

LECCIÓN 3

Esta lección la vamos a dedicar a colocar en nuestro proyecto nuevos objetos no vistos hasta el momento. Una vez terminada toda la presentación pasaremos a depurar nuestra aplicación para que no se produzcan errores inesperados e introducir mensajes que aparecerán en determinados momentos para que el usuario sepa que es lo que debe hacer. En esta lección también introduciremos líneas de código que iremos comentando, aunque las principales instrucciones de **Visual Basic** las veremos en siguientes lecciones.

Controles de imagen

En nuestras aplicaciones es interesante insertar imágenes para mostrar algún tipo de información adicional o para facilitar el uso de esta a los usuarios.

En un principio tenemos **4** controles que nos permiten trabajar con gráficos: cuadro de imagen (**PictureBox**), imagen (**Image**), forma (**Shape**) y línea (**Line**).

Cada uno de ellos lo utilizaremos en casos diferentes, según nos interese una u otra característica de cada objeto.

PictureBox

El cuadro de imagen (**PictureBox**) lo podemos utilizar para mostrar imágenes sueltas, aunque suele utilizarse como contenedor de otros elementos. Con esto queremos decir que dentro de un cuadro de imagen pueden existir otros elementos que dependen de él. Si nosotros movemos el cuadro de imagen con elementos en el interior, todos ellos se moverán junto con él. Si mirásemos la propiedad **Top** y **Left** de cualquier elemento que está insertado dentro de un cuadro de imagen veríamos que están en relación con el borde de este y no con el borde del formulario como en la gran parte de los objetos.

Image

El control imagen (**Image**) solo lo utilizaremos para mostrar imágenes en su interior. No se puede utilizar como contenedor como en el caso del **PictureBox**.

En nuestra práctica vamos a insertar una imagen en la que mostraremos un icono que tendrá como dibujo el signo igual (=).

Practica 1 (continuación de la lección anterior)

1. Abre el proyecto con el que estuvimos trabajando en la lección anterior.

Recuerda lo que teníamos hecho hasta este momento.

Vamos a insertar un control **Image**, para poder poner una imagen con un signo igual (=) en su interior.

2. Haz doble clic en el control **Image**.

Observa como en el centro de nuestro proyecto ha aparecido un nuevo objeto con unas líneas discontinuas que lo delimitan.

3. Accede a la propiedad: **Picture**.

Observa que aparece la palabra (**Ninguno**), esto nos indica que en este momento el objeto **Picture** no está mostrando ningún tipo de imagen.

4. Haz un clic sobre el botón con tres puntos suspensivos que aparece a la derecha de

esta propiedad.

Acto seguido verás como te aparece un cuadro de diálogo donde podrás seleccionar la imagen que quieres que se muestre. Observa que en el apartado: **Archivos de tipo** está escrita la frase **Todos los archivos de imágenes**, esto quiere decir que en este cuadro de diálogo nos aparecerán todos los archivos que tengan como extensión (**Bmp, dib, gif, jpg, wmf, emf, ico y cur**).

5. Accede al directorio **Graphics\Icons\Misc** dentro del directorio donde tengas instalado **Visual Basic 6**, y selecciona el archivo **misc22.ico**.

Observa como el tamaño de dicho objeto se ha modificado automáticamente. Observa también como en la propiedad **Picture** ahora aparece la palabra (**icono**) esto nos da a entender que el objeto que se está mostrando en este momento es un icono.

Cambio de tamaño de un objeto Image

Nosotros en este momento podemos cambiar un poco el aspecto de nuestro icono.

6. Sitúate en uno de los vértices de este objeto.

7. Arrastra hasta aumentar el tamaño del objeto.

Observa como el dibujo no ha sufrido ninguna modificación. Solo ha cambiado el tamaño del control, pero no el del dibujo.

8. Accede a la propiedad **Stretch** y pon su valor a **True**.

Observa como ahora el dibujo ocupa todo el tamaño del objeto. Puede ser que el dibujo se vea deformado.

Para hacer que el dibujo vuelva a su tamaño real, nada más fácil que realizar el siguiente paso.

9. Pon la propiedad **Stretch** a **False**.

Observa como tanto el tamaño del icono, como el del objeto han cambiado.

10. Mueve el objeto **Image** hasta la posición: **3840, 1560**.

Puedes hacerlo utilizando cualquiera de los métodos explicados hasta el momento.

Bordes en la imagen

Una vez colocado nuestro objeto **Image** en su sitio vamos a modificar su contorno.

11. Accede a la propiedad **BorderStyle** y modifica el contenido de **None** a **Fixed Single**.

Observa como ha aparecido un borde en **3D** que rodea a todo el objeto. Con la propiedad **Appearance** podrás hacer que este borde mantenga la apariencia de **3D** o solo sea un cuadro con una línea simple. Deja esta propiedad como está.

Apariencia del ratón

Vamos a modificar la apariencia del ratón cuando pase por encima de este obje-

to. Esto lo haremos para que los usuarios no piensen que se debe pulsar esta imagen para conseguir el resultado de la operación. En un principio si nosotros no ponemos ninguna línea de código dentro del evento **Click** de este objeto no debería pasar nada al pulsar un clic, pero puede ser que el usuario piense que la aplicación funciona incorrectamente por lo que seguiría intentándolo. En cambio nosotros podemos mostrar un icono que facilite la utilización de la aplicación.

12. Accede a la propiedad **MousePointer**.

Aquí especificaremos que tipo de cursor queremos que aparezca en el momento en el que el cursor pasa por encima del objeto.

Tenemos diferentes tipos de cursores:

Valor	Descripción
0	Predeterminado.
1	Flecha.
2	Cruz
3	Forma de I.
4	Pequeño cuadrado dentro de otro cuadrado.
5	Flecha de cuatro puntas.
6	Flecha doble que apunta al NE y al SE.
7	Flecha doble que apunta al N y al S.
8	Flecha doble que apunta al NO y al SE.
9	Flecha doble que apunta al O y al E.
10	Flecha hacia arriba.
11	Reloj de arena.
12	No colocar.
13	Flecha y reloj de arena.
14	Flecha y signo de interrogación.
15	Ajustar todo.
99	Icono especificado en la propiedad MouseIcon .

Esta tabla está extraída de la ayuda de **Visual Basic**. En ella se especifican las diferentes formas que puede tomar el cursor al pasar por encima del objeto. Para poder ver con más exactitud cada una de las formas es recomendable ir seleccionando cada una de ellas e iniciar una ejecución de prueba. Esta propiedad no es visible en el modo de diseño.

13. En nuestro caso deberemos seleccionar: **12 - No Drop**.

14. Ejecuta la aplicación y coloca el cursor sobre el objeto **Image**.

Controles de gráficos

Los controles gráficos **Line** y **Shape** son mucho más simples que los que hemos visto anteriormente, pero nos ayudan a diseñar nuestra aplicación.

Line

Elemento que nos dibuja una línea en nuestro formulario. Este elemento no contiene eventos, sólo se pueden utilizar de forma decorativa.

Nosotros podemos añadir una línea haciendo un doble clic sobre el objeto **Line** o haciendo, en primer lugar, un clic sobre el objeto **Line**, después marcando el primer punto de la línea, mantener pulsado el botón del ratón y soltarlo en el momento en el que queramos el punto final. Después para modificar el tamaño solo deberemos situarnos sobre una de las puntas de la línea, hacer clic con el botón derecho y mientras lo mantienes pulsado movernos hasta la nueva posición. Si lo que queremos es mover toda la línea, manteniendo la inclinación y el tamaño, arrastraremos la línea haciendo

clic en cualquier parte de ella. Otra manera que tenemos para mover los puntos iniciales y finales de la línea es utilizando las propiedades **X1**, **X2**, **Y1** y **Y2**.

X1 nos marca la distancia del primer punto con la parte izquierda del formulario. **X2** es igual que **X1** pero se refiere al segundo punto de la línea. **Y1** nos marca la distancia del primer punto con la parte superior del formulario. **Y2** es igual que **Y1** pero haciendo referencia al segundo punto de la línea.

Podemos utilizar la propiedad **Visible** para mostrar (**True**) u ocultar la línea (**False**). La propiedad **DrawMode** modifica la apariencia de la línea. **BorderWidth** nos modifica el ancho de la línea y con **BorderColor** podemos modificar el color de la línea.

Puedes insertar una línea en el formulario con el que estamos trabajando donde creas conveniente y modificarla a tu gusto.

Shape

Con este control podemos insertar en nuestro formulario un rectángulo, un cuadrado, una elipse, un círculo o un rectángulo o cuadrado con las esquinas redondeadas.

Al igual que en el caso del control **Line**, este elemento no contiene eventos, solo nos sirve para decorar nuestros formularios.

15. Inserta un objeto **Shape** en nuestro formulario.

16. Muévelo hasta la posición **120, 480** y cámbiale el tamaño a **5775 x 2655**.

Cuando nosotros hemos insertado este objeto hemos obtenido un rectángulo, pero como ya hemos dicho anteriormente, nosotros podemos obtener diferentes formas geométricas. Esto lo conseguiremos modificando la propiedad **Shape**. Tenemos **6** posibilidades: rectángulo (**Rectangle**), cuadrado (**Square**), elipse (**Oval**), círculo (**Circle**), rectángulo con las esquinas redondeadas (**RoundedRectangle**) o cuadrado con las esquinas redondeadas (**RoundedSquare**).

17. Prueba cualquiera de las opciones de la propiedad **Shape**. Al final deja **Rectangle**.

18. Cambia la propiedad **BackStyle** de transparente a opaco. Observa que ha ocurrido.

Cambiando la apariencia

El rectángulo que nosotros hemos insertado se ha rellenado de color blanco y muchos de los objetos que están dentro de él han desaparecido. Bien, si queremos que todos los objetos vuelvan a verse lo que tenemos que hacer es "empujarlo" hacia el fondo del formulario, para que los demás objetos pasen a estar por encima de él.

19. Sitúate sobre el borde del rectángulo.

20. Pulsa el botón derecho del ratón para que aparezca el menú contextual.

21. Selecciona la opción **Enviar al fondo**.

Observa como todos los elementos han aparecido nuevamente.

Con la propiedad **BackColor** se puede modificar el color del fondo del rectángulo, en cambio si lo que deseas es modificar el color del borde utiliza **BorderColor**.

A este objeto también le puedes añadir una trama en lugar de un color opaco.

Si modificas la propiedad **FillStyle** podrás obtener diferentes tipos de tramas: Trama sólida (**Solid**), transparente (**Transparent**), líneas horizontales (**Horizontal**)

Line), líneas verticales (**Vertical Line**), líneas diagonales de izquierda a derecha (**Upward Diagonal**), líneas diagonales de derecha a izquierda (**Downward Diagonal**), en cruz (**Cross**) y líneas diagonales cruzadas (**Diagonal Cross**)

22. Coloca una de estas tramas.

23. Modifica el color de la trama con la propiedad: **FillColor**.

24. Una vez visto los cambios quita la trama. Poniendo la propiedad **FillStyle** a **Transparent**.

Inserta otro **Shape** en la posición **120, 3240** con un tamaño **5775 x 2055**. Modifica todas las propiedades pertinentes para que queden como en el caso anterior.

Frame

A nosotros nos interesa poder poner dos nuevos **OptionButton** en nuestra aplicación, que utilizaremos para marcar si lo que queremos que aparezca en la lista, que más adelante añadiremos, es el resultado de la operación o toda la operación completa.

Como ya dijimos en lecciones anteriores en un mismo formulario pueden existir el número de **OptionButton** que deseemos, pero solo puede estar activado uno de ellos simultáneamente. En nuestra práctica esto no nos interesa ya que por un lado deberemos indicar una de las 4 operaciones que deseamos realizar y por otro que es lo que deseamos que se añada a la lista. Veamos esto en la práctica.

25. Añade un nuevo **OptionButton**.

26. Sitúalo en algún sitio del formulario que no te moleste.

Este nuevo objeto es el que utilizaremos para marcar que es lo que deseamos ver en nuestra **Lista de operaciones** es el resultado de la operación.

27. Inicia una ejecución de prueba.

28. Selecciona la resta como operación ha efectuar.

Ahora nosotros antes de pulsar el botón calcular deberíamos indicar que es lo que querríamos ver en la **Lista de operaciones**.

29. Marca el **OptionButton** que hemos insertado anteriormente.

Observa como la selección que teníamos en la resta ha desaparecido. En el momento que tuviéramos que realizar la operación, el programa no nos enseñaría ningún tipo de solución ya que no está seleccionada ninguna de las operaciones.

30. Marca ahora la multiplicación.

Observa como ha desaparecido la selección del último **OptionButton** que habíamos insertado.

31. Detén la ejecución del programa.

Colocar un Frame

Este problema se puede solucionar insertando un nuevo objeto llamado **Frame**. Este objeto lo que hace es mantener separados diferentes objetos **OptionButton** que se encuentran dentro de un mismo formulario. Este elemento se utiliza para que así puedan marcarse de forma independiente grupos de **OptionButton**. Eso sí, solo se

podrán marcar, uno y solamente uno de los que tenemos dentro de cada **Frame**.

Cuando nosotros coloquemos diferentes **OptionButton** dentro de un **Frame** podremos hacer que todos estos se muevan a la vez al mover el **Frame**. Esto es debido a que el **Frame** actúa como contenedor de los **OptionButton**.

Para colocar este objeto y todos los que deberá llevar en su interior crearemos, en primer lugar el **Frame**, después seleccionaremos el objeto **OptionButton** y lo dibujaremos en el interior. De esta manera unos objetos dependerán de los otros y actuarán como si de un grupo se tratase. Si no lo hacemos de esta forma no conseguiremos nuestro propósito. Veamos lo que hemos explicado con un ejemplo.

32. Elimina el **OptionButton** que creaste anteriormente.

33. Inserta un **Frame** en nuestro formulario.

Observa que el objeto insertado es un recuadro con un borde y en la parte superior izquierda aparece un texto. Aquí pondremos texto para que nos aclare la utilidad de este grupo de **OptionButton**. Si deseamos que no se vea este borde al ejecutar la aplicación deberemos poner la propiedad **BorderStyle** a **0**.

Recomendamos utilizar siempre borde ya que de esta forma hacemos que el usuario de la aplicación sepa que **OptionButton** actúan conjuntamente.

34. Mueve el **Frame** hasta la posición **3000, 3600** con un tamaño de **2775 x 735**.

35. Cámbiale el color de fondo con la propiedad **BackColor** para que sea igual que el fondo del formulario.

36. Accede a la propiedad **Caption** y escribe **Mostrar**.

Observa como **Mostrar** ha aparecido en la esquina superior izquierda del **Frame**.

Insertando objetos en su interior

Ahora vamos a colocar dos **OptionButton** en el interior de este **Frame**. Para insertar cada uno de ellos sigue los siguientes pasos, no te saltes ninguno ya que puede ser que no conseguirás que los dos controles estén dentro del **Frame**. Para hacer todos estos pasos deberás asegurarte que tienes seleccionado el **Frame**. Esto lo podrás comprobar en el momento en el que aparecen los cuadros para modificar el tamaño de este objeto.

37. Haz un clic sobre el objeto **OptionButton** del cuadro de herramientas.

Si mueves el ratón hasta colocarte sobre el formulario podrás observar como este ha tomado forma de cruz.

38. Coloca el puntero de ratón dentro de nuestro **Frame**.

39. Pulsa el botón izquierdo del ratón y mientras lo mantienes pulsado muévete hasta que el nuevo objeto tenga un tamaño de **1095 x 375** aproximadamente. Cuando consigas ese tamaño ya puedes soltar el botón del ratón.

Ahora que ya tenemos colocado el primer **OptionButton** vamos a moverlo a su posición.

40. Sitúa nuestro primer **OptionButton** a la posición **240, 240**.

Observa que esta posición tiene referencia con el **Frame** que lo contiene y no con el resto del formulario.

41. Coloca otro **OptionButton** dentro de nuestro **Frame** tal y como hemos hecho en los pasos: 37, 38 y 39. Recuerda tener seleccionado el **Frame**.

42. Sitúalo en la posición **1440, 240**.

43. Selecciona el objeto **Option1**, ponle de **(Nombre): Resultado** y de **Caption: Resultado**. Cámbiale el color de fondo con **BackColor** para que tenga el mismo que el formulario.

44. Selecciona el objeto **Option2**, ponle de **(Nombre): Operación** y de **Caption: Operación**. Cámbiale el color de fondo para que tenga el mismo que el formulario.

45. Asegúrate que la propiedad **Value** de **Resultado** está a **True**.

Vamos a comprobar como estos dos últimos objetos que hemos insertado dependen del **Frame**.

46. Haz un clic sobre el borde del **Frame**.

47. Muévelo hasta cualquier otra posición del formulario.

Observa como al moverlo también has movido los dos objetos que hay en su interior.

48. Vuélvelo a colocar en la misma posición que estaba. (Mira el punto 34).

ComboBox (Lista desplegable)

Un **ComboBox** tiene características comunes de un **TextBox** y de un **ListBox**. Un **TextBox** ya que se puede escribir texto en el recuadro de texto que aparece y de un **ListBox** ya que podemos seleccionar uno de los elementos que aparecen en la lista desplegable de dicho control.

En nuestro caso utilizaremos este nuevo control para hacer que el usuario escoja entre: **Añadir a la lista** y **No añadir a la lista**. Con lo que añadirá o no a la lista, que insertaremos después, el resultado o la operación completa que hemos realizado anteriormente.

49. Pulsa un doble clic sobre **ComboBox**, en la barra de herramientas.

Observa como en todos los casos que hemos querido insertar un objeto y hemos hecho un doble clic el objeto se ha colocado en el centro del formulario.

50. Muévelo hasta la posición **3360, 4800** con un tamaño de **1935 x 315**.

51. Accede a la propiedad **(Nombre)** y escribe **Añadir**.

Insertando elementos a la lista desplegable

52. Accede a la propiedad **List**, pulsa en el botón con una flecha hacia abajo que verás a la derecha de esta propiedad.

Todo lo que escribamos aquí será lo que aparecerá cuando nosotros despleguemos el **ComboBox** que acabamos de insertar. Cada línea corresponde a un elemento diferente.

53. Escribe: **Añadir a la lista**.

Nosotros en este momento ya hemos insertado uno de las dos líneas que debe aparecer dentro de este objeto. Si queremos insertar el siguiente elemento lo tenemos que escribir en la siguiente línea de esta lista. Para ello no deberemos pulsar **Intro** ya que si lo hacemos nos saldríamos de la propiedad **List**. Para seguir insertando elementos deberemos pulsar las teclas **Ctrl** e **Intro** conjuntamente.

54. Pulsa **Ctrl** e **Intro**.

55. Escribe: **No añadir a la lista**. Pulsa **Intro** al terminar.

Bloqueamos el objeto.

Como hemos dicho anteriormente este nuevo objeto tiene la propiedad de poder escribir y añadir elementos en su interior. En nuestro caso no nos interesa que el usuario modifique o escriba en el cuadro de texto, ya que solo queremos que utilice una de las dos propiedades que contiene.

56. Accede a la propiedad **Locked** y pon su valor a **True**.

Iniciar contenido

Para hacer que nuestro objeto ya se inicie con algún contenido en su interior solo tenemos que introducir texto dentro de la propiedad **Text**

En nuestro caso podemos hacer que se inicie con la opción que no inserta ningún tipo de operación en la lista.

57. Accede a la propiedad **Text** y escribe **No añadir a la lista**.

58. Realiza una ejecución de prueba. Despliega este objeto e intenta seleccionar alguna de las opciones que contiene en su interior.

Podrás observar que en el programa no ocurre nada de nada ya que todavía no hemos insertado las líneas de código pertinentes.

59. Detén la ejecución de prueba.

ListBox (Lista)

Un **ListBox** es un elemento que nos muestra una lista de elementos de los que el usuario de la aplicación puede escoger uno o más de ellos. Normalmente si el número de elementos que hay dentro de la lista excede del espacio que hemos reservado para la visión del contenido de esta aparecen unas barras de desplazamiento para podernos mover con facilidad sobre la lista.

Cada elemento de la lista tiene un número que nos indica el lugar que ocupa. Es muy importante tener en cuenta que el primer elemento de la lista tiene como índice **0**. La propiedad que nos indica el índice de cada elemento es **ListIndex**. Si no hay ningún elemento en la lista el valor de esta propiedad es **-1**. Utilizando esta propiedad podremos saber en que momento existen o no elementos dentro de la lista.

Si nosotros quisiéramos saber cuantos elementos hay en la lista utilizaríamos la propiedad **ListCount**.

60. Haz un doble clic sobre el elemento **ListBox**.

Observa que aparentemente tiene la misma estructura que un cuadro de texto.

61. Sitúalo en **240, 3600** con un tamaño de **2655 x 1620**.

62. Accede a la propiedad **(Nombre)** y escribe **ListaOperaciones**.

Recuerda que en la propiedad **(Nombre)** no pueden existir espacios en blanco.

Observa que en el interior de la lista ha aparecido el **(Nombre)** que hemos escrito anteriormente. Este objeto no tiene ni propiedad **Caption** ni **Text** con la cual cosa, si nosotros quisiéramos introducir algún contenido en nuestra lista tendríamos que hacerlo con la propiedad **List** al igual que hemos hecho con el **ComboBox** anteriormente. En nuestro caso no introduciremos ningún tipo de texto ya que los iremos insertando durante la ejecución de la aplicación.

Al realizar la ejecución del programa el texto que vemos en el interior de esta lista en el modo edición no aparece.

Podemos hacer que a medida que se introducen los valores en la lista se ordenen automáticamente. Esto lo haríamos poniendo la propiedad **Sorted** a **True**. Puedes activarla si lo deseas.

Nosotros podemos permitir que nuestro usuario pueda seleccionar uno o varios elementos que aparezcan en nuestra lista, para realizar algún tipo de operación con los elementos seleccionados. En nuestra aplicación esto no tiene ningún tipo de importancia ya que después de añadir elementos en la lista, el usuario no puede hacer ningún tipo de operación con estos elementos.

Para modificar esta opción deberemos utilizar la propiedad **MultiSelect**. Esta propiedad tiene tres valores: **None**: que solo nos permite hacer la selección de un solo objeto, **Simple**: nos permite hacer una selección múltiple haciendo clic con el ratón sobre cada elemento que queremos seleccionar y **Extended**: también nos permite hacer selecciones múltiples pero tendremos que utilizar las teclas **Mayúsculas** o **Ctrl** juntamente con el botón del ratón.

En nuestro caso dejaremos el valor de la propiedad **MultiSelect** a **None**. Ya que no haremos ningún tipo de operación al seleccionar los elementos que aparezcan en la lista.

Bloquear controles

Una vez tenemos todos y cada uno de los elementos que forman parte de esta pequeña aplicación, vamos a bloquear los controles para que de forma fortuita no los movamos por el formulario cambiando así su posición.

63. Selecciona **Bloquear controles** dentro de **Formato**.

Para bloquear los elementos no deberás tener seleccionado ninguno de los elementos que forman parte del formulario.

Si accedes a cualquier objeto que forma parte de nuestro proyecto e intentas moverlo verás que es completamente imposible. Con esta opción activada solo podrás acceder al código de cada elemento haciendo un doble clic en el elemento deseado.

Líneas de código

En esta practica solo te pediremos que copies las líneas de código que te mostraremos a continuación en los elementos que te indiquemos, no explicaremos ninguna instrucción, ni ninguna estructura, ya que esto lo haremos en lecciones posteriores.

Intenta averiguar para que sirven cada una de las líneas de código que forma parte de esta aplicación, si no las entiendes, tranquilo, más adelante sabrás para que sirven y como funcionan cada una de ellas. Conforme vas introduciendo las líneas de código observa las ventanas de ayuda que te van apareciendo en pantalla, familiarízate con ellas. Están explicadas en lecciones posteriores.

64. Haz doble clic sobre el botón **Borrar**.

65. Escribe las siguientes líneas, recuerda que la primera y la última de ellas no debes escribirlas ya que te las mostrará el ordenador.

```
Private Sub Borrar_Click()
    PrimerOperando.Text = «»
    SegundoOperando.Text = «»
    MostrarResultado.Caption = 0
    PrimerOperando.SetFocus
End Sub
```

Este botón lo utilizamos para iniciar la calculadora.

66. Pulsa **Mayúsculas + F7** para visualizar el formulario.

67. Accede a la ventana de código del botón **Calcular** y escribe las siguiente líneas de código:

Observa que al principio de algunas líneas aparece este símbolo (*), no debes copiarlo. Este símbolo quiere decir que la línea que aparece a continuación va seguida, en la misma línea, a la anterior. Ten cuidado con esto ya que una misma instrucción debe ocupar una sola línea, más adelante veremos como escribimos una instrucción en diferentes líneas.

```
Private Sub Calcular_Click()
    Dim Operador As String
    If PrimerOperando.Text = "" Or SegundoOperando.Text = "" Then
        MsgBox ("Falta algún operando")
        Exit Sub
    End If
    If Sumar.Value = True Then MostrarResultado.Caption =
(*)      Val(PrimerOperando.Text)      +
Val(SegundoOperando.Text)
    If Restar.Value = True Then MostrarResultado.Caption =
(*)      Val(PrimerOperando.Text)      -
Val(SegundoOperando.Text)
    If Multiplicar.Value = True Then
        MostrarResultado.Caption (*) = Val(PrimerOperando.Text)
        * Val(SegundoOperando.Text)
    If Dividir.Value = True Then
        If Val(SegundoOperando.Text) = 0 Then
            MsgBox ("No se puede dividir entre 0")
            Exit Sub
        End If
        MostrarResultado.Caption = Val(PrimerOperando.Text)
        / (*) Val(SegundoOperando.Text)
    End If
    If Añadir.Text = "Añadir a la lista" Then
        If Operacion.Value = True Then
```

```

If Sumar.Value = True Then Operador = "+"
If Restar.Value = True Then Operador = "-"
If Multiplicar.Value = True Then Operador = "*"
If Dividir.Value = True Then Operador = "/"
ListaOperaciones.AddItem PrimerOperando.Text &
(*) Operador & SegundoOperando.Text & "=" &
(*) MostrarResultado.Caption
Else
ListaOperaciones.AddItem
MostrarResultado.Caption
End If
End If
End Sub

```

Vamos a comentar por encima que es lo que realiza este botón: Antes de mirar que operación tenemos seleccionada comprobamos que el usuario haya puesto algún número dentro de los dos operadores, si falta alguno aparece un mensaje de error en la pantalla informando al usuario. Después miramos cual de las operaciones está activada, si es la **suma**, la **resta** o la **multiplicación** se realiza la operación sin ningún tipo de problema. Miramos si la operación que debemos realizar es la **división**, si es así miramos que el segundo operador no sea igual a 0 ya que esto nos podría dar un error y el programa abortaría¹. Si es así nos aparece un nuevo mensaje que nos informa que no se puede realizar una división entre 0, evitando así el error y el aborto del programa. Una vez realizada la operación miramos si está seleccionada la opción: **Añadir a la lista**, si no está seleccionada no pasa absolutamente nada, en cambio si esta opción está activada pasamos a mirar si debemos añadir a la lista es la operación completa o solo el resultado. Acto seguido añadimos lo que corresponda a la lista.

68. Guarda el proyecto.

69. Crea un archivo ejecutable.

70. Realiza una ejecución de prueba para poder observar como funciona la aplicación.

Te recomendamos que intentes averiguar para que sirven cada una de las líneas de código que hemos escrito para que funcione la aplicación. Más adelante las entenderás todas sin ningún tipo de problema.

Fin de la lección 3.

¹ Terminar la ejecución de un programa de forma incorrecta.