventos</b>	<b>Descripción</b>
<i>DragOver</i>	Ocurre cuando una operación de arrastrar y colocar está en curso. Puede usar este evento para controlar el puntero del mouse a medida que entra, sale o descansa directamente sobre un destino válido. La posición del puntero del mouse determina el objeto de destino que recibe este evento.
<i>DbClick</i>	Se produce cuando un usuario presiona y suelta un botón del mouse (ratón) dos veces sobre un objeto.
<i>GotFocus</i>	Ocurre cuando un objeto recibe el enfoque, ya sea mediante una acción del usuario, como tabular o hacer clic en el objeto, o cambiando el enfoque en el código mediante el método <code>SetFocus</code> . Un formulario recibe el enfoque sólo cuando todos los controles visibles están desactivados.
<i>KeyDown</i> y <i>KeyUp</i>	Ocurren cuando el usuario presiona ( <code>KeyDown</code> ) o suelta ( <code>KeyUp</code> ) una tecla mientras un objeto tiene el enfoque. (Para interpretar los caracteres ANSI, utilice el evento <code>KeyPress</code> .)
<i>KeyPress</i>	Ocurre cuando el usuario presiona y suelta una tecla ANSI.
<i>LostFocus</i>	Ocurre cuando un objeto pierde el enfoque, ya sea por una acción del usuario, como tabular o hacer clic en otro objeto, o bien mediante un cambio del enfoque en el código con el método <code>SetFocus</code> .
<i>MouseDown</i> y <i>MouseUp</i>	Ocurren cuando el usuario presiona ( <code>MouseDown</code> ) o suelta ( <code>MouseUp</code> ) un botón del mouse.
<i>MouseMove</i>	Ocurre cuando el usuario mueve el mouse.
<i>OLECompleteDrag</i>	Se produce cuando un componente de origen se coloca en un componente de destino e informa al componente de origen de que se ha realizado o cancelado una operación de arrastre.
<i>OLEDragDrop</i>	Se produce cuando un componente de origen se coloca en un componente de destino, si el componente de origen determina que la colocación puede realizarse. Nota: Este evento sólo se produce cuando <code>OLEDropMode</code> está establecido a 1 (Manual).
<i>OLEDragOver</i>	Se produce cuando un componente se arrastra sobre otro.
<i>OLEGiveFeedback</i>	Se produce después de cada evento <code>OLEDragOver</code> . <code>OLEGiveFeedback</code> permite al componente de origen proporcionar al usuario una indicación visual, como cambiar el cursor del mouse para indicar lo que ocurrirá si coloca el objeto o señalar visualmente la selección (en el componente de origen) para indicar lo que va a ocurrir.
<i>OLESetData</i>	Se produce en un componente de origen cuando un componente de destino realiza el método <code>GetData</code> sobre el objeto <code>DataObject</code> del origen, pero antes de que se carguen los datos del formato especificado.
<i>OLEStartDrag</i>	Se produce cuando se ejecuta el método <code>OLEDrag</code> de un componente o cuando un componente inicia una operación OLE de arrastrar y colocar, y la propiedad <code>OLEDragMode</code> está establecida a <code>Automatic</code> . Este evento especifica los formatos de los datos y los efectos de colocación que admite el componente de origen. También puede usarlo para insertar datos en el objeto <code>DataObject</code> .
<i>Validate</i>	Ocurre cuando el foco cambia a un (segundo) control que tiene su propiedad <code>CausesValidation</code> establecida a <code>True</code> .



Métodos correspondientes al control *OptionButton*:

Métodos	Descripción
<i>Drag</i>	Inicia, termina o cancela una operación de arrastre del control. No acepta argumentos con nombre.
<i>Move</i>	Mueve el control. No acepta argumentos con nombre.
<i>OLEDrag</i>	Hace que un componente inicie una operación OLE de arrastrar y colocar.
<i>Refresh</i>	Fuerza el volver a dibujar un formulario o un control completo.
<i>SetFocus</i>	Mueve el enfoque al control o formulario especificado.
<i>ShowWhatsThis</i>	Muestra un tema seleccionado de un archivo de Ayuda utilizando el menú emergente ¿Qué es esto? que ofrece la Ayuda de Windows 95.
<i>Zorder</i>	Coloca al control al principio o al final del orden z dentro de su nivel gráfico. No acepta argumentos con nombre.

Propiedades correspondientes al control *OptionButton*:

Propiedades	Descripción
<i>Alignment</i>	Devuelve o establece un valor que determina la alineación de un control <i>CheckBox</i> u <i>OptionButton</i> , del texto de un control. Es de sólo lectura en tiempo de ejecución para los controles <i>CheckBox</i> , <i>OptionButton</i> y <i>TextBox</i> .
<i>Appearance</i>	Devuelve o establece el estilo de dibujo de los controles de un objeto <i>MDIForm</i> o <i>Form</i> en tiempo de diseño. Es de sólo lectura en tiempo de ejecución.
<i>BackColor</i> y <i>ForeColor</i>	<i>BackColor</i> : devuelve o establece el color de fondo de un objeto. <i>ForeColor</i> : devuelve o establece el color de primer plano utilizado para mostrar texto y gráficos en un objeto.
<i>Caption</i>	Determina el texto que se muestra en un control.
<i>Container</i>	Devuelve o establece el contenedor de un control de un <i>Form</i> . No está disponible en tiempo de diseño.
<i>DataFormat</i>	Devuelve o establece el objeto <i>StdDataFormat</i> al que está adjunto un objeto vinculado. Lectura y escritura en tiempo de diseño y en tiempo de ejecución.
<i>DisabledPicture</i>	Devuelve o establece una referencia a una imagen que se va a mostrar en un control cuando está inactivo. (Es decir, cuando su propiedad <i>Enabled</i> se establece a <i>False</i> ).
<i>DownPicture</i>	Devuelve o establece una referencia a una imagen que se muestra en un control cuando se hace clic en él y está presionado.
<i>DragIcon</i>	Devuelve o establece el icono que se presenta como puntero en una operación de arrastrar y colocar.
<i>DragMode</i>	Devuelve o establece un valor que determina si se usa el modo de arrastre manual o automático en una operación de arrastrar y colocar.
<i>Enabled</i>	Devuelve o establece un valor que determina si un formulario o un control puede responder a eventos generados por el usuario.
<i>Font</i>	Devuelve un objeto <i>Font</i> .

<b>Propiedades</b>	<b>Descripción</b>
<i>FontBold</i> , <i>FontItalic</i> , <i>FontStrikethru</i> y <i>FontUnderline</i>	Devuelve o establece los estilos de fuente en los siguientes formatos: Negrita, Cursiva, <del>Tachado</del> y <u>Subrayado</u> .
<i>FontName</i>	Devuelve o establece la fuente utilizada para mostrar texto en el control.
<i>FontSize</i>	Devuelve o establece el tamaño de la fuente que se va a usar para el texto mostrado en el control.
<i>Height</i> y <i>Width</i>	Devuelven o establecen las dimensiones del control.
<i>HelpContextID</i>	Devuelve o establece un número de contexto asociado a un objeto. Se utiliza para proporcionar Ayuda interactiva para la aplicación.
<i>HWnd</i>	Devuelve el controlador de un formulario o un control. Nota: Esta propiedad no es compatible con el control contenedor OLE.
<i>Index(Control Array)</i>	Devuelve o establece el número que identifica un control de forma exclusiva en una matriz de controles. Sólo está disponible si el control forma parte de una matriz de controles.
<i>Left</i> y <i>Top</i>	<i>Left</i> : devuelve o establece la distancia entre el borde interno izquierdo de un objeto y el borde izquierdo de su contenedor. <i>Top</i> : devuelve o establece la distancia entre el borde interno superior de un objeto y el borde superior de su contenedor.
<i>MaskColor</i>	Devuelve o establece un color de la imagen del botón que actuará de "máscara" (es decir, será transparente).
<i>MouseIcon</i>	Devuelve o establece un icono personalizado de mouse (ratón).
<i>MousePointer</i>	Devuelve o establece un valor que indica el tipo de puntero de mouse que aparece cuando se sitúa el mouse sobre un área determinada de un objeto en tiempo de ejecución.
<i>Name</i>	Devuelve el nombre utilizado en el código para identificar un formulario, un control o un objeto de acceso a datos. Es de sólo lectura en tiempo de ejecución.
<i>OLEDropMode</i>	Devuelve o establece el modo en que un componente de destino realiza las operaciones de colocación.
<i>Parent</i>	Devuelve el formulario, objeto o colección que contiene al control.
<i>Picture</i>	Devuelve el formulario, objeto o colección que contiene al control u otro objeto o colección.
<i>RightToLeft</i>	Devuelve un valor booleano que indica el sentido de presentación del texto y controla la apariencia visual en un sistema bidireccional.
<i>Style</i>	Devuelve o establece un valor que indica el tipo de muestra y el comportamiento del control. Es de sólo lectura en tiempo de ejecución.
<i>TabIndex</i>	Devuelve o establece el orden de tabulación de la mayoría de los objetos dentro de su formulario primario.
<i>TabStop</i>	Devuelve o establece un valor que indica si el usuario puede usar la tecla TAB para llevar el enfoque a un objeto.
<i>Tag</i>	Devuelve o establece una expresión que almacena cualquier dato adicional que necesite el programa. Al contrario de lo que sucede con otras propiedades, Visual Basic no utiliza el valor de Tag. Puede usar

<b>Propiedades</b>	<b>Descripción</b>
	esta propiedad para identificar objetos.
<i>ToolTipText</i>	Devuelve o establece Información sobre herramientas.
<i>UseMaskColor</i>	Devuelve o establece un valor que determina si el color asignado en la propiedad MaskColor se usa como "máscara", es decir, se usa para crear zonas transparentes.
<i>Value</i>	Devuelve o establece el estado del control.
<i>Visible</i>	Devuelve o establece un valor que indica si un objeto es visible o está oculto.
<i>WhatsThisHelpID</i>	Devuelve o establece un número de contexto asociado a un objeto. Se utiliza para dotar a las aplicaciones de Ayuda interactiva con el menú emergente ¿Qué es esto? de la Ayuda de Windows 95.

## Listas (ListBox)

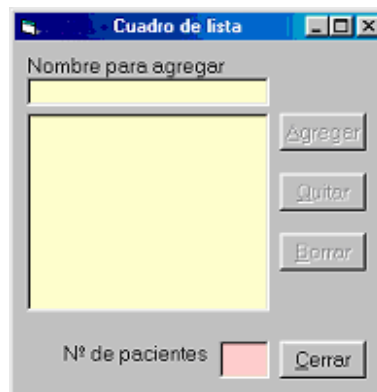
Una *lista* es un control que pone a disposición del usuario un conjunto de elementos, de los cuales elegirá uno. Si la cantidad de elementos rebasa el número de los que pueden ser visualizados simultáneamente en el espacio disponible en la lista, aparecerá automáticamente una barra de desplazamiento vertical para que el usuario pueda desplazar los elementos de la lista hacia arriba o hacia abajo. Generalmente, una lista es apropiada cuando se quiere limitar la entrada a una serie de elementos determinados.

Por omisión, los elementos de una lista son visualizados verticalmente en una a columna, aunque usted puede, si quiere, establecer múltiples columnas (sólo a en listas simples no en listas desplegables, propiedad *Columns*). Si el ancho total las columnas supera el ancho de la lista, aparecerá automáticamente una barra de desplazamiento horizontal para que el usuario pueda desplazar las columnas hacia la izquierda o hacia la derecha.

Una *lista desplegable* es un control que combina las características de una caja de texto y de una lista. Esto permite al usuario elegir un elemento de varios, escribiéndolo directamente en la caja de texto o seleccionándolo de la lista. Generalmente, un control combinado es apropiado cuando hay una lista de elemento sugeridos; además, el usuario puede introducir otros que no estén en la lista.

### Utilización de listas

Nuestra aplicación va a constar de una ventana principal que permita introducir, visualizar o borrar los datos de ingresados en una caja de texto y transferidos a una lista con el fin de conformar una lista de los pacientes que son atendidos en la clínica odontológica Smile.



Para diseñar esta aplicación, comience un nuevo proyecto. A continuación añada los controles que se especifican en la tabla siguiente:

Control	Propiedad	Valor
<i>Label1</i>	Caption	Nombre para agregar
<i>Text1</i>	Name	txtNombre
<i>ListBox1</i>	Name	IstPacientes
<i>CommandButton1</i>	Name Caption Enabled	cmdAgregar Agregar False
<i>CommandButton2</i>	Name Caption Enabled	cmdQuitar Quitar False
<i>CommandButton3</i>	Name Caption Enabled	cmdBorrar Borrar False
<i>CommandButton4</i>	Name Caption	cmdCerrar Cerrar

<i>Label2</i>	Caption	Nº de pacientes
<i>Label3</i>	Name Caption	IbINroPacientes (nada)

Al iniciar la ejecución de la aplicación notará que el único botón habitado es el botón *Cerrar*. Cuando el usuario ingrese el nombre del cliente en la caja de texto *txtNombre*, el botón *Agregar* será habilitado. Esto quiere decir que hasta que se escriba una letra dentro de la caja de texto no podrá usar dicho botón. Entonces escribamos el código correspondiente al evento *KeyPress* de la caja de texto *txtNombre*:

```
Private Sub txtNombre_KeyPress(KeyAscii As Integer)
    cmdAgregar.Enabled = True
End Sub
```

### Agregar un elemento a la lista

A continuación vamos a escribir el código correspondiente al botón *Agregar*. Cuando el usuario haga clic en este botón, deseamos que el nombre del cliente ingresado en la caja de texto sea transferido a la lista de clientes.

```
Private Sub cmdAgregar_Click()
    IstPacientes.AddItem txtNombre.Text
    txtNombre.Text = ""
    txtNombre.SetFocus
    IbINroPacientes.Caption = IstPacientes.ListCount
    cmdBorrar.Enabled = True
    cmdAgregar.Enabled = False
End Sub
```

El método *AddItem* permite añadir un elemento a una lista o lista desplegable. Su sintaxis es la siguiente:

*Nombre\_lista.AddItem elemento[, indice]*

Luego de agregar un elemento a la lista se asigna "" a la propiedad *Text* de la caja de texto y se le coloca el cursor, esto se realiza para facilitar el ingreso del siguiente paciente. La sentencia *IbINroPacientes.Caption = IstPacientes.ListCount*, asigna a la etiqueta *IbINroPacientes* la cantidad de elementos con los que actualmente cuenta la lista.

### Eliminar un elemento de una lista

Para eliminar un elemento de una lista o de una lista desplegable, se utiliza el método *RemoveItem*. La sintaxis es la siguiente:

*Nombre\_lista.RemoveItem indice*

donde *nombre\_lista* es el nombre del control (lista o lista desplegable) y argumento *indice* indica la posición del elemento que se desea eliminar. Un valor 0 indica la primera posición.

Cuando el usuario pulse el botón *Quitar*, el elemento seleccionado tiene que borrarse de la lista *IstPacientes*.

Para borrar el elemento seleccionado de la lista tenemos que agregar el código siguiente al evento *Clic* del botón *Quitar*, recuerde que deberá calcular nuevamente la cantidad de elementos de la lista para asignarlo a la etiqueta correspondiente:

```
Private Sub cmdQuitar_Click()
    Dim indice As Integer
    indice = IstPacientes.ListIndex
    If indice >= 0 Then
        IstPacientes.RemoveItem indice
        IbINroPacientes.Caption = IstPacientes.ListCount
    Else
        cmdQuitar.Enabled = False
    End If
```

End Sub

### Borrar los elementos de una lista

Para borrar los elementos de una lista o de una lista desplegable, se utiliza el método *Clear*. La sintaxis es la siguiente:

*Nombre\_lista.Clear*

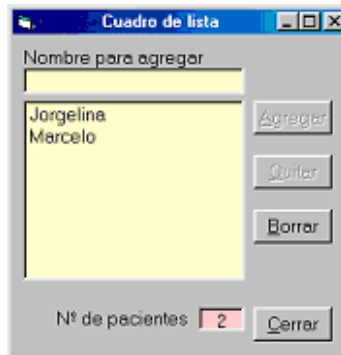
donde *nombre-lista* es el nombre del control (lista o lista desplegable).

Escribamos entonces el código correspondiente al evento Clic del botón *Borrar*:

```
Private Sub cmdBorrar_Click()
    IstPacientes.Clear
    cmdQuitar.Enabled = False
    lblNroPacientes.Caption = 0
End Sub
```

Aquí debemos asignar a la etiqueta *lblNroPacientes* el número 0 puesto que la lista ya no contiene elementos.

Al ejecutar la aplicación se mostrará como sigue:



### Eventos, métodos y propiedades del control ListBox

Eventos	Descripción
<i>Clic</i>	Ocurre cuando el usuario presiona y suelta un botón del mouse (ratón) en un objeto.
<i>DragDrop</i>	Ocurre cuando se completa una operación de arrastrar y colocar como resultado de arrastrar un control sobre un objeto y soltar el botón del mouse o usar el método Drag con su argumento acción establecido a 2 (Drop).
<i>DragOver</i>	Ocurre cuando una operación de arrastrar y colocar está en curso. Puede usar este evento para controlar el puntero del mouse a medida que entra, sale o descansa directamente sobre un destino válido. La posición del puntero del mouse determina el objeto de destino que recibe este evento.
<i>DbiClick</i>	Se produce cuando un usuario presiona y suelta un botón del mouse (ratón) dos veces sobre un objeto.
<i>GotFocus</i>	Ocurre cuando un objeto recibe el enfoque, ya sea mediante una acción del usuario, como tabular o hacer clic en el objeto, o cambiando el enfoque en el código mediante el método SetFocus. Un formulario recibe el enfoque sólo cuando todos los controles visibles están desactivados.

<b>Eventos</b>	<b>Descripción</b>
<i>ItemCheck</i>	Se produce cuando la propiedad <i>Style</i> de un control <i>ListBox</i> se establece a 1 (casillas de verificación) y se activa o se desactiva la casilla de verificación de un elemento del control <i>ListBox</i> .
<i>KeyDown</i> y <i>KeyUp</i>	Ocurren cuando el usuario presiona ( <i>KeyDown</i> ) o suelta ( <i>KeyUp</i> ) una tecla mientras un objeto tiene el enfoque.
<i>LostFocus</i>	Ocurre cuando un objeto pierde el enfoque, ya sea por una acción del usuario, como tabular o hacer clic en otro objeto, o bien mediante un cambio del enfoque en el código con el método <i>SetFocus</i> .
<i>MouseDown</i> y <i>MouseUp</i>	Ocurren cuando el usuario presiona ( <i>MouseDown</i> ) o suelta ( <i>MouseUp</i> ) un botón del mouse.
<i>MouseMove</i>	Ocurre cuando el usuario mueve el mouse.
<i>OLECompleteDrag</i>	Se produce cuando un componente de origen se coloca en un componente de destino e informa al componente de origen de que se ha realizado o cancelado una operación de arrastre.
<i>OLEDragDrop</i>	Se produce cuando un componente de origen se coloca en un componente de destino, si el componente de origen determina que la colocación puede realizarse. Nota: Este evento sólo se produce cuando <i>OLEDropMode</i> está establecido a 1 (Manual).
<i>OLEDragOver</i>	Se produce cuando un componente se arrastra sobre otro.
<i>OLEGiveFeedback</i>	Se produce después de cada evento <i>OLEDragOver</i> . <i>OLEGiveFeedback</i> permite al componente de origen proporcionar al usuario una indicación visual, como cambiar el cursor del mouse para indicar lo que ocurrirá si coloca el objeto o señalar visualmente la selección (en el componente de origen) para indicar lo que va a ocurrir.
<i>OLESetData</i>	Se produce en un componente de origen cuando un componente de destino realiza el método <i>GetData</i> sobre el objeto <i>DataObject</i> del origen, pero antes de que se carguen los datos del formato especificado.
<i>OLEStartDrag</i>	Se produce cuando se ejecuta el método <i>OLEDrag</i> de un componente o cuando un componente inicia una operación OLE de arrastrar y colocar, y la propiedad <i>OLEDragMode</i> está establecida a <i>Automatic</i> . Este evento especifica los formatos de los datos y los efectos de colocación que admite el componente de origen. También puede usarlo para insertar datos en el objeto <i>DataObject</i> .
<i>Scroll</i>	Ocurre cuando el cuadro de desplazamiento de un control <i>ScrollBar</i> , o un objeto que contenga una barra de desplazamiento, cambia de posición o se desplaza horizontal o verticalmente.
<i>Validate</i>	Ocurre cuando el foco cambia a un (segundo) control que tiene su propiedad <i>CausesValidation</i> establecida a <i>True</i> .

<b>Métodos</b>	<b>Descripción</b>
<i>AddItem</i>	Agrega un elemento a un control <i>ListBox</i> .
<i>Clear</i>	Borra el contenido del control <i>ListBox</i> .
<i>Drag</i>	Inicia, termina o cancela una operación de arrastre del control. No acepta argumentos con nombre.
<i>Move</i>	Mueve el control. No acepta argumentos con nombre.

Métodos	Descripción
<i>OLEDrag</i>	Hace que un componente inicie una operación OLE de arrastrar y colocar.
<i>Refresh</i>	Fuerza el volver a dibujar un formulario o un control completo.
<i>RemoveItem</i>	Quita un elemento de un control ListBox.
<i>SetFocus</i>	Mueve el enfoque al control o formulario especificado.
<i>ShowWhatsThis</i>	Muestra un tema seleccionado de un archivo de Ayuda utilizando el menú emergente ¿Qué es esto? que ofrece la Ayuda de Windows 95.
<i>ZOrder</i>	Coloca al control al principio o al final del orden z dentro de su nivel gráfico. No acepta argumentos con nombre.

Propiedades	Descripción
<i>Appearance</i>	Devuelve o establece el estilo de dibujo de los controles de un objeto MDIForm o Form en tiempo de diseño. Es de sólo lectura en tiempo de ejecución.
<i>BackColor</i> y <i>ForeColor</i>	<i>BackColor</i> : devuelve o establece el color de fondo de un objeto. <i>ForeColor</i> : devuelve o establece el color de primer plano utilizado para mostrar texto y gráficos en un objeto.
<i>Columns</i>	Devuelve o establece un valor que determina si un control ListBox se desplaza horizontal o verticalmente y cómo se presentan los elementos en las columnas. Si se desplaza horizontalmente, la propiedad <i>Columns</i> determina cuántas columnas se presentan.
<i>Container</i>	Devuelve o establece el contenedor de un control de un Form. No está disponible en tiempo de diseño.
<i>DataChanged</i>	Devuelve o establece un valor que indica que algún proceso distinto de la recuperación de datos del registro actual ha cambiado los datos del control enlazado. No está disponible en tiempo de diseño.
<i>DataField</i>	Devuelve o establece el nombre de un campo al que será enlazado un receptor de datos.
<i>DataFormat</i>	Devuelve o establece el objeto <i>StdDataFormat</i> al que está adjunto un objeto vinculado. Lectura y escritura en tiempo de diseño y en tiempo de ejecución.
<i>DataMember</i>	Devuelve o establece un miembro de datos especificado entre los varios que ofrece el proveedor de datos.
<i>DataSource</i>	Devuelve o establece el origen de datos mediante el cual un receptor de datos enlaza con una base de datos.
<i>DragIcon</i>	Devuelve o establece el icono que se presenta como puntero en una operación de arrastrar y colocar.
<i>DragMode</i>	Devuelve o establece un valor que determina si se usa el modo de arrastre manual o automático en una operación de arrastrar y colocar.
<i>Enabled</i>	Devuelve o establece un valor que determina si un formulario o un control puede responder a eventos generados por el usuario.
<i>Font</i>	Devuelve un objeto <i>Font</i> .
<i>FontBold</i> , <i>FontItalic</i> , <i>FontStrikethru</i> y <i>FontUnderline</i>	Devuelve o establece los estilos de fuente en los siguientes formatos: Negrita, Cursiva, <del>Tachado</del> y <u>Subrayado</u> .

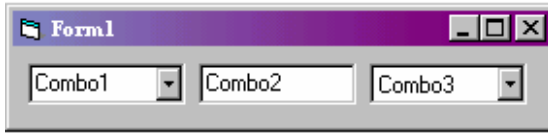


<b>Propiedades</b>	<b>Descripción</b>
<i>FontName</i>	Devuelve o establece la fuente utilizada para mostrar texto en el control.
<i>FontSize</i>	Devuelve o establece el tamaño de la fuente que se va a usar para el texto mostrado en el control.
<i>Height y Width</i>	Devuelven o establecen las dimensiones del control.
<i>HelpContextID</i>	Devuelve o establece un número de contexto asociado a un objeto. Se utiliza para proporcionar Ayuda interactiva para la aplicación.
<i>HWND</i>	Devuelve el controlador de un formulario o un control. Nota: Esta propiedad no es compatible con el control contenedor OLE.
<i>Index(Control Array)</i>	Devuelve o establece el número que identifica un control de forma exclusiva en una matriz de controles. Sólo está disponible si el control forma parte de una matriz de controles.
<i>ItemData</i>	Devuelve o establece un número específico para cada elemento de un control ComboBox o ListBox.
<i>Left y Top</i>	Left: devuelve o establece la distancia entre el borde interno izquierdo de un objeto y el borde izquierdo de su contenedor. Top: devuelve o establece la distancia entre el borde interno superior de un objeto y el borde superior de su contenedor.
<i>List</i>	Devuelve o establece los elementos contenidos en la parte de lista de un control. La lista es una matriz de cadenas cuyos elementos se corresponden con los de la lista.
<i>ListCount</i>	Devuelve el número de elementos de la parte de lista de un control.
<i>ListIndex</i>	Devuelve o establece el índice del elemento seleccionado actualmente en el control. No está disponible en tiempo de diseño.
<i>MouseIcon</i>	Devuelve o establece un icono personalizado de mouse (ratón).
<i>MousePointer</i>	Devuelve o establece un valor que indica el tipo de puntero de mouse que aparece cuando se sitúa el mouse sobre un área determinada de un objeto en tiempo de ejecución.
<i>MultiSelect</i>	Devuelve o establece un valor que indica si el usuario puede realizar selecciones múltiples en un control FileListBox o ListBox, y la forma de llevarlas a cabo. Es de sólo lectura en tiempo de ejecución.
<i>Name</i>	Devuelve el nombre utilizado en el código para identificar un formulario, un control o un objeto de acceso a datos. Es de sólo lectura en tiempo de ejecución.
<i>NewIndex</i>	Devuelve el índice del último elemento agregado a un control ComboBox o ListBox. Es de sólo lectura en tiempo de ejecución.
<i>OLEDragMode</i>	Devuelve o establece un valor que indica si las operaciones OLE de arrastrar y colocar las controla el componente o el programador.
<i>OLEDropMode</i>	Devuelve o establece el modo en que un componente de destino realiza las operaciones de colocación.
<i>Parent</i>	Devuelve el formulario, objeto o colección que contiene al control.
<i>RightToLeft</i>	Devuelve un valor booleano que indica el sentido de presentación del texto y controla la apariencia visual en un sistema bidireccional.
<i>SelCount</i>	Devuelve el número de elementos seleccionados en un control

<b>Propiedades</b>	<b>Descripción</b>
	ListBox.
<i>Selected</i>	Devuelve o establece el estado de selección de un elemento de un control FileListBox o ListBox. Esta propiedad es una matriz de valores booleanos con el mismo número de elementos que la propiedad List. No está disponible en tiempo de diseño.
<i>Sorted</i>	Devuelve un valor que indica si los elementos de un control se colocan automáticamente en orden alfabético.
<i>Style</i>	Devuelve o establece un valor que indica el tipo de muestra y el comportamiento del control. Es de sólo lectura en tiempo de ejecución.
<i>TabIndex</i>	Devuelve o establece el orden de tabulación de la mayoría de los objetos dentro de su formulario primario.
<i>TabStop</i>	Devuelve o establece un valor que indica si el usuario puede usar la tecla TAB para llevar el enfoque a un objeto.
<i>Tag</i>	Devuelve o establece una expresión que almacena cualquier dato adicional que necesite el programa. Al contrario de lo que sucede con otras propiedades, Visual Basic no utiliza el valor de Tag. Puede usar esta propiedad para identificar objetos.
<i>Text</i>	devuelve el elemento seleccionado en el cuadro de lista; el valor devuelto es siempre equivalente al que devuelve la expresión List(ListIndex).
<i>ToolTipText</i>	Devuelve o establece Información sobre herramientas.
<i>TopIndex</i>	Devuelve o establece un valor que especifica el elemento de un control ComboBox, DirListBox, DriveListBox, FileListBox o ListBox que se muestra en el extremo superior del control. No está disponible en tiempo de diseño.
<i>Visible</i>	Devuelve o establece un valor que indica si un objeto es visible o está oculto.
<i>WhatsThisHelpID</i>	Devuelve o establece un número de contexto asociado a un objeto. Se utiliza para dotar a las aplicaciones de Ayuda interactiva con el menú emergente ¿Qué es esto? de la Ayuda de Windows 95.

### Listas desplegables (ComboBox)

La diferencia entre una *lista fija* y una *lista desplegable* (también denominada *cuadro combinado*) es que la *lista desplegable* es una combinación de una lista y una caja de texto. Hay tres estilos diferentes de listas desplegables, los cuales se muestran en la figura siguiente como *Combo1*, *Combo2* y *Combo3*.



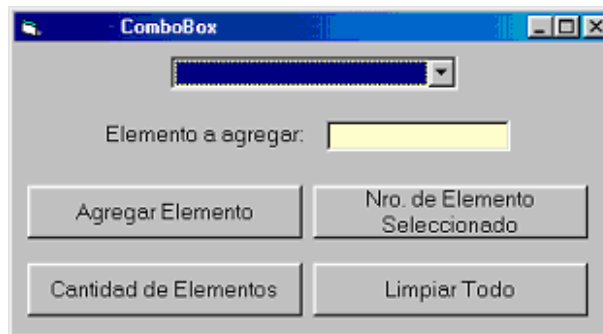
El control *Combo1* representa una lista desplegable estándar y se obtiene poniendo la propiedad *Style* de dicho control a valor 0 (*DropDown Combo*). Cuando el usuario haga clic en la flecha, se

visualizará la lista de elementos y podrá optar por elegir un elemento de la lista o por escribir directamente el elemento deseado en la caja de texto.

El control *Combo2* representa una lista desplegable en la que la lista de elementos siempre está visualizada y se obtiene poniendo la propiedad *Style* de dicho control a valor 1 (*Simple Combo*). El usuario podrá optar por elegir un elemento de la lista o por escribir directamente el elemento deseado en la caja de texto.

El control *Combo3* representa una lista desplegable en la que, a diferencia de las anteriores, el usuario sólo tiene la posibilidad de elegir un elemento de la lista; esto es, no se le permite escribir en la caja de texto. Se obtiene poniendo la propiedad *Style* de dicho control a valor 2 (*DropDown List*). En otras palabras, *Combo3* más que un combinado es una lista enrollable.

Para ver cómo trabaja este control crearemos una nueva aplicación, en la cual el formulario tendrá un diseño similar al siguiente. El objetivo de esta aplicación es el de agregar un elemento al combo desde el texto introducido en la caja de texto, mostrar cuál es el número del ítem seleccionado, mostrar la cantidad de elementos del combo y blanquearlo.



La codificación correspondiente al botón de pulsación *Agregar Elemento* es la siguiente:

```
Private Sub cmdAgregar_Click()
    Combo1.AddItem Text1.Text
End Sub
```

El método *AddItem* agrega el texto introducido en la caja de texto como un elemento del combo *Combo1*.

Para mostrar el número de elemento seleccionado en el combo se utiliza la propiedad *ListIndex*.

```
Private Sub cmdNroElemento_Click()
    MsgBox Combo1.ListIndex
End Sub
```

Para mostrar el número de elementos de un combo se utiliza la propiedad *ListCount*.

```
Private Sub cmdCantidad_Click()
    MsgBox Combo1.ListCount
End Sub
```

Y por último, para limpiar el combo, o sea para eliminar todos sus elementos se utiliza el método *Clear*.

```
Private Sub cmdLimpiar_Click()
    Combo1.Clear
End Sub
```

### Eventos, métodos y propiedades del control ComboBox

Eventos	Descripción
<i>Change</i>	Indica que el contenido de un control ha cambiado. Cómo y cuándo ha ocurrido este evento varía según el control, en este caso, cambia el texto de la parte de cuadro de texto del control. Ocurre sólo si la propiedad Style está establecida a 0 (Dropdown Combo) o 1 (Simple Combo) y el usuario cambia el texto o usted cambia la configuración de la propiedad Text mediante código.
<i>Clic</i>	Ocurre cuando el usuario presiona y suelta un botón del mouse (ratón) en un objeto.
<i>DbClick</i>	Ocurre cuando hace doble clic en un elemento de un control ComboBox cuya propiedad Style es 1 (Simple)
<i>DragDrop</i>	Ocurre cuando se completa una operación de arrastrar y colocar como resultado de arrastrar un control sobre un objeto y soltar el botón del mouse o usar el método Drag con su argumento acción establecido a 2 (Drop).
<i>DragOver</i>	Ocurre cuando una operación de arrastrar y colocar está en curso. Puede usar este evento para controlar el puntero del mouse a medida que entra, sale o descansa directamente sobre un destino válido. La posición del puntero del mouse determina el objeto de destino que recibe este evento.
<i>GotFocus</i>	Ocurre cuando un objeto recibe el enfoque, ya sea mediante una acción del usuario, como tabular o hacer clic en el objeto, o cambiando el enfoque en el código mediante el método SetFocus. Un formulario recibe el enfoque sólo cuando todos los controles visibles están desactivados.
<i>KeyDown y KeyUp</i>	Ocurren cuando el usuario presiona (KeyDown) o suelta (KeyUp) una tecla mientras un objeto tiene el enfoque.
<i>KeyPress</i>	Ocurre cuando el usuario presiona y suelta una tecla ANSI.
<i>LostFocus</i>	Ocurre cuando un objeto pierde el enfoque, ya sea por una acción del usuario, como tabular o hacer clic en otro objeto, o bien mediante un cambio del enfoque en el código con el método SetFocus.
<i>OLECompleteDrag</i>	Se produce cuando un componente de origen se coloca en un componente de destino e informa al componente de origen de que se ha realizado o cancelado una operación de arrastre.
<i>OLEDragDrop</i>	Se produce cuando un componente de origen se coloca en un componente de destino, si el componente de origen determina que la colocación puede realizarse. Nota: Este evento sólo se produce cuando OLEDropMode está establecido a 1 (Manual).
<i>OLEDragOver</i>	Se produce cuando un componente se arrastra sobre otro.
<i>OLEGiveFeedback</i>	Se produce después de cada evento OLEDragOver. OLEGiveFeedback permite al componente de origen proporcionar al usuario una indicación visual, como cambiar el cursor del mouse para indicar lo que ocurrirá si coloca el objeto o señalar visualmente la selección (en el componente de origen) para indicar lo que va a ocurrir.
<i>OLESetData</i>	Se produce en un componente de origen cuando un componente de

Eventos	Descripción
	destino realiza el método GetData sobre el objeto DataObject del origen, pero antes de que se carguen los datos del formato e especificado.
<i>OLEStartDrag</i>	Se produce cuando se ejecuta el método OLEDrag de un componente o cuando un componente inicia una operación OLE de arrastrar y colocar, y la propiedad OLEDragMode está establecida a Automatic. Este evento especifica los formatos de los datos y los efectos de colocación que admite el componente de origen. También puede usarlo para insertar datos en el objeto DataObject.
<i>Scroll</i>	Ocurre cuando el cuadro de desplazamiento de un control ScrollBar, o un objeto que contenga una barra de desplazamiento, cambia de posición o se desplaza horizontal o verticalmente.
<i>Validate</i>	Ocurre cuando el foco cambia a un (segundo) control que tiene su propiedad CausesValidation establecida a True.

Métodos	Descripción
<i>AddItem</i>	Agrega un elemento a un control ListBox.
<i>Clear</i>	Borra el contenido del control ListBox.
<i>Drag</i>	Inicia, termina o cancela una operación de arrastre del control. No acepta argumentos con nombre.
<i>Move</i>	Mueve el control. No acepta argumentos con nombre.
<i>OLEDrag</i>	Hace que un componente inicie una operación OLE de arrastrar y colocar.
<i>Refresh</i>	Fuerza el volver a dibujar un formulario o un control completo.
<i>RemoveItem</i>	Quita un elemento de un control ListBox.
<i>SetFocus</i>	Mueve el enfoque al control o formulario especificado.
<i>ShowWhatsThis</i>	Muestra un tema seleccionado de un archivo de Ayuda utilizando el menú emergente ¿Qué es esto? que ofrece la Ayuda de Windows 95.
<i>ZOrder</i>	Coloca al control al principio o al final del orden z dentro de su nivel gráfico. No acepta argumentos con nombre.

Propiedades	Descripción
<i>Appearance</i>	Devuelve o establece el estilo de dibujo de los controles de un objeto MDIForm o Form en tiempo de diseño. Es de sólo lectura en tiempo de ejecución.
<i>BackColor</i> y <i>ForeColor</i>	<i>BackColor</i> : devuelve o establece el color de fondo de un objeto. <i>ForeColor</i> : devuelve o establece el color de primer plano utilizado para mostrar texto y gráficos en un objeto.
<i>Container</i>	Devuelve o establece el contenedor de un control de un Form. No está disponible en tiempo de diseño.
<i>DataChanged</i>	Devuelve o establece un valor que indica que algún proceso distinto de la recuperación de datos del registro actual ha cambiado los datos del control enlazado. No está disponible en tiempo de diseño.
<i>DataField</i>	Devuelve o establece el nombre de un campo al que será enlazado un receptor de datos.

<b>Propiedades</b>	<b>Descripción</b>
<i>DataFormat</i>	Devuelve o establece el objeto StdDataFormat al que está adjunto un objeto vinculado. Lectura y escritura en tiempo de diseño y en tiempo de ejecución.
<i>DataMember</i>	Devuelve o establece un miembro de datos especificado entre los varios que ofrece el proveedor de datos.
<i>DragIcon</i>	Devuelve o establece el icono que se presenta como puntero en una operación de arrastrar y colocar.
<i>DragMode</i>	Devuelve o establece un valor que determina si se usa el modo de arrastre manual o automático en una operación de arrastrar y colocar.
<i>Enabled</i>	Devuelve o establece un valor que determina si un formulario o un control puede responder a eventos generados por el usuario.
<i>Font</i>	Devuelve un objeto Font.
<i>FontBold</i> , <i>FontItalic</i> , <i>FontStrikethru</i> y <i>FontUnderline</i>	Devuelve o establece los estilos de fuente en los siguientes formatos: Negrita, Cursiva, <del>Tachado</del> y <u>Subrayado</u> .
<i>FontName</i>	Devuelve o establece la fuente utilizada para mostrar texto en el control.
<i>FontSize</i>	Devuelve o establece el tamaño de la fuente que se va a usar para el texto mostrado en el control.
<i>Height</i> y <i>Width</i>	Devuelven o establecen las dimensiones del control.
<i>HelpContextID</i>	Devuelve o establece un número de contexto asociado a un objeto. Se utiliza para proporcionar Ayuda interactiva para la aplicación.
<i>HWnd</i>	Devuelve el controlador de un formulario o un control. Nota: Esta propiedad no es compatible con el control contenedor OLE.
<i>Index(Control Array)</i>	Devuelve o establece el número que identifica un control de forma exclusiva en una matriz de controles. Sólo está disponible si el control forma parte de una matriz de controles.
<i>IntegralHeight</i>	Devuelve o establece un valor que indica si el control muestra elementos parciales. Es de sólo lectura en tiempo de ejecución.
<i>ItemData</i>	Devuelve o establece un número específico para cada elemento de un control ComboBox o ListBox.
<i>Left</i> y <i>Top</i>	Left: devuelve o establece la distancia entre el borde interno izquierdo de un objeto y el borde izquierdo de su contenedor. Top: devuelve o establece la distancia entre el borde interno superior de un objeto y el borde superior de su contenedor.
<i>List</i>	Devuelve o establece los elementos contenidos en la parte de lista de un control. La lista es una matriz de cadenas cuyos elementos se corresponden con los de la lista.
<i>ListCount</i>	Devuelve el número de elementos de la parte de lista de un control.
<i>ListIndex</i>	Devuelve o establece el índice del elemento seleccionado actualmente en el control. No está disponible en tiempo de diseño.
<i>Locked</i>	Devuelve o establece un valor que indica si un control se puede modificar.

<b>Propiedades</b>	<b>Descripción</b>
<i>MouseIcon</i>	Devuelve o establece un icono personalizado de mouse (ratón).
<i>MousePointer</i>	Devuelve o establece un valor que indica el tipo de puntero de mouse que aparece cuando se sitúa el mouse sobre un área determinada de un objeto en tiempo de ejecución.
<i>Name</i>	Devuelve el nombre utilizado en el código para identificar un formulario, un control o un objeto de acceso a datos. Es de sólo lectura en tiempo de ejecución.
<i>NewIndex</i>	Devuelve el índice del último elemento agregado a un control ComboBox o ListBox. Es de sólo lectura en tiempo de ejecución.
<i>OLEDragMode</i>	Devuelve o establece un valor que indica si las operaciones OLE de arrastrar y colocar las controla el componente o el programador.
<i>OLEDropMode</i>	Devuelve o establece el modo en que un componente de destino realiza las operaciones de colocación.
<i>Parent</i>	Devuelve el formulario, objeto o colección que contiene al control.
<i>RightToLeft</i>	Devuelve un valor booleano que indica el sentido de presentación del texto y controla la apariencia visual en un sistema bidireccional.
<i>SelLenght</i> , <i>SelStart</i> , <i>SelText</i>	<i>SelLength</i> : devuelve o establece el número de caracteres seleccionados. <i>SelStart</i> : devuelve o establece el punto inicial del texto seleccionado, indicando la posición del punto de inserción si no se ha seleccionado texto. <i>SelText</i> : devuelve o establece una cadena con el texto seleccionado actualmente o es una cadena de longitud cero ("" ) si no hay caracteres seleccionados.
<i>Sorted</i>	Devuelve un valor que indica si los elementos de un control se colocan automáticamente en orden alfabético.
<i>Style</i>	Devuelve o establece un valor que indica el tipo de muestra y el comportamiento del control. Es de sólo lectura en tiempo de ejecución.
<i>TabIndex</i>	Devuelve o establece el orden de tabulación de la mayoría de los objetos dentro de su formulario primario.
<i>TabStop</i>	Devuelve o establece un valor que indica si el usuario puede usar la tecla TAB para llevar el enfoque a un objeto.
<i>Tag</i>	Devuelve o establece una expresión que almacena cualquier dato adicional que necesite el programa. Al contrario de lo que sucede con otras propiedades, Visual Basic no utiliza el valor de Tag. Puede usar esta propiedad para identificar objetos.
<i>Text</i>	devuelve el elemento seleccionado en el cuadro de lista; el valor devuelto es siempre equivalente al que devuelve la expresión List(ListIndex).
<i>ToolTipText</i>	Devuelve o establece Información sobre herramientas.
<i>TopIndex</i>	Devuelve o establece un valor que especifica el elemento de un control ComboBox, DirListBox, DriveListBox, FileListBox o ListBox que se muestra en el extremo superior del control. No está disponible en tiempo de diseño.
<i>Visible</i>	Devuelve o establece un valor que indica si un objeto es visible o

<b>Propiedades</b>	<b>Descripción</b>
	está oculto.
<i>WhatsThisHelpID</i>	Devuelve o establece un número de contexto asociado a un objeto. Se utiliza para dotar a las aplicaciones de Ayuda interactiva con el menú emergente ¿Qué es esto? de la Ayuda de Windows 95.



### Timer (Cronómetro)

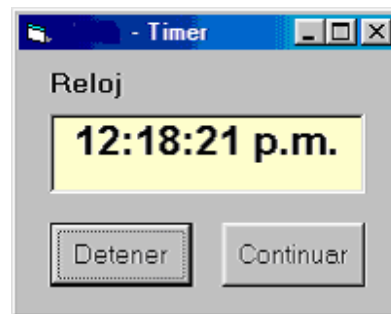
Un temporizador es un control de Visual Basic que responde a intervalos regulares de tiempo. Es útil para realizar procesos en segundo plano. Esto quiere decir que en el procedimiento asociado con el mismo especificaremos las acciones que deseemos que se ejecuten cada vez que transcurra un intervalo de tiempo.

Cada temporizador tiene una propiedad *Interval* que especifica el intervalo de tiempo en milisegundos que tiene que transcurrir para que su procedimiento asociado se ejecute independientemente del usuario. El valor de la propiedad *Interval* puede oscilar entre 0 y 65535 milisegundos (0 a 65,5 segundos). Un valor 0, valor predeterminado, desactiva el temporizador. No obstante, tenga en cuenta que como el sistema genera 18 tics de reloj por segundo, la precisión no puede ser mayor de 1000/18 milésimas de segundo.

Una utilidad típica de este control es verificar de alguna forma al sistema, para ver si es el momento de ejecutar alguna tarea.

Si su aplicación u otra aplicación está realizando una tarea que mantiene ocupados los recursos del ordenador por un espacio largo de tiempo, tal como un bucle largo, cálculos intensivos, acceso a los puertos, etc., puede ser que su aplicación no responda de acuerdo con 1os intervalos de tiempo programados.

El siguiente ejemplo le enseñará, entre otras cosas, a utilizar un temporizador. Para ello diseñaremos un formulario como el siguiente:



El formulario está compuesto por una etiqueta *Label1*, dos botones de pulsación y un control *Timer* con *Interval* igual a 1000. La etiqueta simulará un reloj el cual será detenido al hacer clic sobre el botón de pulsación *Detener* y seguirá corriendo al hacer clic en el botón de pulsación *Continuar*. Para que la etiqueta simule el reloj habrá que asignarle a la propiedad *Caption* de la etiqueta *lblReloj* cada vez que se produzca el evento *Timer* del control *Timer1*.

```
Private Sub Timer1_Timer()
    lblReloj.Caption = Time
End Sub
```

Para detener el contador habrá que deshabilitar el control *Timer1*.

```
Private Sub cmdDetener_Click()
    Timer1.Enabled = False
End Sub
```

Para que el contador siga funcionando habrá que habilitar el control *Timer1*.

```
Private Sub cmdContinuar_Click()
    Timer1.Enabled = True
End Sub
```

Y para poner en cero al contador tendremos que asignarle a la propiedad *caption* de la etiqueta *Label1* el valor cero, en el evento *Form\_Load*.

```
Private Sub Form_Load()
    lblReloj.Caption = "00:00:00"
End Sub
```

Problemos ahora con un ejemplo un tanto más complejo. Simularemos una marquesina desplazándose sobre el formulario. Para ello necesitaremos una etiqueta llamada *Label1* y un control *Timer*. Desplazaremos la etiqueta utilizando su propiedad *Left*, la cual indica la distancia entre el borde interno de la etiqueta y el borde izquierdo del formulario. Cuando la etiqueta llegue al borde izquierdo del formulario, la misma volverá a aparecer en el borde derecho del formulario. El código siguiente corresponde al evento *Timer* del control *Timer1*.

```
Private Sub Timer1_Timer()
    Label1.Left = Label1.Left - 50
    If Label1.Left < -615 Then
        Label1.Left = 4800
    End If
End Sub
```

Ejecute la aplicación y observe los resultados.

Evento	Descripción
<i>Timer</i>	Ocurre cuando ha transcurrido un intervalo preestablecido para un control <i>Timer</i> . La frecuencia del intervalo se almacena en la propiedad <i>Interval</i> del control, que especifica el tiempo en milisegundos.

Propiedades	Descripción
<i>Enabled</i>	Devuelve o establece un valor que determina si un formulario o un control puede responder a eventos generados por el usuario.
<i>Index(Matriz)</i>	Devuelve o establece el número que identifica un control de forma exclusiva en una matriz de controles. Sólo está disponible si el control forma parte de una matriz de controles.
<i>Interval</i>	Devuelve o establece el número de milisegundos entre las llamadas al evento <i>Timer</i> de un control <i>Timer</i> .
<i>Left y Top</i>	<i>Left</i> : devuelve o establece la distancia entre el borde interno izquierdo de un objeto y el borde izquierdo de su contenedor. <i>Top</i> : devuelve o establece la distancia entre el borde interno superior de un objeto y el borde superior de su contenedor.
<i>Name</i>	Devuelve el nombre utilizado en el código para identificar un formulario, un control o un objeto de acceso a datos. Es de sólo lectura en tiempo de ejecución.
<i>Parent</i>	Devuelve el formulario, objeto o colección que contiene al control.
<i>Tag</i>	Devuelve o establece una expresión que almacena cualquier dato adicional que necesite el programa. Al contrario de lo que sucede con otras propiedades, Visual Basic no utiliza el valor de <i>Tag</i> . Puede usar esta propiedad para identificar objetos.

## Image (Imagen)

Utilice el control *Image* para mostrar un gráfico. Un control *Image* puede mostrar un gráfico desde un mapa de bits, un icono o un metarchivo, así como un metarchivo mejorado, un archivo JPEG o archivos GIF.

El control *Image* utiliza menos recursos del sistema y actualiza con más rapidez que un control *PictureBox*, pero sólo admite un subconjunto de las propiedades, los eventos y los métodos de *PictureBox*. Use la propiedad *Stretch* para determinar si el gráfico se escala para ajustarse al control o viceversa. Aunque puede colocar un control *Image* dentro de un contenedor, un control *Image* no puede actuar como contenedor.

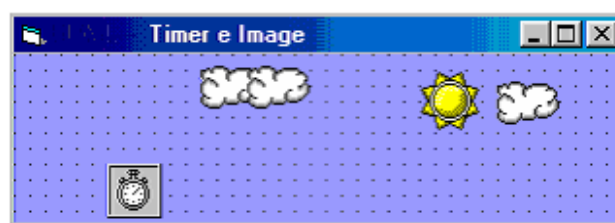
El truco para crear animación en un programa consiste mover una imagen en un procedimiento de evento temporizador de tal forma que, a determinados intervalos, el temporizador provocará el movimiento de uno o más objetos por la pantalla. Cuando trabaje con animación, deberá asignar a la propiedad *Interval* del temporizador una velocidad bastante más rápida: 1/5 segundos (200 milisegundos), 1/10 segundos ( 100 milisegundos ) o incluso menos. La velocidad exacta que deberá elegir dependerá de la rapidez con la que desee que actúe la animación.

Otro truco será emplear las propiedades *Top* y *Left* para detectar el margen superior y el margen izquierdo del formulario. El empleo de estos valores en un procedimiento de evento le permitirá detener la animación (desactivar el temporizador) cuando el objeto alcance el borde del formulario. También podrá emplear la propiedad *Top*, la propiedad *Left*, o ambas, en una estructura de decisión *If...Then* o *Select Case* para simular el rebote de un objeto cuando éste alcance alguno de los bordes del formulario.

El siguiente ejercicio muestra cómo podrá animar un cuadro de dibujo en un programa empleando el método *Left* y un objeto temporizador. En este ejercicio añadirá un sol al programa. El sol se moverá a través del formulario.

1. Abra un nuevo proyecto.
2. Pulse el control *Image* contenido en el cuadro de herramientas y dibuje pequeño rectángulo en la parte superior del formulario. Introducirá un icono sol dentro de este cuadro cuando asigne las propiedades del objeto. Dibuje varios controles *Image* para agregar nubes al formulario.
3. Pulse el control *Timer* del cuadro de herramientas y dibuje un objeto temporizador en la esquina inferior izquierda del formulario. El objeto temporizador (*Timer1*) reajusta su tamaño por sí solo en el formulario.

Después de asignar los iconos a los controles *Image*, su formulario se parecerá al de figura siguiente.



Agregue las siguientes líneas de código en el evento *Timer* del control *Time1r*:

```
Private Sub Timer1_Timer()
    Image1.Left = Image1.Left - 50
    If Image1.Left < -615 Then
        Image1.Left = 4800
    End If
End Sub
```

Las nuevas sentencias ponen en funcionamiento el temporizador para desplazar al sol. Como ya ha ajustado el intervalo del temporizador a 100 milisegundos, el temporizador está listo para ponerse en marcha. Sólo tendrá que añadir el método *Left*.

Tal como podrá comprobar cuando ejecute el programa, estas instrucciones harán que el sol parezca que se desplaza suavemente.

Cierre la ventana Código y pulse el botón Guardar proyecto para almacenar los cambios realizados. Ahora ejecute el programa.



Eventos a los que responde el control *Image*:

Eventos	Descripción
<i>Clic</i>	Ocurre cuando el usuario presiona y suelta un botón del mouse (ratón) en un objeto. También puede ocurrir cuando se cambia el valor de un control.
<i>DbClick</i>	Se produce cuando un usuario presiona y suelta un botón del mouse (ratón) dos veces sobre un objeto.
<i>DragDrop</i>	Ocurre cuando se completa una operación de arrastrar y colocar como resultado de arrastrar un control sobre un objeto y soltar el botón del mouse o usar el método Drag con su argumento acción establecido a 2 (Drop).
<i>DragOver</i>	Ocurre cuando una operación de arrastrar y colocar está en curso. Puede usar este evento para controlar el puntero del mouse a medida que entra, sale o descansa directamente sobre un destino válido. La posición del puntero del mouse determina el objeto de destino que recibe este evento.
<i>MouseDown</i> y <i>MouseUp</i>	Ocurren cuando el usuario presiona ( <i>MouseDown</i> ) o suelta ( <i>MouseUp</i> ) un botón del mouse.
<i>MouseMove</i>	Ocurre cuando el usuario mueve el mouse.
<i>OLECompleteDrag</i>	Se produce cuando un componente de origen se coloca en un componente de destino e informa al componente de origen de que se ha realizado o cancelado una operación de arrastre.
<i>OLEDragDrop</i>	Se produce cuando un componente de origen se coloca en un componente de destino, si el componente de origen determina que la colocación puede realizarse. Este evento sólo se produce cuando <i>OLEDropMode</i> está establecido a 1 (Manual).
<i>OLEDragOver</i>	Se produce cuando un componente se arrastra sobre otro.
<i>OLEGiveFeedback</i>	Se produce después de cada evento <i>OLEDragOver</i> . <i>OLEGiveFeedback</i> permite al componente de origen proporcionar al usuario una indicación visual, como cambiar el cursor del mouse para indicar lo que ocurrirá si coloca el objeto o señalar visualmente la selección (en el componente de origen) para indicar lo que va a ocurrir.
<i>OLESetData</i>	Se produce en un componente de origen cuando un componente de destino realiza el método <i>GetData</i> sobre el objeto <i>DataObject</i> del origen, pero antes de que se carguen los datos del formato especificado.

Eventos	Descripción
<i>OLEStartDrag</i>	Se produce cuando se ejecuta el método <i>OLEDrag</i> de un componente o cuando un componente inicia una operación OLE de arrastrar y colocar, y la propiedad <i>OLEDragMode</i> está establecida a <i>Automatic</i> . Este evento especifica los formatos de los datos y los efectos de colocación que admite el componente de origen. También puede usarlo para insertar datos en el objeto <i>DataObject</i> .

Métodos correspondientes al control *Image*:

Métodos	Descripción
<i>Drag</i>	Inicia, termina o cancela una operación de arrastre de cualquier control excepto los controles <i>Line</i> , <i>Menu</i> , <i>Shape</i> , <i>Timer</i> o <i>CommonDialog</i> . No acepta argumentos con nombre.
<i>Move</i>	Mueve objetos <i>MDIForm</i> o <i>Form</i> , o controles. No acepta argumentos con nombre.
<i>OLEDrag</i>	Hace que un componente inicie una operación OLE de arrastrar y colocar.
<i>Refresh</i>	Fuerza el volver a dibujar un formulario o un control completo.
<i>ShowWhatsThis</i>	Muestra un tema seleccionado de un archivo de Ayuda utilizando el menú emergente ¿Qué es esto? que ofrece la Ayuda de Windows 95.
<i>ZOrder</i>	Coloca un objeto <i>MDIForm</i> o <i>Form</i> , o un control al principio o al final del orden z dentro de su nivel gráfico. No acepta argumentos con nombre.

Propiedades correspondientes al control *Image*:

Propiedades	Descripción
<i>Appearance</i>	Devuelve o establece el estilo de dibujo de los controles de un objeto <i>MDIForm</i> o <i>Form</i> en tiempo de diseño. Es de sólo lectura en tiempo de ejecución.
<i>BorderStyle</i>	Devuelve o establece el estilo del borde de un objeto.
<i>Container</i>	Devuelve o establece el contenedor de un control de un <i>Form</i> . No está disponible en tiempo de diseño.
<i>DataChanged</i>	Devuelve o establece un valor que indica que algún proceso distinto de la recuperación de datos del registro actual ha cambiado los datos del control enlazado. No está disponible en tiempo de diseño.
<i>DataField</i>	Devuelve o establece el nombre de un campo al que será enlazado un receptor de datos.
<i>DataFormat</i>	Devuelve o establece el objeto <i>StdDataFormat</i> al que está adjunto un objeto vinculado. Lectura y escritura en tiempo de diseño y en tiempo de ejecución.
<i>DataMember</i>	Devuelve o establece un miembro de datos especificado entre los varios que ofrece el proveedor de datos.
<i>DataSource</i>	Devuelve o establece el origen de datos mediante el cual un receptor de datos enlaza con una base de datos.
<i>DragIcon</i>	Devuelve o establece el icono que se presenta como puntero en una operación de arrastrar y colocar.

<b>Propiedades</b>	<b>Descripción</b>
<i>DragMode</i>	Devuelve o establece un valor que determina si se usa el modo de arrastre manual o automático en una operación de arrastrar y colocar.
<i>Enabled</i>	Devuelve o establece un valor que determina si un formulario o un control puede responder a eventos generados por el usuario.
<i>Height y Width</i>	Devuelven o establecen las dimensiones de un objeto.
<i>Index (Control Array)</i>	Devuelve o establece el número que identifica un control de forma exclusiva en una matriz de controles. Sólo está disponible si el control forma parte de una matriz de controles.
<i>MouseIcon</i>	Devuelve o establece el icono que se presenta como puntero en una operación de arrastrar y colocar.
<i>Left y Top</i>	Left: devuelve o establece la distancia entre el borde interno izquierdo de un objeto y el borde izquierdo de su contenedor. Top: devuelve o establece la distancia entre el borde interno superior de un objeto y el borde superior de su contenedor.
<i>MousePointer</i>	Devuelve o establece un valor que indica el tipo de puntero de mouse que aparece cuando se sitúa el mouse sobre un área determinada de un objeto en tiempo de ejecución.
<i>Name</i>	Devuelve el nombre utilizado en el código para identificar un formulario, un control o un objeto de acceso a datos. Es de sólo lectura en tiempo de ejecución.
<i>OLEDragMode</i>	Hace que un componente inicie una operación OLE de arrastrar y colocar.
<i>OLEDropMode</i>	Devuelve o establece el modo en que un componente de destino realiza las operaciones de colocación.
<i>Parent</i>	Devuelve el formulario, objeto o colección que contiene un control u otro objeto o colección.
<i>Picture</i>	Devuelve o establece un gráfico que se mostrará en un control. Para el control contenedor OLE, no está disponible en tiempo de diseño y es de sólo lectura en tiempo de ejecución.
<i>Stretch</i>	Devuelve o establece un valor que indica si un gráfico cambia su tamaño para ajustarse al de un control Image.
<i>Tag</i>	Devuelve o establece una expresión que almacena cualquier dato adicional que necesite el programa. Al contrario de lo que sucede con otras propiedades, Visual Basic no utiliza el valor de Tag. Puede usar esta propiedad para identificar objetos.
<i>ToolTipText</i>	Devuelve o establece Información sobre herramientas.
<i>Visible</i>	Devuelve o establece un valor que indica si un objeto es visible o está oculto.
<i>WhatsThisHelpID</i>	Devuelve o establece un número de contexto asociado a un objeto. Se utiliza para dotar a las aplicaciones de Ayuda interactiva con el menú emergente ¿Qué es esto? de la Ayuda de Windows 95.

## Inclusión de efectos artísticos utilizando los controles line y shape

Ya sabe cómo añadir mapas de bits, iconos y metaarchivos de Windows en un formulario creando cuadros de dibujo y de imagen. Incluir dibujos artísticos predesarrollados en sus programas le resultará una tarea muy fácil en Visual Basic. Ahora aprenderá a crear dibujos artísticos originales utilizando los controles *Line* y *Shape*. Estas herramientas de gran utilidad están localizadas en el cuadro de herramientas y podrá usarlas para construir una gran variedad de imágenes de diferentes formas, tamaños y colores. Los objetos que cree utilizando estos controles tendrán muy pocas limitaciones (por ejemplo, no pueden recibir el foco en tiempo de ejecución ni pueden aparecer encima de otros objetos) a la vez que son muy potentes, rápidos y fáciles de usar.

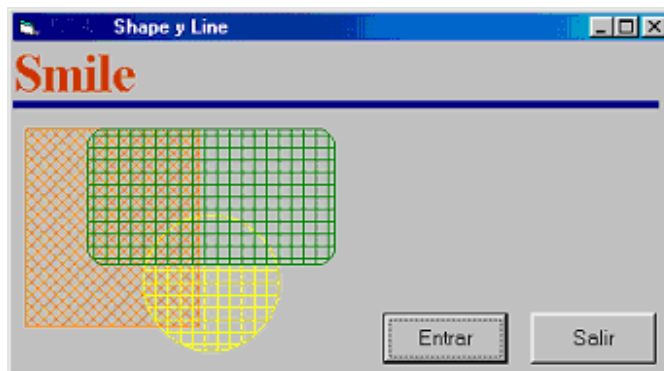
### El control Line

Podrá utilizar el control *Line* para introducir líneas rectas en un formulario. Después podrá disponer de una amplia variedad de propiedades para cambiar el aspecto de la línea que acaba de crear. Las propiedades más importantes del objeto línea son: el ancho del borde (*BorderWidth*), el estilo del borde (*BorderStyle*), el color del borde (*BorderColor*) y la visibilidad (*Visible*). La propiedad *BorderWidth* ajusta el grosor de la línea contenida en su formulario. Esto es especialmente útil cuando esté creando un subrayado o una línea que separa un objeto de otro. La propiedad *BorderStyle* le permitirá crear una línea continua, punteada o a trazos y la propiedad *BorderColor* le permitirá colorearla con cualquiera de los colores estándar de Visual Basic. Finalmente, la propiedad *Visible* le permitirá ocultar o mostrar la línea según sea necesario en su programa. Tendrá la oportunidad de trabajar con el control *Line* un poco más adelante, después de que conozca algo del control *Shape*.

### El control Shape

Podrá utilizar el control *Shape* para crear rectángulos, cuadrados, elipses y circunferencias en sus formularios. Empleará este control para realizar el dibujo deseado y luego utilizará la *ventana Propiedades* para ajustar las características de la imagen. La propiedad *Shape* controla la forma de la imagen; podrá seleccionar un rectángulo, un rectángulo con bordes redondeados, un cuadrado, un cuadrado redondeado, una elipse o una forma circular después de haber creado el objeto. Podrá construir imágenes complejas dibujando varias formas y líneas. Entre otras propiedades importantes que afectan a la forma de este tipo de objetos son: el color de relleno (*FillColor*), que le permitirá especificar el color del objeto; el estilo de relleno (*FillStyle*), que le permitirá establecer un patrón para el color de relleno, y el color del borde (*BoderColor*), con el que podrá especificar un color distinto para el borde de la forma. Este tipo de objetos también cuenta con la propiedad *Visible*, que le permitirá mostrar u ocultar el dibujo según sean sus necesidades.

Los siguiente ejercicios le mostrarán el empleo de los controles *Line* y *Shape*. Utilizará estos controles para crear una atractiva pantalla de bienvenida para un sistema odontológico denominado *Smile*. La pantalla de bienvenida será similar a la mostrada a continuación:



### Empleo de los controles Line y Shape

1. Abra un nuevo proyecto estándar.
2. Oculte el formulario para disponer del suficiente espacio para incluir objetos de forma y botones de órdenes.
3. Pulse el control *Label* del cuadro de herramientas y cree una etiqueta alargada en la parte superior del formulario.
4. Abra la ventana Propiedades y asigne el valor *Smile* a la propiedad *Caption* de la etiqueta.
5. Asigne a la propiedad *Font* de la etiqueta el valor *Times New Roman*, Negrita, 26 puntos. Asigne a la propiedad *ForeColor* el valor Naranja oscuro. La etiqueta deberá ocupar casi todo el ancho de la pantalla. Deberá ajustar el ancho y el alto del cuadro etiqueta, si fuese necesario, de manera que el nombre del sistema quepa en una única línea.
6. Pulse el control *Line* contenido en el cuadro de herramientas y cree una línea bajo el nombre del sistema. Haga que la línea ocupe toda la anchura del formulario. El control *Line* muestra manejadores de selección a cada lado de la línea una vez creada. Podrá utilizar estos manejadores para aumentar o disminuir el tamaño de la línea, si fuese necesario.
7. Pulse el botón Ventana de Propiedades para mostrar la ventana Propiedades y después asigne los siguientes valores a las propiedades del objeto *Line*:

Objeto	Propiedad	Valor
<i>Line1</i>	<i>BorderWidth</i>	5
	<i>BorderColor</i>	Azul oscuro

El nombre del objeto de línea es *Line1*. El valor de *BorderWidth* cambia el ancho de la línea a cinco twips (un twip es equivalente a 1/20 puntos o a 1/1.440 pulgadas). La propiedad *BorderColor* modificará el color de la línea a Azul oscuro.

A continuación, introducirá dos cuadros de imágenes en el formulario.

8. Pulse el control *Shape* contenido en el cuadro de herramientas y cree un rectángulo en la parte izquierda del formulario. Este rectángulo es el marco de la primera ventana. Más adelante ajustará la propiedad *FillStyle* de este objeto para conseguir que se asemeje a una ventana enrejada.
9. Pulse otra vez el control *Shape* y cree un segundo rectángulo en la parte izquierda del formulario, solapándose ligeramente con el primero. El funcionamiento del control *Shape* es muy similar a las herramientas de dibujo de propósito general presentes en los programas de dibujo artístico. El control crea la imagen y luego deberá asignar las propiedades para configurar el dibujo con su aspecto definitivo.
10. Pulse el botón Ventana Propiedades y asigne los siguientes valores para las propiedades de los objetos que acaba de crear:

Objeto	Propiedad	Valor
<i>Shape1</i>	<i>Shape</i>	0-Rectangle
	<i>FillColor</i>	Naranja
	<i>FillStyle</i>	7-Diagonal Cross
	<i>BorderColor</i>	Naranja
<i>Shape2</i>	<i>Shape</i>	3-Circle
	<i>FillColor</i>	Amarillo
	<i>FillStyle</i>	6-Cross
	<i>BorderColor</i>	Amarillo
<i>Shape3</i>	<i>Shape</i>	4-Rounded rectangle
	<i>FillColor</i>	Verde
	<i>FillStyle</i>	6-Cross
	<i>BorderColor</i>	Verde



11. Cree una etiqueta en la parte derecha del formulario lo suficientemente estrecha como para que el rótulo de la etiqueta se extienda por dos líneas completas. A continuación, asigne las propiedades mostradas en la tabla siguiente.

Objeto	Propiedad	Valor
Label2	Caption	Sistema Integral Odontológico
	Font	Times New Roman, Cursiva, 12-puntos
	Alignment	1 - Righth justify

12. Pulse el botón *CommandButton* del cuadro de herramientas y cree un botón de orden en la esquina inferior derecha del formulario. Cree un segundo botón de orden a la izquierda del primero.
13. Asigne los siguientes valores a las propiedades de los botones de orden.

Objeto	Propiedad	Valor
Command1	Caption	Salir
Command2	Caption	Entrar

14. Pulse dos veces el botón de orden *Salir*, escriba *End* en el procedimiento de suceso *Command1\_Click* y luego cierre la ventana Código.

El formulario de bienvenida que está creando pretende ser una puerta de entrada al programa, pero si el usuario quisiera abandonarlo sin recorrer el resto del programa, el botón *Salir* le proporcionará una vía de salida. Puesto que la pantalla de bienvenida es, por ahora, la única parte del programa que existe, también utilizará el botón *Salir* para finalizar la ejecución del programa.