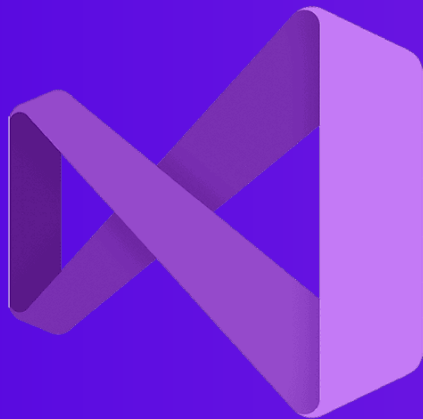


Formularios en Visual Basic



Visual Basic

Septiembre 2023

Índice

1. Generación de formularios en Excel (VBA)



1 |

Generación de formularios en Excel (VBA)

Formularios. Controles

Un formulario es una Ventana o cuadro de diálogo (objeto) que contiene otros controles (objetos) como cuadros de texto, botones, cuadros combinados así como el código asociado para gestionarlos.

Cada control debe tener un nombre mediante el cual se puede acceder al control desde el código. Aunque el sistema propone nombres por defecto, es conveniente que el usuario elija sus propios nombres.

Un formulario se inserta en el entorno Visual Basic desde el menú:

Insertar => UserForm

El formulario se construye en modo diseño pero , para su ejecución, **debe existir una macro en un módulo del libro que realice la activación del formulario.**

Formularios. Controles

Ejemplos de controles dentro de un formulario:

 **Label**. Etiqueta (un simple rótulo).


 **TextBox**. Cuadro de texto.

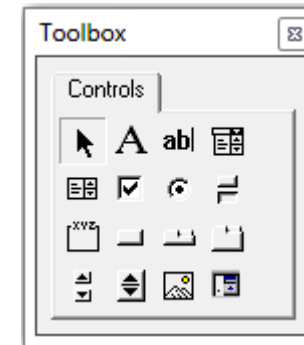
 **ComboBox**. Cuadro combinado o desplegable.

 **CommandButton**. Botón.

 **CheckBox**. Casilla de verificación.

 **OptionButton**. Botón de opción.

 **MultiPage**. Página múltiple. Para formularios con varias pestañas (gran número de controles).



Lista de controles disponibles en el Cuadro de Herramientas (mostrar desde el menú Ver).

Formularios. Propiedades

Los formularios y sus controles contenidos tienen características como color, tamaño y tipo de letras, color de fondo, etc, que se gestionan mediante sus **propiedades**.

Todos los controles tienen sus propiedades definidas por los mismos nombres y éstas son como variables en las que configuramos sus características. Cada propiedad tiene un nombre que ya está definido en el sistema. Como ejemplo de propiedades:

Name. Nombre del objeto, con él será referido el control desde el código

Caption. Rótulo que aparece visible en el objeto

BackColor. Color del fondo

ForeColor. Color del primer plano

Font. Tamaño y tipo de letra

Visible. Propiedad que indica si el control está visible (True) o no (False)

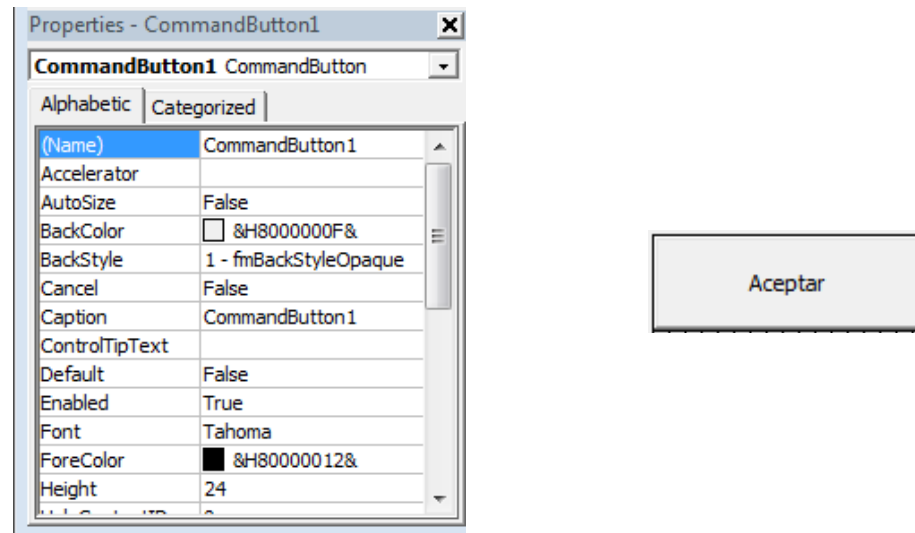
Value. Propiedad que contiene el valor del control. Es, por ejemplo, muy importante en controles como un cuadro de texto, pues almacena lo que el usuario haya introducido en el cuadro.

Para acceder a las propiedades desde el código se pone el nombre del control, seguido del punto y seguido de la propiedad. Ejemplo:

txtNominal.Value

Formularios. Propiedades

La lista de propiedades de un objeto, cuando está seleccionado, se pueden observar en la Ventana de Propiedades. Por ejemplo, las propiedades de un botón (CommandButton):



El entorno de Visual Basic en Excel permite configurar manualmente estas propiedades durante el diseño del formulario sin necesidad de introducir líneas de código. Algunas propiedades pueden ser configuradas en tiempo de ejecución mediante líneas de código, como los elementos de un cuadro combinado.

Por ejemplo, para acceder desde código Visual Basic:

```
cmdBoton1.Caption = "Aceptar"
```

Formularios. Métodos

Los métodos de un formulario son funciones que ya vienen construidas dentro del control, no tiene que construirlas el usuario.

Los métodos realizan las funciones típicas asociadas al formulario y sus controles.

Cada control tiene su propio conjunto de funciones y se podrán ejecutar desde el código, realizando la correspondiente llamada.

Ejemplo de métodos:

Show: Muestra el objeto, lo activa (y lo carga en memoria si no lo estaba).

Hide → Oculta el objeto.

RePaint → Finaliza las actualizaciones pendientes de los controles del formulario.

SetFocus → Poner el foco en el objeto, hacerlo activo.

Copy

Paste

AddItem → Añadir elementos a un cuadro combinado.

La ejecución de los métodos se realiza poniendo una línea de código que contenga el nombre del objeto, seguido del punto y seguido del nombre del método.

Como ejemplo, el código que debe estar en una macro de un módulo para cargar un formulario si su nombre es Frm1:

Frm1.Show

Formularios. Eventos

Las acciones de un usuario sobre un formulario que está activo y mostrándose en pantalla se llaman eventos (no confundir con los métodos).

Son eventos hacer clic sobre un botón, un doble clic sobre un formulario para maximizarlo, pulsar en botón Intro, etc.

Activate, Deactivate → El evento Activate se produce cuando un objeto pasa a ser la ventana activa. El evento Deactivate se produce cuando un objeto deja de ser la ventana activa. Un objeto puede pasar a activo utilizando el método Show en código.

Click → Se produce cuando se hace clic en un control con el ratón.

Initialize → Evento del formulario que se produce la primera vez que se carga un formulario.

Change → Se produce cuando cambia el valor de un objeto (por ejemplo, un cuadro de texto o un cuadro combinado).

Cuando se produce el evento, el sistema ejecuta la llamada al procedimiento asociado a dicho evento y es labor del programador/a declarar ese procedimiento y dotarlo del código apropiado.

Los nombres de los procedimientos de eventos se componen del nombre del objeto, seguido del guión bajo, seguido del nombre del evento. Ejemplo:

cmdBoton1_Click

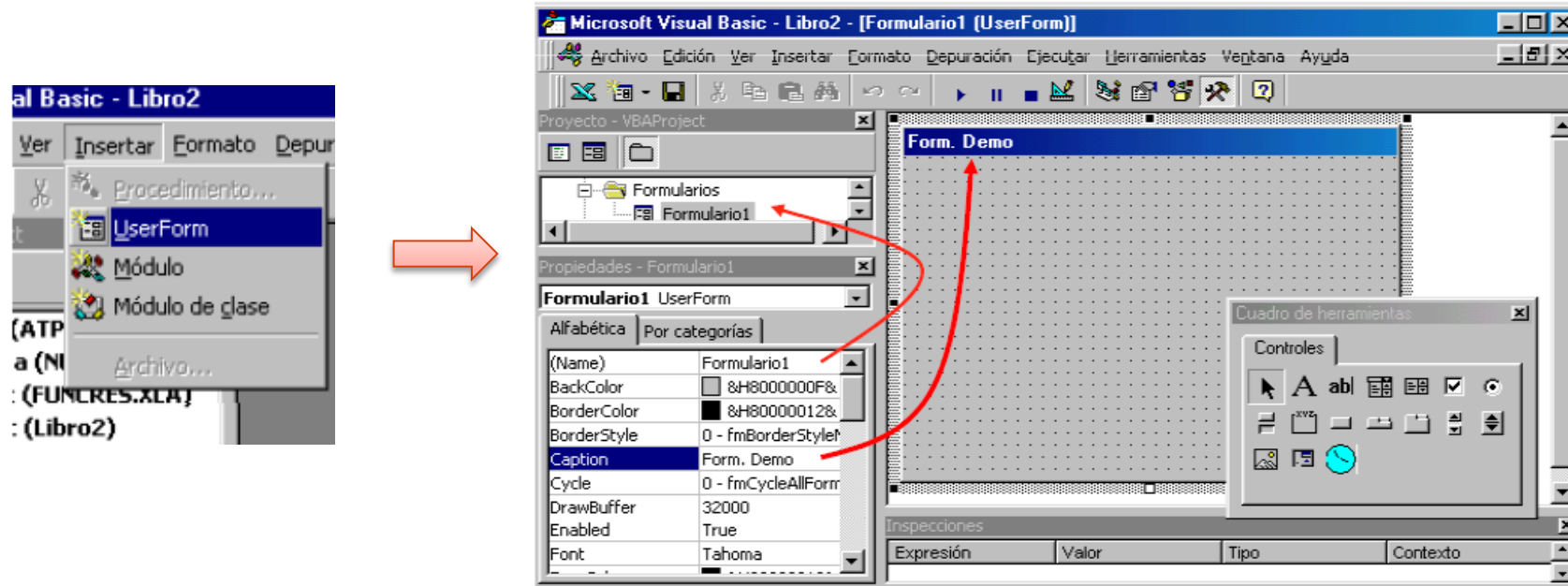
Creación de un formulario

En el editor de Visual Basic, menú **Insertar**, insertar un **UserForm**.

Desde la ventana de propiedades podremos configurar todas las características del formulario.

Desde el cuadro de herramientas podremos ir añadiendo controles al formulario, configurando sus propiedades en la ventana de propiedades. La apariencia gráfica se realiza de manera sencilla mediante el ratón (situar en el formulario, dimensionar tamaños, etc).

Cada control insertado tiene un nombre por defecto (propiedad **Name**) y un título que aparece visible (propiedad **Caption**). Debemos personalizar sus nombres.

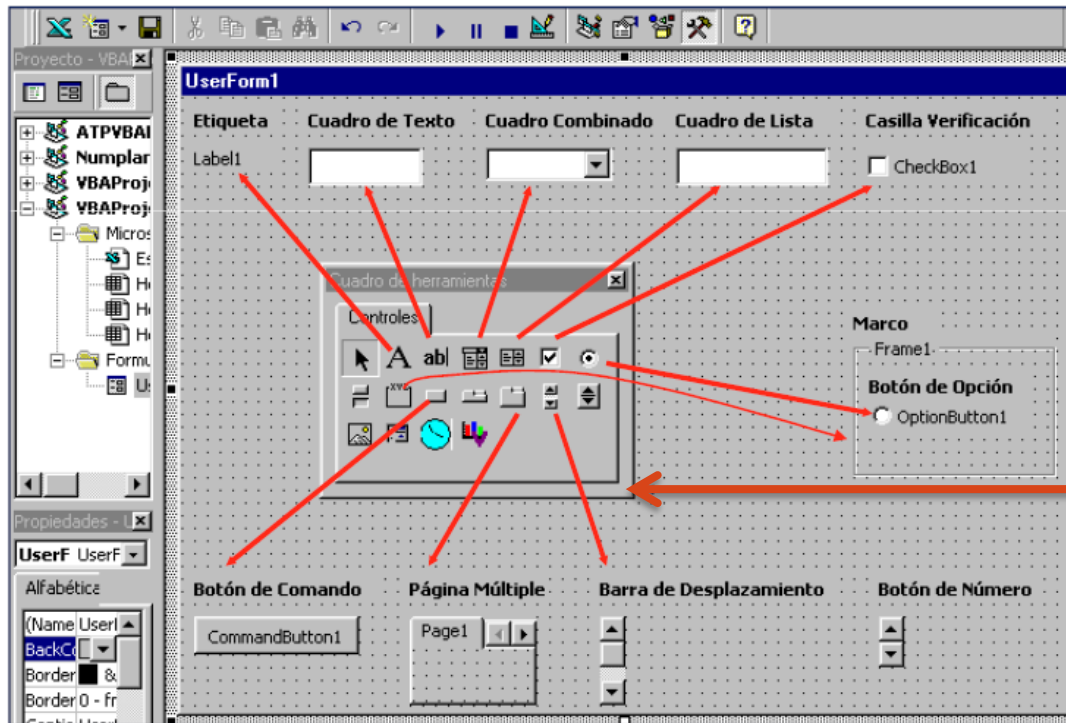


Controles de un formulario

Cada control insertado es, a su vez, un objeto. Los más habituales están referidos en el siguientes gráfico.

Para utilizarlos, hacer clic en el Cuadro de Herramientas y arrastrar en el Userform hasta darle las dimensiones deseadas.

Cada control debe tener un nombre distinto (configurarlos en propiedad Name).



Cuadro de Herramientas:

Visible cuando está seleccionado el formulario.

Si lo quitamos, se puede mostrar desde el menú **Ver** de Visual Basic.

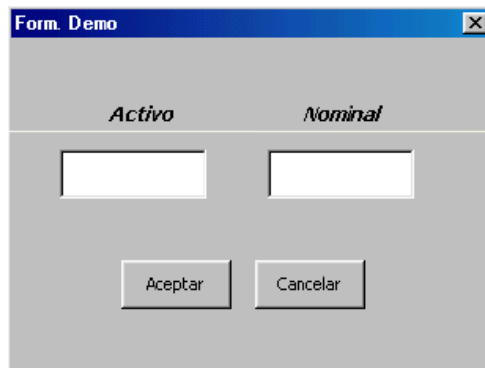
Ejercicio/ Ejemplo

Construir un Formulario que permita al usuario introducir dos datos para ser volcados a una hoja de Excel: el nombre de un activo y un nominal.

El formulario consta de dos etiquetas, dos cuadros de texto y dos botones, *Aceptar* y *Cancelar*, con las funciones:

Aceptar : se vuelcan los dos valores en Excel (hoja activa, celdas A1 y B1).

Cancelar : se cierra el formulario.



Objeto	Name	Caption
UserForm	FRM1	Form. Demo
Label	LBL1	Activo
Label	LBL2	Nominal
TextBox	txtActivo	N/A
TextBox	txtNominal	N/A
CommandButton	cmdAceptar	Aceptar
CommandButton	cmdCancelar	Cancelar

Nota: Si el formulario se llama, por ejemplo FRM1, también deberemos crear una macro, externa al formulario (en un módulo estándar de VB), para la carga del formulario. Deberá contener:

```
Load FRM1      'carga FRM1 en memoria. No necesaria, basta con hacer FRM1.Show
FRM1.Show      'muestra el formulario (y lo carga, si no estaba en memoria)
```

Ejercicio/ Ejemplo

Sobre el formulario insertado, debemos situar los siguientes controles:

Etiqueta 1: Contendrá el texto “Activo” (Caption).Nombre “LBL1” (Name).

Etiqueta 2: Contendrá el texto “Nominal” (Caption).Nombre “LBL2” (Name).

Cuadro de texto 1: Lugar para introducir el Activo.Nombre “txtActivo” (Name).

Cuadro de texto 2: Lugar para introducir el Nominal.Nombre “txtNominal” (Name).

Botón de Comando 1: Contiene el texto “Aceptar” (Caption).Nombre “cmdAceptar” (Name).

Botón de Comando 2: Contiene el texto “Cancelar” (Caption).Nombre “cmdCancelar” (Name).

De cada control se pueden cambiar sus características (colores, tipos de letra, etc) en la ventana de propiedades cuando esté seleccionado el correspondiente control.

Ejercicio/ Ejemplo

Tenemos pendiente asociar los eventos a los botones Aceptar y Cancelar.

Utilizaremos los eventos “Click”, que llaman a los procedimientos cmdAceptar_Click y cmdCancelar_Click cuando se pulsa con el ratón en los botones Aceptar y Cancelar, respectivamente.

Estos procedimientos son llamados por el sistema de manera automática cuando se produce el evento. Pero el programador debe construir el cuerpo del código que desea ejecutar en el caso de evento producido.

```
Private Sub cmdAceptar_Click()  
  
Cells(1, 1) = txtActivo.Value  
Cells(1, 2) = txtNominal.Value  
  
End Sub
```

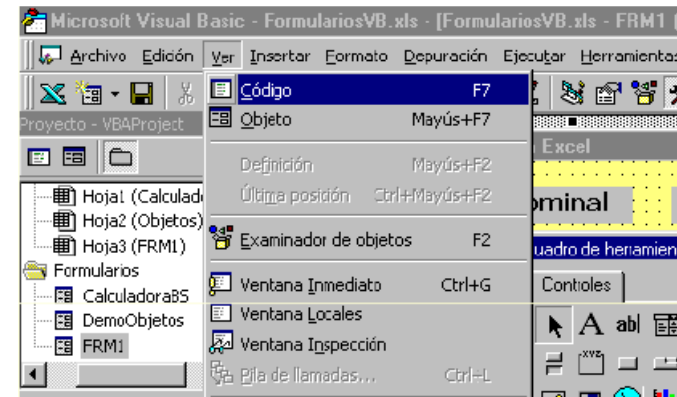
```
Private Sub cmdCancelar_Click()  
  
Unload FRM1  
  
End Sub
```

Ejercicio/ Ejemplo

Durante la construcción del formulario podremos ir desde el entorno gráfico hasta la ventana de código (para los eventos) y viceversa: desde el menú **Ver**

Desde el entorno gráfico del formulario, para acceder a la ventana de código pulsamos:

Ver => Código (F7)



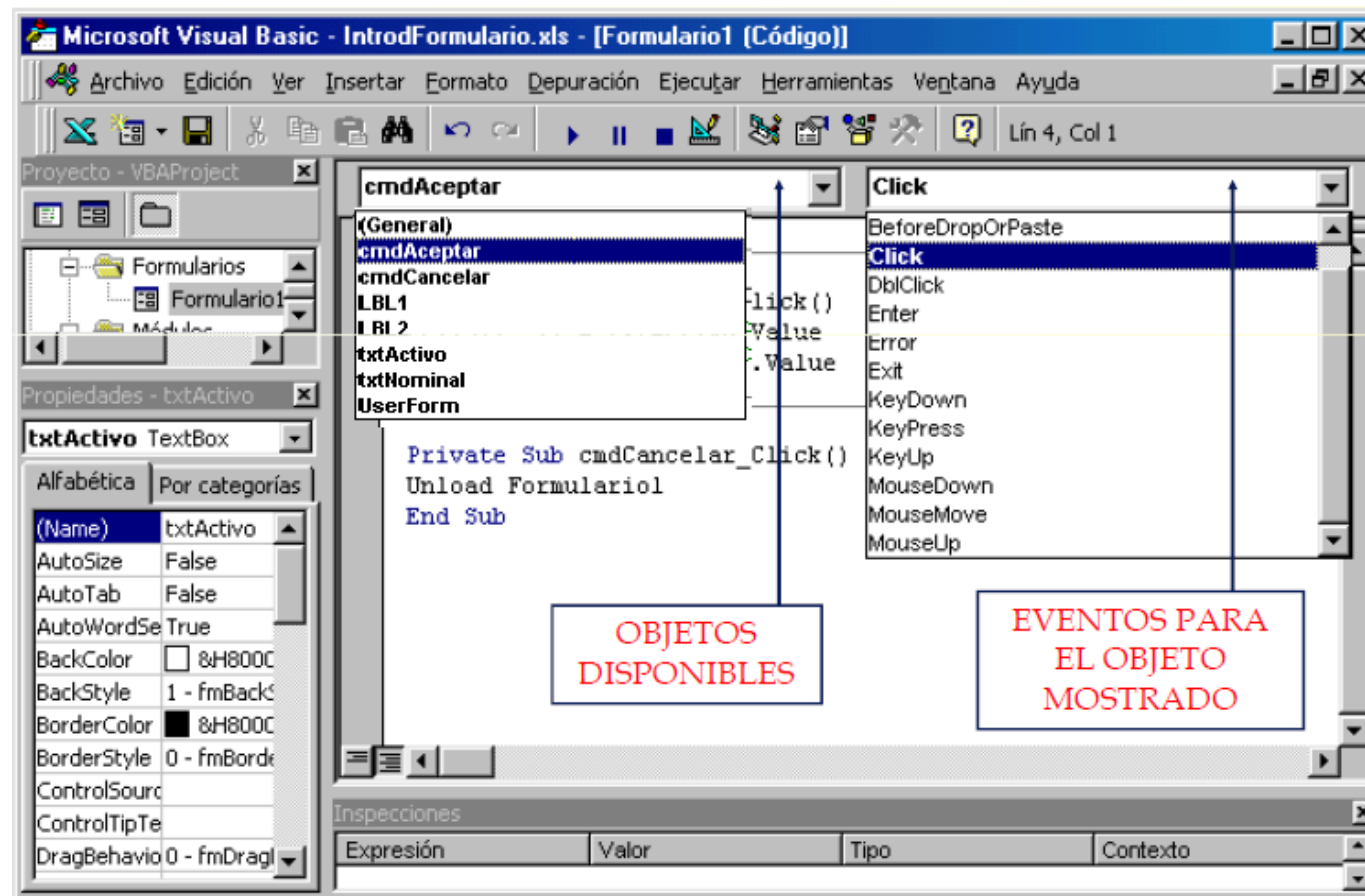
Desde la ventana de código, para acceder a la ventana de objetos (formulario) pulsamos:

Ver => Objeto(Mayus+F7)



Ejercicio/ Ejemplo

En la ventana de código podremos seleccionar todos los eventos mediante dos desplegables: uno para elegir el nombre del objeto y otro para elegir el evento asociado a dicho objeto.

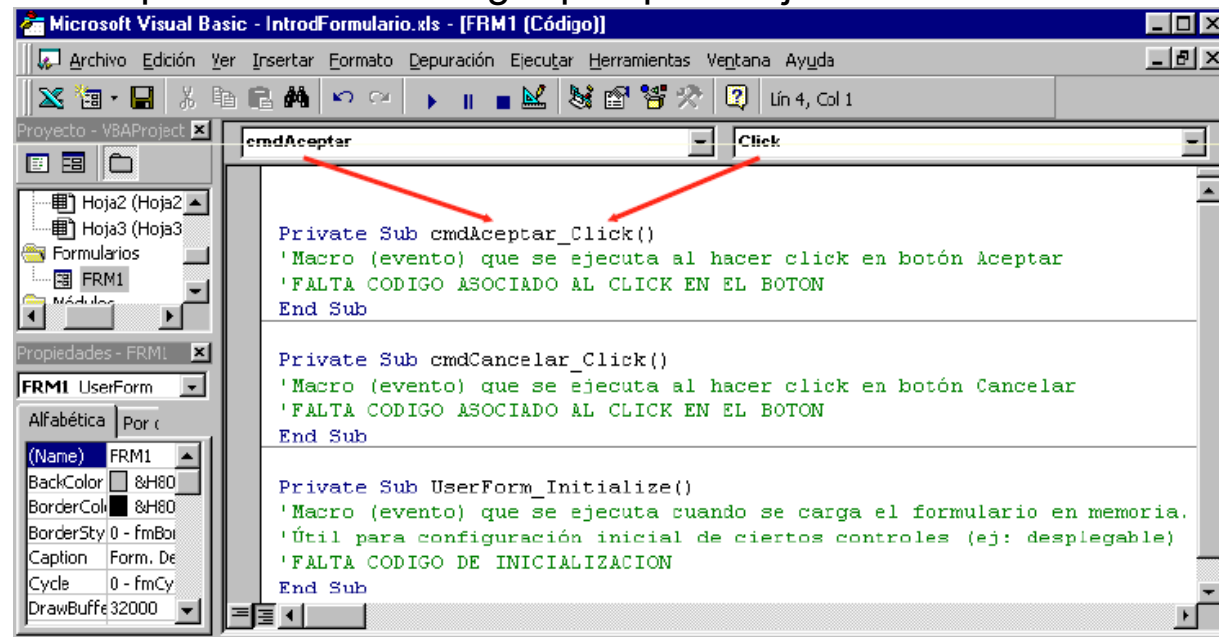


Ejercicio/ Ejemplo

No es necesario saber los nombres de los eventos (los nombres de las macros). De selección de los dos desplegables, el sistema automáticamente inserta la macro correspondiente.

En los objetos disponibles, aparecen los nombres de todos los controles que hemos colocado en el formulario. Una vez elegido el objeto desplegable de la izquierda, con el de la derecha elegiremos el evento para dicho objeto.

El usuario sólo tiene que rellenar el código que quiera ejecutar asociado al evento elegido.



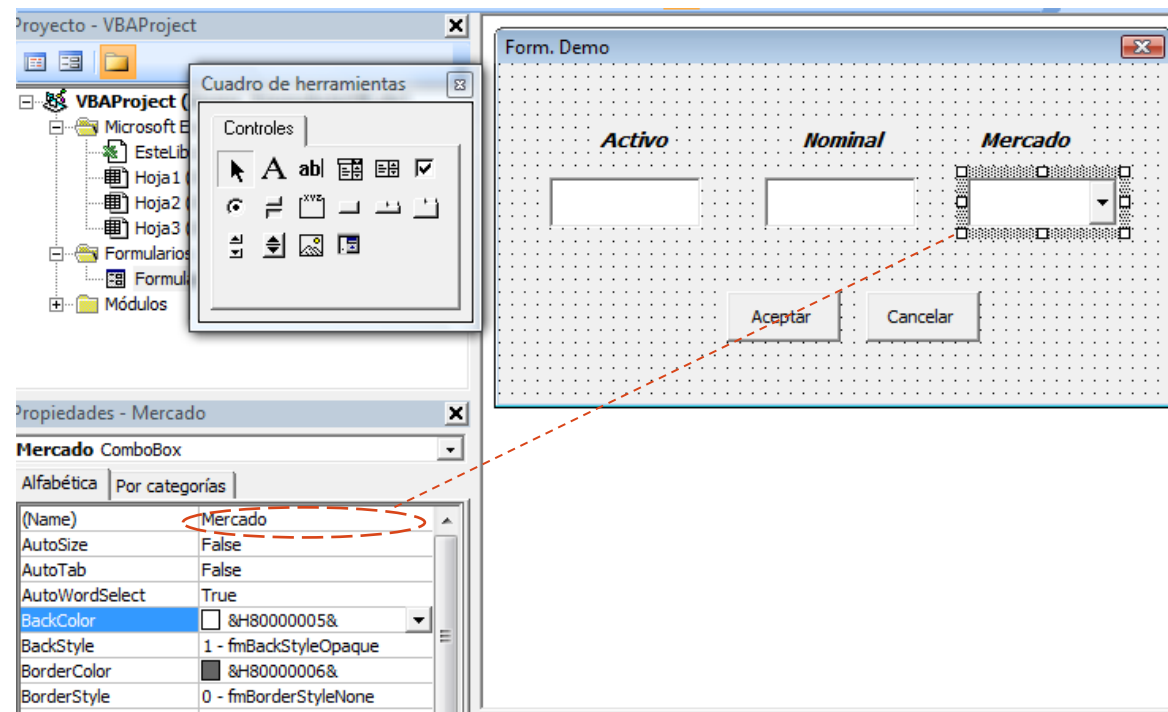
Ejercicio/ Ejemplo

Al formulario creado, podríamos añadirle un desplegable (cuadro combinado) para seleccionar el mercado (ej: Madrid, Londres, etc).

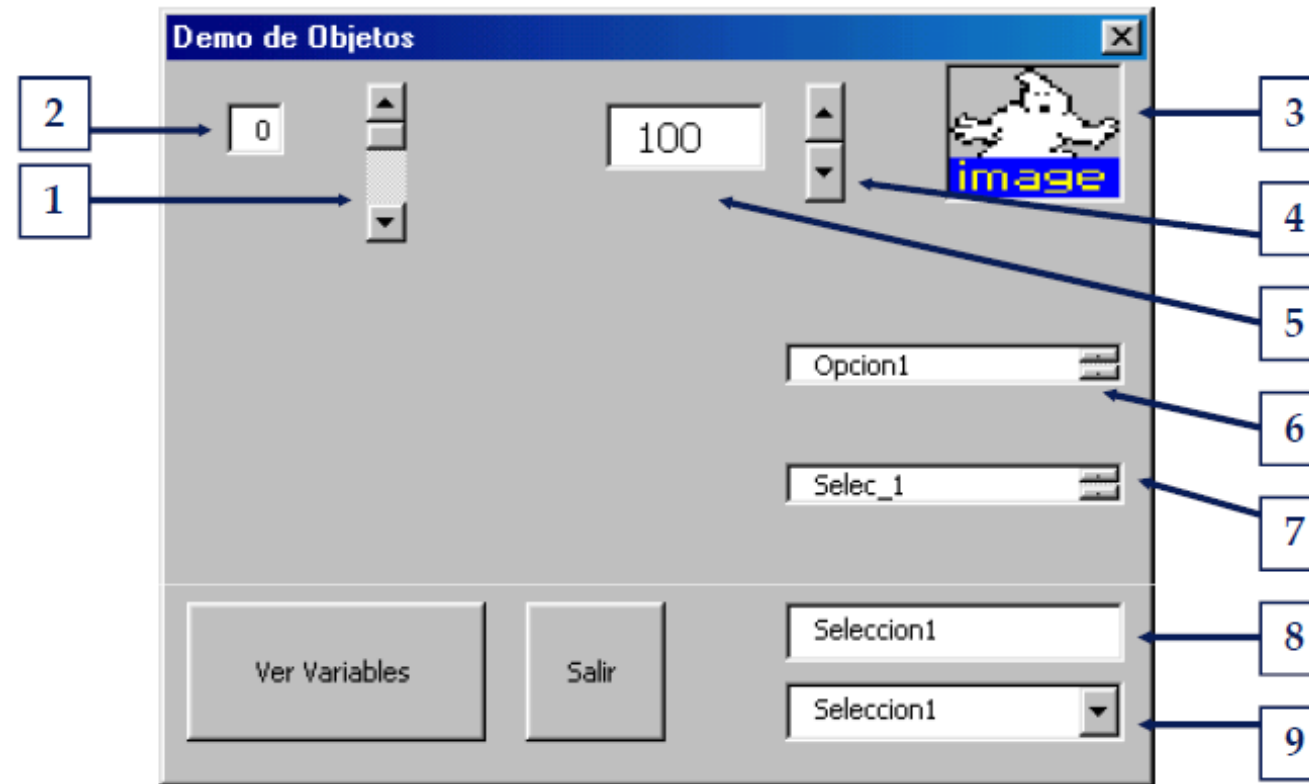
Los valores de los posibles mercado podrían estar en Excel, en una hoja del libro. Debemos configurar el origen de datos en la propiedad "RowSource" del cuadro combinado.

La propiedad
RowSource nos indica la
lista inicial de los datos
(que estará en una hoja
Excel) lo rellenamos con:
Hoja1!I1:I3

Para inicializar un formulario
usamos el evento:
UserForm_Initialize



Controles más usuales



Este es un formulario que permite mostrar el uso de los controles más habituales. Los controles 2, 5 y 8 son cuadros de texto cuyo único objetivo es mostrar el valor de los controles 1, 4 y 9 (respectivamente).

A continuación se explican cada uno de ellos.

1. Barra de desplazamiento (ScrollBar1). En sus propiedades Min y Max se fija el intervalo de valores que puede tomar. Cada vez que cambia de valor se muestra su nuevo valor en el cuadro de texto **2** (TXTB_1). Su evento asociado:



```
Private Sub ScrollBar1_Change()  
    TXTB_1.Value = ScrollBar1.Value  
End Sub
```

3. Icono de una imagen (Imagen1). Se le puede asociar el evento de “click” con en la ratón en la imagen:



```
Private Sub Imagen1_Click()  
    MsgBox “Icono de una Imagen”  
End Sub
```

4. Botón de número (Spinbutton1). Puede tomar valores entre lo indicado en sus propiedades Min y Max. Cuando cambia de valor, éste se muestra en el cuadro de texto **5** (TXTB_2).



```
Private Sub Spinbutton1_Change()  
    TXTB_2.Value = Spinbutton1.Value  
End Sub
```

6. Cuadro de lista (LST_1). Permite seleccionar entre un grupo de valores. Se inicializa mediante el método AddItem, en la carga del formulario.

7. Cuadro de lista (LST_2). Permite seleccionar entre un grupo de valores. Se inicializa desde un grupo de celdas Excel mediante la propiedad RowSource.

9. Cuadro combinado (ComboBox_1). Selección de una lista desplegable. Lo seleccionado se muestra en el cuadro de texto **8** (TXTB_4) mediante el evento Change. Se puede inicializar mediante el método AddItem.



```
Private Sub ScrollBar1_Change()  
    TXTB_1.Value = ScrollBar1.Value  
End Sub
```

Para asignar los valores iniciales necesarios, se emplea el evento Activate del formulario:

```
Private Sub UserForm_activate()  
  
    TXTB_2.Value = "100"  
    TXTB_1.Value = "0"  
  
    LST_1.AddItem ("Opcion1") 'Método de añadir opciones al Cuadro de Lista (ListBox)  
    LST_1.AddItem ("Opcion2")  
    LST_1.AddItem ("Opcion3")  
  
    COMBOB_1.AddItem ("Seleccion1") 'Añadir opciones al Cuadro Combinado (ComboBox)  
    COMBOB_1.AddItem ("Seleccion2")  
    COMBOB_1.AddItem ("Seleccion3")  
    COMBOB_1.AddItem ("Seleccion4")  
    COMBOB_1.Value = "Seleccion1" 'Asigna un valor al cuadro combinado  
  
End Sub
```

Ejercicio

Construir un formulario que se active mediante un botón desde hoja Excel, en el que el usuario elija una lista de monedas permitidas para las operaciones.

Para ello habrá dos ListBox, una tendrá la lista de monedas disponibles (que se inicializará desde un primer momento) y la otra las monedas que el usuario vaya eligiendo y añadiendo con el botón “Añadir >>”. Las monedas no podrán repetirse.

El botón “Quitar <<” eliminará de la lista de Monedas Seleccionadas la que el usuario haya marcado. Si no ha marcado ninguna deberá aparecer un mensaje indicándoselo.

El botón “Cancelar” deshabilitará el formulario.

El botón “Pegar Lista Hoja” deberá escribir las monedas que ha seleccionado el usuario en la hoja de Excel que lleva por nombre “ListaMonedasAdmitidas”, y cerrar el formulario.

Ejercicio

Monedas Permitidas

Monedas Disponibles

- EUR
- USD
- HKD
- ZAR
- SAR
- CZK
- HUF

Monedas Seleccionadas

Añadir >>

Quitar <<

CANCELAR

PEGAR LISTA FINAL

Monedas Permitidas

Monedas Disponibles

- EUR
- USD
- HKD
- ZAR
- SAR
- CZK
- HUF

Monedas Seleccionadas

- HKD
- SAR
- HUF

Añadir >>

Quitar <<

CANCELAR

PEGAR LISTA FINAL

Monedas Permitidas

HKD

SAR

HUF

