



Técnicas Básicas Excel

(II)

*Trucos, opciones y personalización de Excel.
Introducción al VBA*

Jose Ignacio González Gómez
Departamento de Economía Financiera y Contabilidad - Universidad de La Laguna
www.jggomez.eu

INDICE

1	Preparando la apertura de nuestro libro de trabajo personalizado	3
1.1	<i>Poniendo nuestro sello de autoría al cargar nuestros libros Excel</i>	3
1.2	<i>Apertura del libro en una hoja concreta.....</i>	3
1.3	<i>Ocultado la fichas de hojas del libro de trabajo.</i>	3
2	Configuración de una hoja de inicio o de presentación.....	4
2.1	<i>Introducción</i>	4
2.2	<i>Quitando la visualización de las referencias A, B..1,2.. y las líneas de división</i>	4
2.3	<i>Insertando botones controles en la hoja de inicio. Caso imagen con hipervínculo ...</i>	4
2.4	<i>Proteger la hoja de inicio.....</i>	5
2.5	<i>Ocultar y recuperar filas y columnas de forma sencilla</i>	6
2.6	<i>Limitar el desplazamiento a lo largo de una hoja.....</i>	6
3	Ocultar hojas para que no puedan ser mostradas	9
3.1	<i>Objetivo</i>	9
3.2	<i>Versión Simple</i>	9
3.3	<i>Versión Avanzada.....</i>	9
4	Introducción a los Formularios. Conceptos y Tipos de UsersForms	11
4.1	<i>Objetivos Básicos de los Formularios en Excel.....</i>	11
4.1.1	<i>Crear un Userform.....</i>	11
4.1.2	<i>Mostrar/Ocultar y Cerrar un UserForm</i>	12
4.2	<i>Mostrar y cerrar un formulario de bienvenida o de menú general.....</i>	12
5	Proteger y desproteger varias hojas a la vez (todo el libro) con una sola macro para evitar cambios no deseados	13
5.1	<i>Objetivo y planteamiento del problema.....</i>	13
5.2	<i>Macro propuesta para proteger y desproteger a la vez todas las hojas de un libro</i>	13
6	Creando un menú avanzado para nuestros desarrollos Excel.....	14
6.1	<i>Creación del formulario Menu y personalización.....</i>	14

6.2	<i>Insertando y programando botones genéricos en el formulario Menú</i>	15
6.2.1	<i>Introducción</i>	15
6.2.2	<i>Creación de botones de navegación por las distintas hojas de cálculo</i>	15
6.2.3	<i>Botones para mostrar y ocultar las pestañas de las hojas de cálculo</i>	16
6.2.4	<i>Incluir un mensaje de orientación. MsgBox</i>	16
7	<i>Proteger nuestro código VBA.....</i>	17

1 Preparando la apertura de nuestro libro de trabajo personalizado

1.1 Poniendo nuestro sello de autoría al cargar nuestros libros Excel

Queremos que cada vez que se abra nuestro libro de trabajo tengamos presente la autoría del mismo con un sello personalizado tal y como se muestra en la Ilustración 1

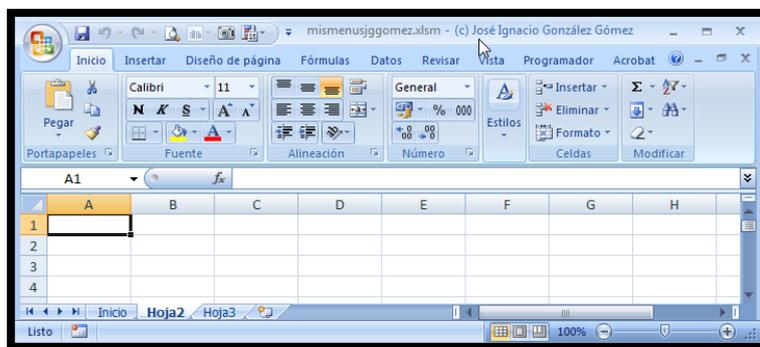


Ilustración 1

Para ello asociamos al elemento libro el evento open y la propiedad Application.Caption. Caption, tal y como se muestra en la Ilustración 2.

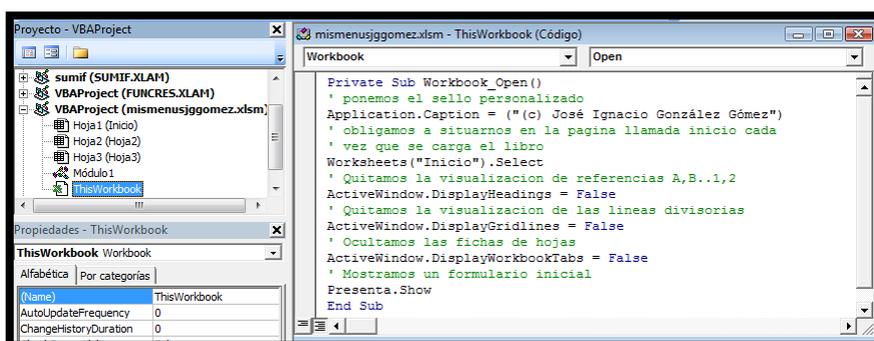


Ilustración 2

En concreto la instrucción es:

Application.Caption = "(c) José Ignacio González Gómez")

1.2 Apertura del libro en una hoja concreta

A continuación queremos que cada vez que se abra el libro, por defecto comience en la hoja que hemos denominado como Inicio, para ello y siguiendo los pasos anteriores es decir asociado al evento Open del libro de trabajo la instrucción correspondiente seria (ver Ilustración 2):

Worksheets("Inicio").Select

1.3 Ocultado la fichas de hojas del libro de trabajo.

Nos puede interesar que deseamos ocultar todas las fichas de las hojas en la apertura de un libro de trabajo personalizado, para ello la instrucción a usar es (ver Ilustración 2):

ActiveWindow.DisplayWorkbookTabs = False

2 Configuración de una hoja de inicio o de presentación

2.1 Introducción

Nuestro objetivo es mostrar o tener a disposición una de hoja de inicio de nuestro libro de trabajo similar a la presentada en la siguiente ilustración.

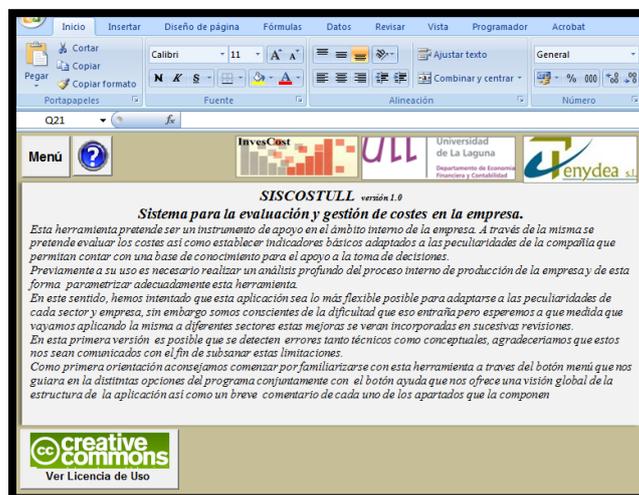


Ilustración 3

2.2 Quitando la visualización de las referencias A, B..1,2.. y las líneas de división

Para darle un aspecto más profesional a la pantalla de presentación, nos interesa quitarle las referencias de filas y columnas tradicionales de Excel para ello tenemos que desactivar los títulos de la hoja como se muestra en el menú Vista además de las líneas de cuadrícula.

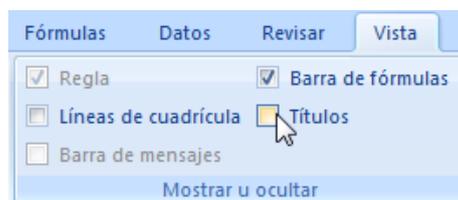


Ilustración 4

Si quisiéramos realizarlo a nivel de programación la instrucción sería:

ActiveWindow.DisplayGridlines = False (para el caso de las líneas)
ActiveWindow.DisplayHeadings = False (para el caso de los títulos)

Veamos las instrucciones correspondientes en la Ilustración 2, donde abierto el libro de trabajo seleccionamos la hoja inicio y configuramos la misma para desactivar las referencias y las líneas de división.

2.3 Insertando botones controles en la hoja de inicio. Caso imagen con hipervínculo

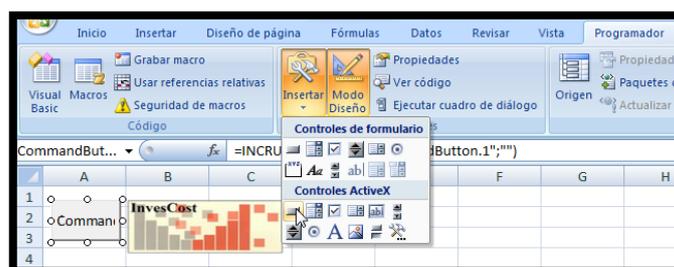


Ilustración 5

Para insertar botones o controles en nuestra futura hoja de inicio accedemos a la pestaña programador y seleccionamos algunos de los controles ActiveX disponibles como es el caso del botón que nos permitirá insertar el mismo en la hoja de cálculo necesaria.

Es necesario tener activado el modo de diseño. Si posteriormente queremos ver el resultado salimos del modo de diseño y podremos observar cómo nos quedaría finalmente.

Cada uno de estos botones e imágenes pueden ser programados, así por ejemplo en modo diseño si hacemos doble click sobre la imagen accederemos a las opciones de programación asociadas algún evento relacionado con el citado botón o imagen tal y como podemos ver en la siguiente ilustración.

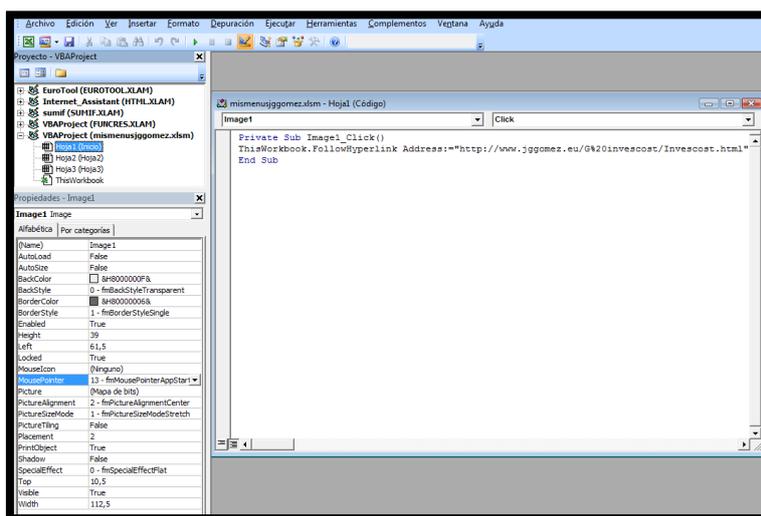


Ilustración 6

En este caso el código es sencillo:

```
Private Sub Image1_Click()
```

```
    ThisWorkbook.FollowHyperlink Address:="http://www.jggomez.eu/G%20invescost/Invescost.html"
```

```
End Sub
```

Su significado es, con el elemento llamado Image1 y asociado al evento o acción click seguir el siguiente hipervínculo.

Aquí podremos poner tantos controles como imágenes deseadas tal y como se muestra en la Ilustración 3.

2.4 Proteger la hoja de inicio

En este caso, contamos con una primera hoja de presentación que queremos proteger para que no sea modificada ni alternada y no se pueda navegar por ella.

Partimos de la siguiente hoja

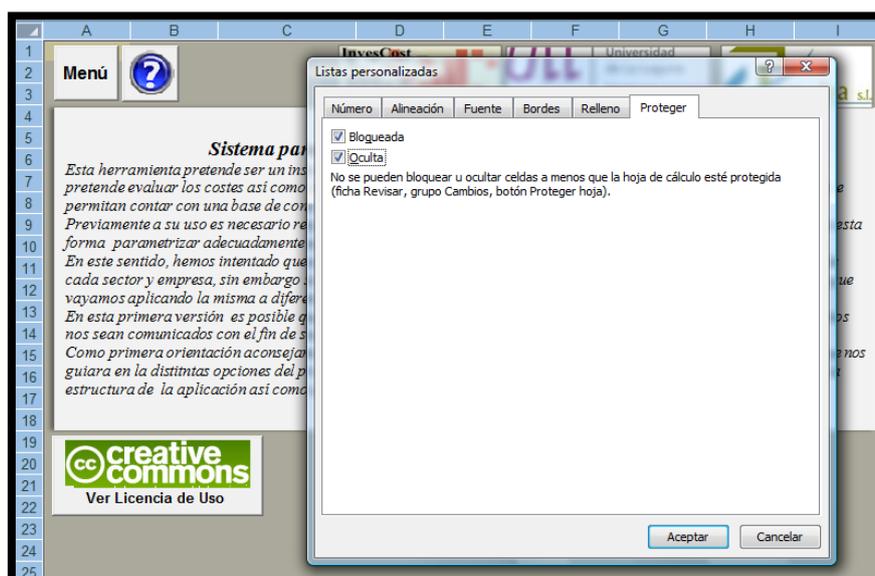


Ilustración 7

Seleccionamos todas las celdas y las bloqueamos y ocultamos. A continuación protegemos el libro, así solo se podrán usar los botones de navegación habilitados.

2.5 Ocultar y recuperar filas y columnas de forma sencilla.

Ocultar / recuperar filas: Como ya sabemos, en Excel es posible tener filas o columnas ocultas, lo que evita tener “a la vista” celdas con resultados intermedios, o que no nos interesa ver. La forma más simple de ocultar una fila es presionar “CTRL” y “9” a la vez. La fila en la que está la celda actual habrá “desaparecido”. Si la queremos recuperar, simplemente la combinación de teclas “CTRL”, “Mayús.” y “8” la hará aparecer nuevamente.

Ocultar / recuperar columnas: Al igual que en el truco precedente, para ocultar la fila de la celda actual basta con presionar “CTRL” y “0” (cero). Cuando deseemos que la columna se haga visible nuevamente, deberemos presionar en simultáneo tres teclas: “CTRL”, “Mayús.” y “9”.

2.6 Limitar el desplazamiento a lo largo de una hoja

En la mayoría de los casos, las hojas sólo utilizan un pequeño porcentaje de todas las celdas disponibles.

En otros casos puede interesar limitar el área de desplazamiento de tal forma que sólo se pueda visualizar un rango determinado de filas y columnas. Piense que, de esta manera, se podría impedir el acceso a ciertas zonas de la hoja de cálculo o sencillamente hacer más cómoda la tarea de seleccionar celdas.

Existe la posibilidad de establecer el área por el que se puede desplazar el usuario de forma que sólo pueda ver los datos que desee. Luego, puede colocar datos que no deben ser vistos fuera de esa área, es decir si tenemos datos que no deseamos visualizar, puede ser útil limitar el área visible de la hoja de cálculo sólo al rango que actualmente tiene datos.

Esta propuesta de método avanzado es de [JLD Excel](#), suponiendo que el área a restringir sea el rango B3:E10 de la Hoja1, procedemos así, abrimos el editor VB:

1. En la ventanilla VBAProject del archivo, seleccionamos con un clic el objeto ThisWorkbook.

2. Elegimos la opción Workbook. Inmediatamente debajo de Private Sub Workbook_Open(), escribimos Sheets(1).ScrollArea = "B3:E10"

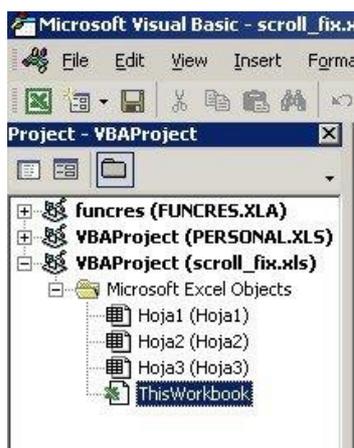


Ilustración 8

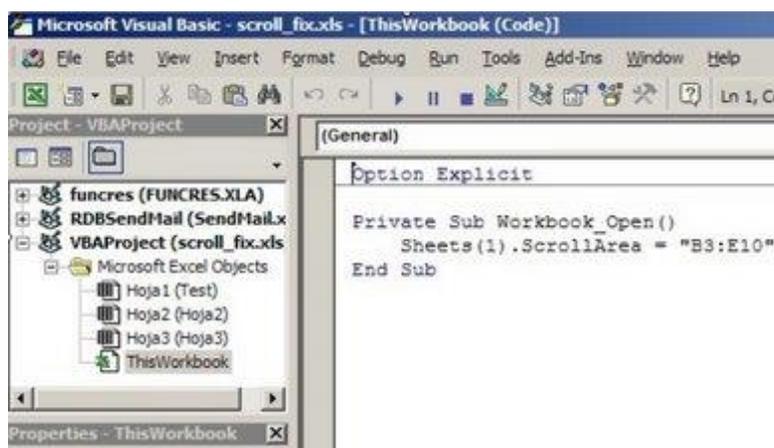


Ilustración 9

3. Volvemos al cuaderno Excel y lo guardamos.

A partir de este momento, cada vez que abrimos el cuaderno (Workbook_Open) Excel define el área de trabajo de la Hoja1 con el rango B3:E10.

Hacer notar que en el caso de que la Hoja1 o Sheet1 se llamara Pepe el código sería el mismo. Es decir daría el mismo resultado poner

- Sheets(1).ScrollArea
- Sheets("Pepe").ScrollArea

Excel permite que nos refiramos a la hojas por su nombre ("Pepe") o por su número de índice Sheets(1).

En caso de que tuviéramos varias hojas con el acceso limitado en su desplazamiento como el visto en el caso anterior el procedimiento a seguir sería repetirse varias veces la misma subrutina, de manera que no solo la primera, sino varias páginas (o todas) pueden tener un "ScrollArea" delimitado y diferente entre si en un libro, por ejemplo incluir en la subrutina la instrucción Sheets(2)."ScrollArea=A1:Z10" para afectar una segunda hoja de cálculo.

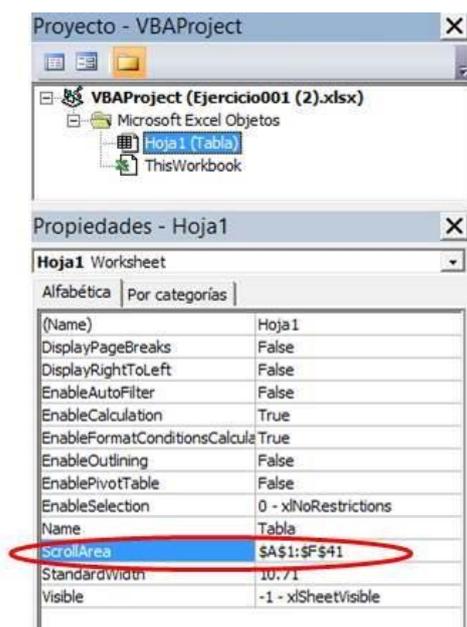


Ilustración 10

Otra forma también de hacerlo es a través del Explorador de Proyecto seleccionamos la hoja donde se encuentra en rango; y luego dentro de la Ventana de Propiedades, en la propiedad ScrollArea escribiré el rango A1:F41

Otra opción es ejecutar el siguiente código cada vez que la hoja se active tal y como mostramos en las siguientes ilustraciones

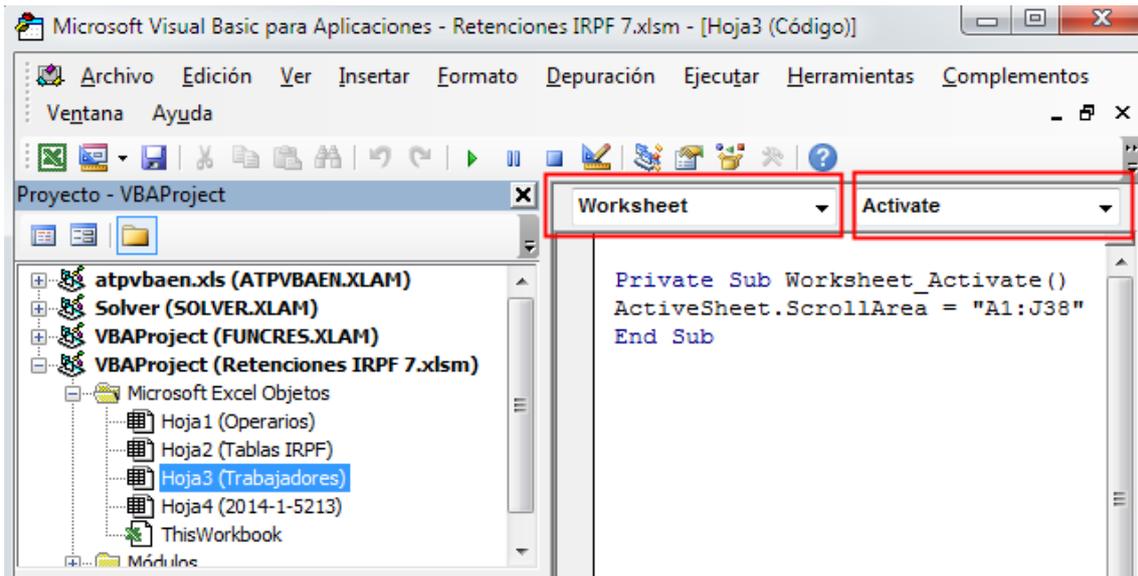


Ilustración 11

Es decir al objeto seleccionado Worksheet (en nuestro caso Hoja 3) establecemos al evento Activate

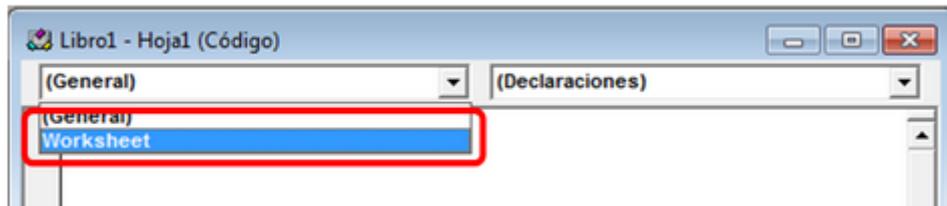


Ilustración 12

El siguiente código: `ActiveSheet.ScrollArea = "A1:J38"`

Esto viene a significar que al activarse la hoja 3 el área de de movimiento estará limitada por el rango A1: J38

3 Ocultar hojas para que no puedan ser mostradas

3.1 Objetivo

Este es el caso típico de salvaguardar el origen de los datos, por ejemplo las bases de datos en Excel, es decir a veces desearía tener un lugar donde colocar información que no pueda ser leída o modificada por los usuarios. Puede construir un lugar secreto dentro del libro/ un lugar donde almacenar información, fórmulas y otros recursos que se utilizan en las hojas pero que no desea que se vean.

Una práctica muy útil cuando se configura un nuevo libro de Excel es reservar una hoja para almacenar información que los usuarios no necesitan ver: cálculos de fórmulas, validación de datos, listas, variables de interés y valores especiales, datos privados, etc.

Para ello presentamos dos opciones:

- Versión Simple
- Versión Avanzada

3.2 Versión Simple

Seleccionando la hoja que queremos ocultar y con el botón derecho activamos esta posibilidad tal y como se muestra en la Ilustración 13, pero evidente el usuario si conoce suficiente Excel podrá activarla manualmente.

En algunos casos si queremos que esta prohibición sea más fuerte tenemos que recurrir a las propiedades del objeto mediante código VBA, es la opción que presentamos a continuación como avanzada.

3.3 Versión Avanzada

Para este segundo caso es necesario acceder a la pestaña Programador y a la opción VBA, de esta forma se abre el entorno de desarrollo VBA de Excel y seleccionamos el objeto que nos interesa y cambiamos la propiedad Visible del mismo, tal y como se muestra en la Ilustración 15 y seleccionamos la opción para ocultar la hoja deseada.

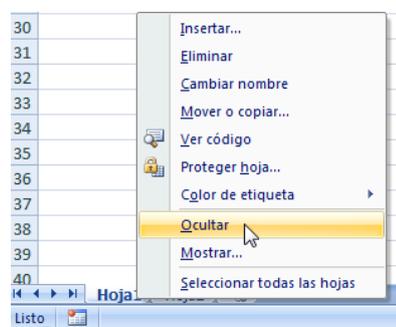


Ilustración 13

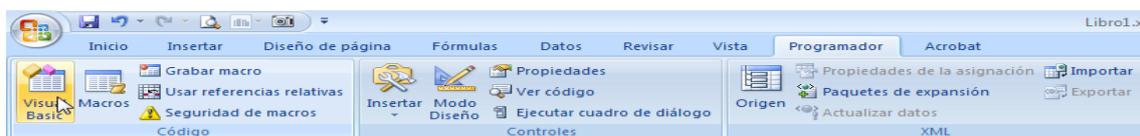


Ilustración 14

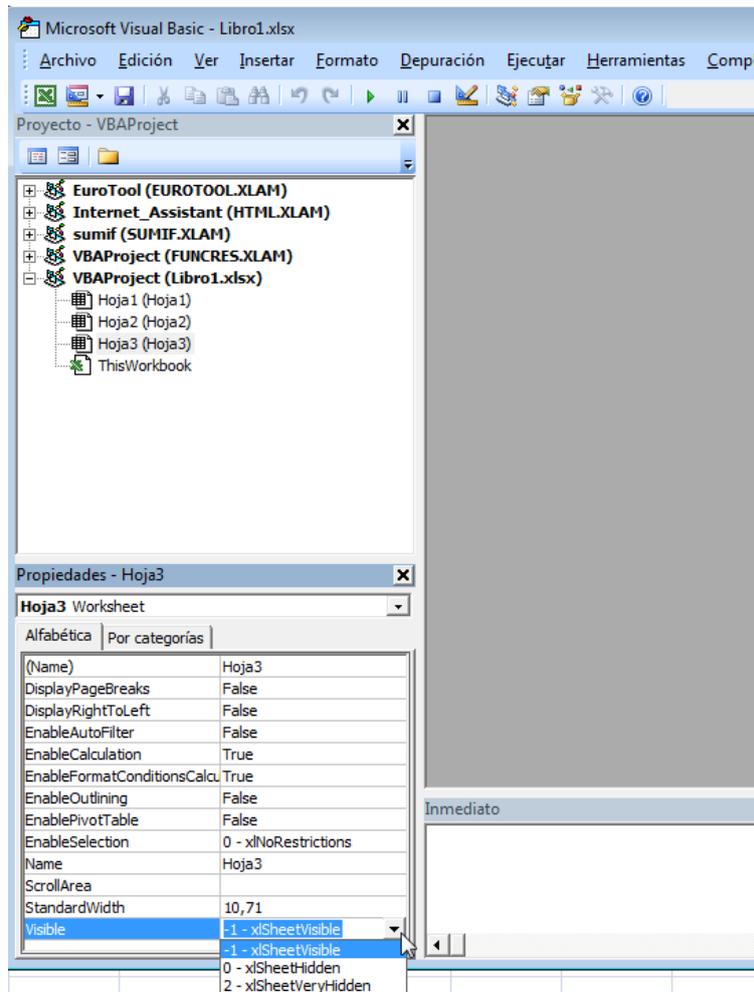


Ilustración 15

A partir de ahora, la hoja ya no estará visible desde la interfaz de Excel e incluso tampoco podrá mostrarse a través de la opción Formato>Hoja>Mostrar.

Otra forma a través de código es decir programando la ocultación de una hoja con VBA es:

```
Sheets("Transp1").Visible = Excel.XlSheetVisibility.xlSheetVeryHidden
```

En este caso la hoja "Transp1" quedara oculta sin posibilidad de modificación

```
Workbook Open
Private Sub Workbook_Open()
Application.Caption = ("J. Juan Cáceres - J. Ignacio González y Juan Nuez")
' ocultamos hojas que no nos interesa
Sheets("Transp1").Visible = Excel.XlSheetVisibility.xlSheetVeryHidden

' Le decimos donde tiene que ir cuando se abra el libro
Sheets("Estructura de Costes 1").Select
Range("D39").Select
End Sub
```

Ilustración 16

4 Introducción a los Formularios. Conceptos y Tipos de UsersForms

4.1 Objetivos Básicos de los Formularios en Excel.

Los cuadros de dialogo personalizados (UserForms) tienen como objetivos básicos:

- Introducción de datos por parte del usuario
- Mostrar mensajes y conseguir respuestas sencillas
- Seleccionar un archivo o cuadro de dialogo
- Mostrar cuadros de diálogos integrados en Excel.

Debemos señalar que los Formularios de Excel conocidos como USERFORM o Formularios de Usuario tienen limitaciones y no poseen las características de los formularios de VBA que poseen características más avanzadas.

Los formularios Excel tienen una gran variedad de controles, los cuales tienen una variedad de propiedades, funciones y eventos.

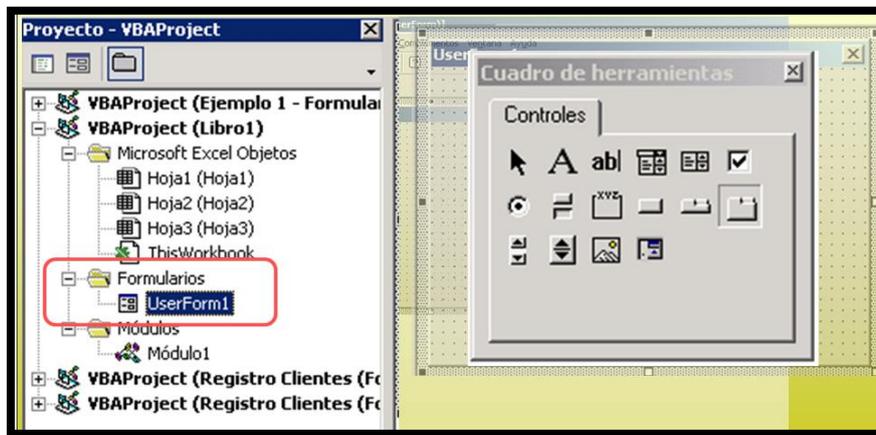


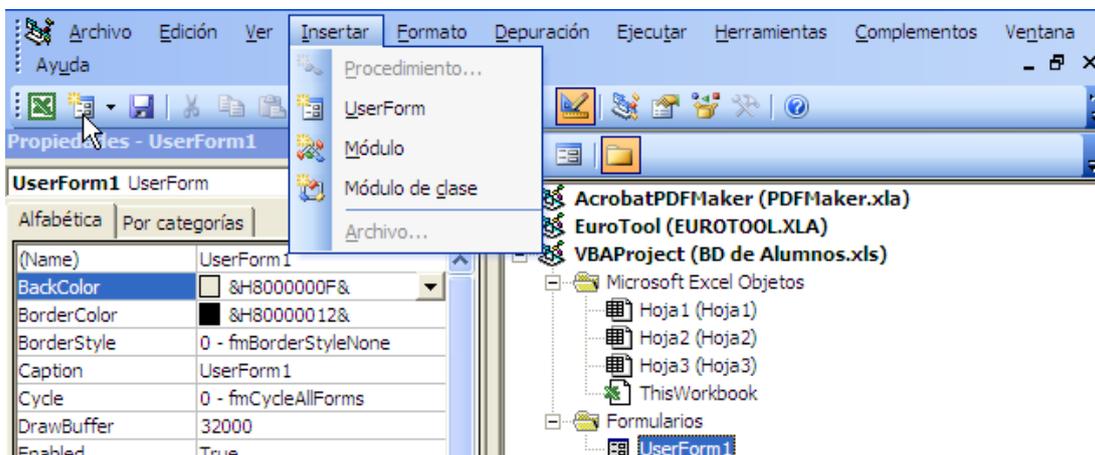
Ilustración 17

4.1.1 Crear un Userform

1

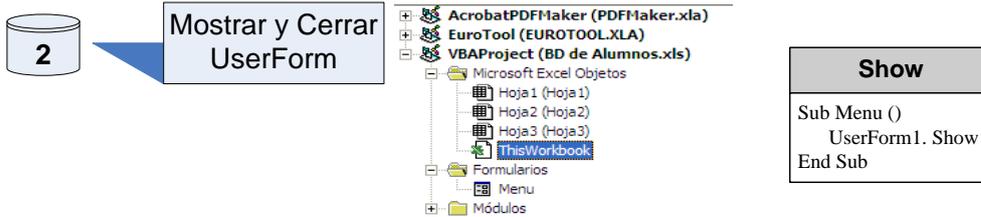
Insertar un UserForms

Seleccionado el proyecto donde queremos crear el UserForm, escogemos Insertar>UserForm o bien el icono de acceso directo disponible al efecto



Por defecto los UserForm tienen nombres como UserForm1, UserForm2, etc.. Para poderlo identificar más fácilmente podemos cambiarle el nombre en la ventana de propiedades del UserForm, en concreto en la propiedad Name

4.1.2 Mostrar/Ocultar y Cerrar un UserForm



Es necesario crear un procedimiento que debe situarse en un módulo estándar y no en el módulo de código del UserForm.

```

(General) | Menugral
-----
Option Explicit

Private Sub Workbook_Open()
Application.Caption = "(C) Jose Ignacio González Gómez"
MsgBox Prompt:=" Sistema para el control y seguimiento de las notas de los alumnos", I
Menugral
End Sub

Sub Menugral()
Menu.Show
End Sub
    
```

Para ocultar un formulario utilizamos el comando `Hide` `Userform1.Hide`
 Para cerrar un formulario utilizamos el comando `Unload` `Unload Userform1`

- Por tanto para abrir un formulario la sentencia sería: **Menu.Show**
- Para cerrar un formulario concreto la sentencia sería: **Unload.Menu**
- Para ocultar un formulario concreto la sentencia sería: **Menu.Hide**

4.2 Mostrar y cerrar un formulario de bienvenida o de menú general

En algunas ocasiones nos puede interesar que en el momento en que abrimos nuestro libro personalizado mostrar un formulario de bienvenida (supongamos que este se llama Presenta) o de menú general, para ello debemos asociar la siguiente instrucción al evento correspondiente, tal y como se muestra en la Ilustración 2:

`Presenta.Show`

5 Proteger y desproteger varias hojas a la vez (todo el libro) con una sola macro para evitar cambios no deseados

Fuente: <http://www.todoexcel.com/foro-excel/otras-consultas/desproteger-vez-varias-hojas-libro-t11449.html>

5.1 Objetivo y planteamiento del problema

Contamos con un archivo (libro) con 27 hojas (pestañas). A todas les hemos puesto la misma contraseña para que los usuarios no borren fórmulas por error. Necesitamos hacer cambios de vez en cuando en todas las hojas y ello nos obliga a desbloquear las 27 una a una y volverlas a bloquear una a una, lo que es un verdadero engorro y pérdida de tiempo.

Buscamos una forma para poder bloquear y desbloquear todas las hojas a la vez evitando así tener que escribir la clave 27 veces para desbloquear y 54 más para volver a bloquear las hojas (81 veces)..

5.2 Macro propuesta para proteger y desproteger a la vez todas las hojas de un libro

Sub DesprotegerHojas()

```
Dim i As Integer, HojaActual As Integer
Application.ScreenUpdating = False
HojaActual = ActiveSheet.Index 'es para después restaurar
For i = 1 To Sheets.Count
    Sheets(i).Select
    ActiveSheet.Unprotect "1234"
Next i
Sheets(HojaActual).Select
Application.ScreenUpdating = True
```

End Sub

Sub ProtegerHojas()

```
Dim i As Integer, HojaActual As Integer
Application.ScreenUpdating = False
HojaActual = ActiveSheet.Index 'es para después restaurar
For i = 1 To Sheets.Count
    Sheets(i).Select
    ActiveSheet.Protect "1234"
Next i
Sheets(HojaActual).Select
Application.ScreenUpdating = True
```

End Sub

6 Creando un menú avanzado para nuestros desarrollos Excel

6.1 Creación del formulario Menu y personalización

El objetivo de este apartado es crear un menú personalizado a través de un formulario o userform que nos permita orientar al usuario a través de las opciones básicas de nuestro libro de trabajo personalizado, tal y como se muestra en la Ilustración 18.

En cada una de las hojas que componen nuestro libro de trabajo insertaremos un botón que nos de acceso al citado formulario o menú.

Para ello y en primer lugar, tal y como hemos visto en el apartado 4, crearemos un formulario que llamaremos Menu y lo personalizaremos.

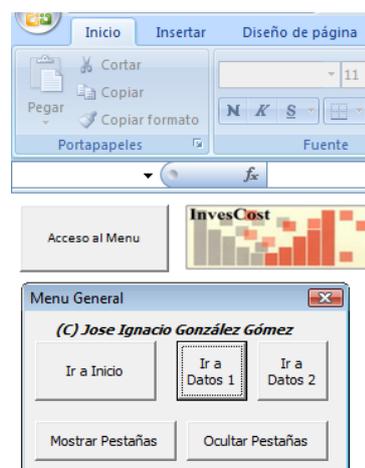


Ilustración 18

Atendemos a las personalización de este formulario, en especial a las propiedades como: Name y Caption.

6.2 Insertando y programando botones genéricos en el formulario Menú

6.2.1 Introducción

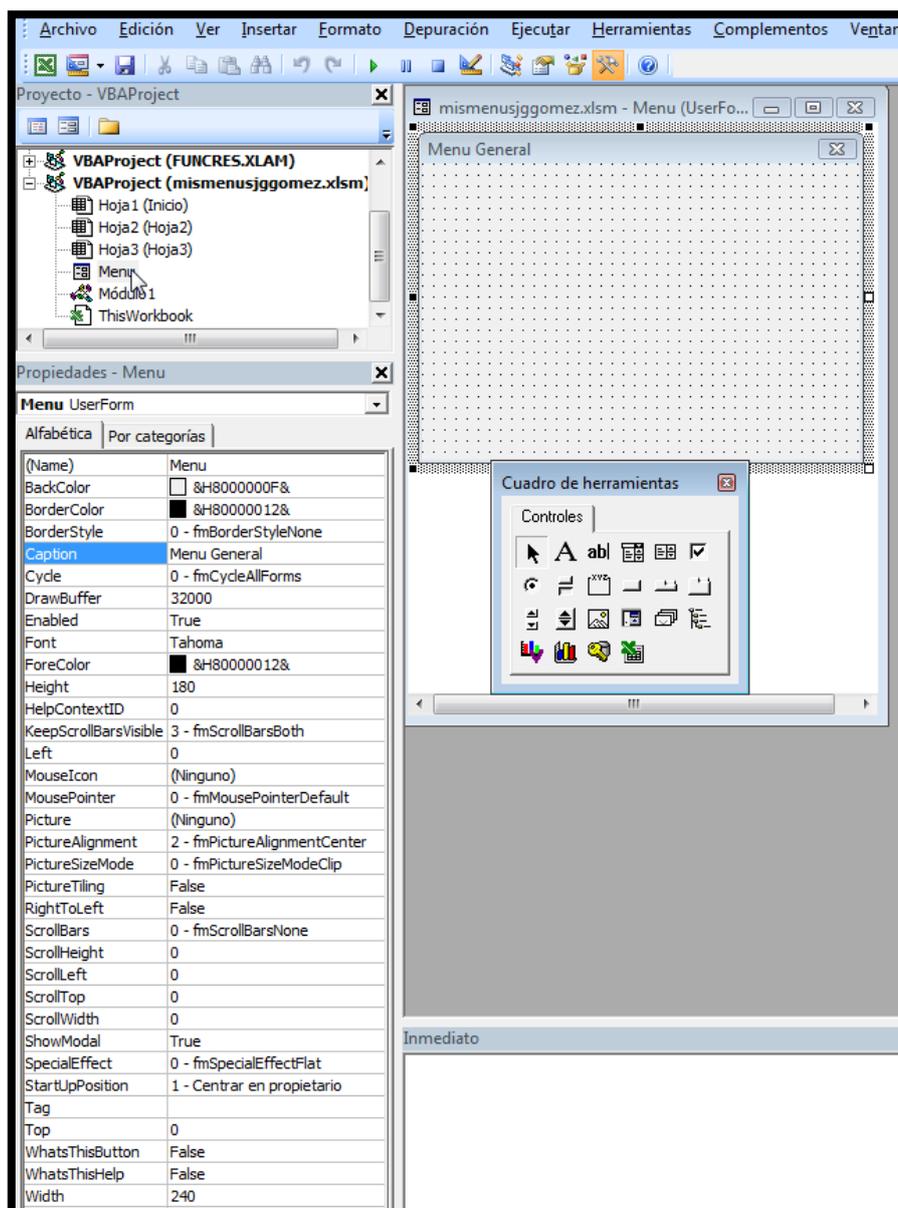


Ilustración 19

A través del cuadro de herramientas disponible en la cinta de programación podemos insertar y programar los diversos controles disponibles tal como se muestra en la Ilustración 19.

El control que más utilizaremos en este apartado es el botón de comando también conocido como `CommandButton`.

6.2.2 Creación de botones de navegación por las distintas hojas de cálculo

Tal y como vemos en la Ilustración 18 hemos creado tres botones que nos permite navegar a lo largo de las distintas hojas, estos botones se han programado de la siguiente forma:

```
Private Sub boton0_Click()  
Worksheets("Inicio").Activate  
Range("C5").CurrentRegion.Select  
End Sub  
  
Private Sub boton1_Click()  
Worksheets("Datos1").Activate  
Range("C2").CurrentRegion.Select  
End Sub  
  
Private Sub boton2_Click()  
Worksheets("Datos2").Activate  
Range("ZonaA").Select  
End Sub
```

Ilustración 20

Así podemos ver como el primer botón que hemos denominado como "boton0" responde al evento click de la siguiente forma:

- Pone en activo la hoja denominada inicio.
- Y nos sitúa en la celda C2

De igual forma, por ejemplo tenemos el boton2 en este caso hace referencia a un rango definido previamente como Zona A.

6.2.3 Botones para mostrar y ocultar las pestañas de las hojas de cálculo

Retomando la Ilustración 18, exponemos a continuación el código sencillo que nos permite cerrar y activar las pestañas, en concreto son los botones que hemos denominado como boton3 y boton4 asociados al evento o acción click del ratón.

```
Private Sub boton3_Click()  
' Mostramos las fichas de hojas  
ActiveWindow.DisplayWorkbookTabs = True  
End Sub  
  
Private Sub boton4_Click()  
' Ocultamos las fichas de hojas  
ActiveWindow.DisplayWorkbookTabs = False  
End Sub
```

Ilustración 21

6.2.4 Incluir un mensaje de orientación. MsgBox

Si en cualquier momento quisiéramos mostrar un mensaje de advertencia o información al pulsar un botón tendríamos que incluir una sentencia como la que mostramos a continuación:

MsgBox ("Por favor, ajuste manualmente el área de impresión que dese tomando como referencia lo marcado por defecto")

7 Proteger nuestro código VBA

Para proteger nuestro código VBA de Excel accedemos a la Pestaña Herramientas y a la opción Propiedades de VBAProject esto nos da acceso a poder introducir la contraseña para modificar e incluso impedir la visualización del código desarrollado, ver la Ilustración 22.

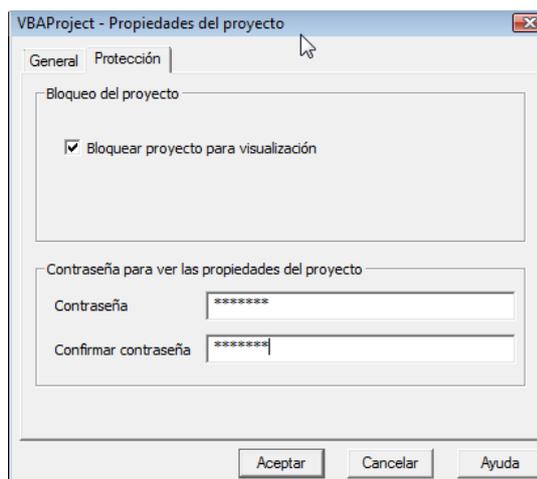


Ilustración 22