



Componentes y controles en Excel (I)

Introducción a los controles y formularios. Controles ActiveX, Spreadsheet y ChartSpace, etc

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

INDICE

1	Introducción a la programación de controles	2
1.1	Cuadro de control (dashboard) sencillo con Excel	2
1.1.1	Nota I.....	2
1.1.2	Nota II.....	6
2	Simulador presupuestario.xlsm: trabajando con controles ActiveX, Microsoft Office Web Components, ChartSpace, Spreadsheet, Spin, Conectar y bloquear textbox, enlazar textbox, etc.....	9
2.1	Planteamiento del problema.....	9
2.2	El formulario y análisis de los principales componentes	10
2.2.1	Primera aproximación al formulario "Facturas".....	10
2.2.2	Análisis de los textbox del formulario. Características principales. Formato del cuadro de texto.	11
2.2.3	Bloqueando los textbox.....	12
2.2.4	Control SpinButton (botón de numero) asociado a Text Box.	14
2.3	Principales eventos del formulario facturas y módulos del proyecto.....	16
2.3.1	Evento al inicializar el formulario.....	16
2.3.2	Botón inicializar/abrir el formulario. Módulo 1	17
2.3.3	Eventos especiales asociados a los textbox. Modulo Actualiza Datos.....	17
2.3.4	Llamada a los módulos desde los TextBbox independientes.	18
2.4	Controles ActiveX, Spreadsheet y ChartSpace.....	19
2.4.1	Introducción	19
1.1.1	Análisis del componente Spreadsheet.	20
2.4.2	Análisis del Módulo Webcomponentes. Caso Spreadsheet.....	20
2.4.3	Análisis del Módulo Webcomponentes. Caso ChartSpace	21
3	Bibliografía, ficheros y enlaces.....	22

1 Introducción a la programación de controles

1.1 Cuadro de control (dashboard) sencillo con Excel

1.1.1 Nota I

Ya habíamos tocado el tema del uso de controles en hojas. En esta nota mostraré como crear un informe dinámico o cuadro de control (dashboard) sencillo.

En esta primera nota veremos cómo hacerlo con los controles de la barra de Formulario. Estos controles tienen la ventaja de ser sencillos de usar. La desventaja es que son poco flexibles, como veremos más adelante.

Supongamos que tenemos una lista de datos de ventas (plan y realizado) de una empresa para desde enero del 2007 hasta octubre del 2009. Nuestro objetivo es crear este cuadro de control (dashboard):

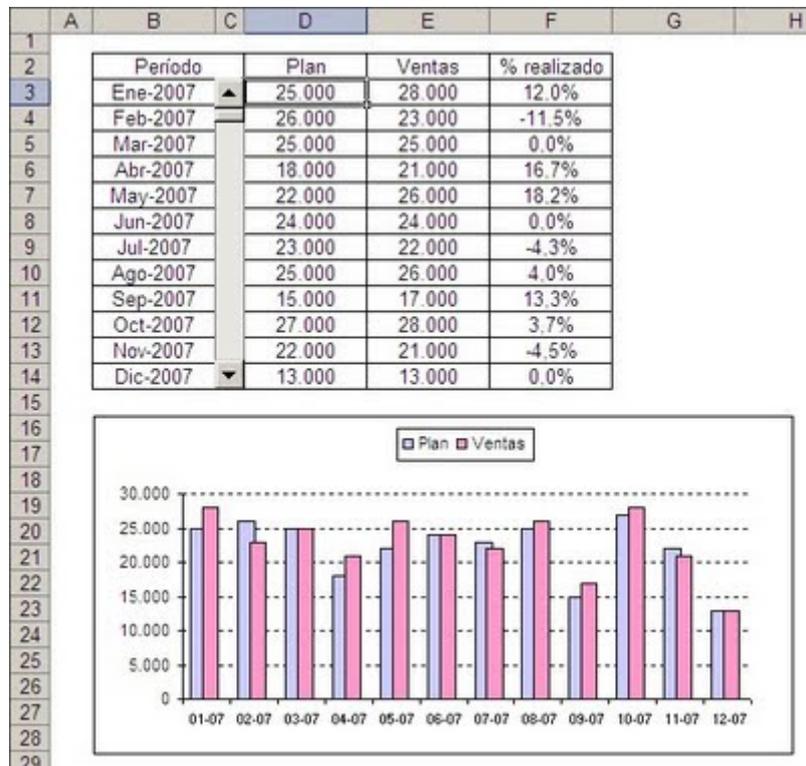


Ilustración 1

Hemos puesto una barra de desplazamiento que nos permite cambiar las fechas del informe con un clic en las flechas de la barra:

Nuestro modelo tiene tres hojas:

- Informe – donde ponemos el cuadro de control y el gráfico (ver Ilustración 1)
- Base de datos que contiene los datos de ventas. Como pueden ver, las fechas están formateadas como mm-aaaa (o mm-yyyy)
- Control – ligaremos el control (la barra de desplazamiento) a la celda B3 de esta hoja.

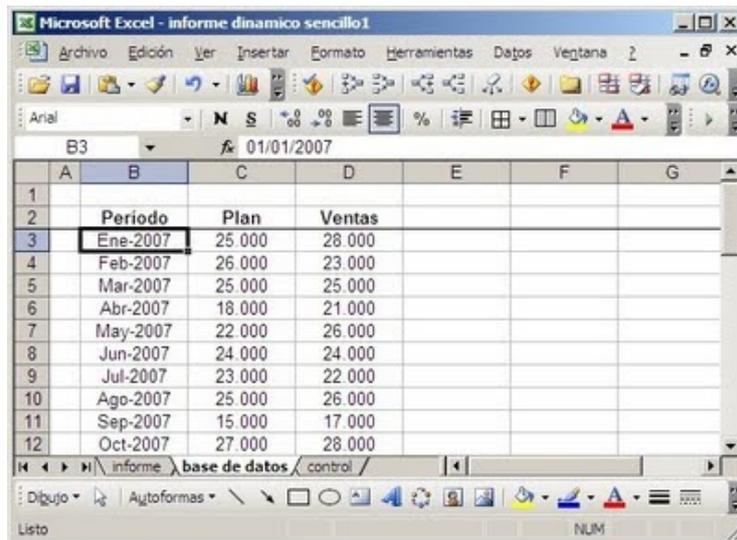


Ilustración 2

También hemos definidos este nombres

inicio=control!\$B\$3

Ahora creamos el cuadro de control en la hoja “informe”.



Ilustración 3

En el rango C3:C14 ponemos la barra de desplazamiento. Ese es el motivo por el cual la columna C es más angosta que las restantes.

Activamos la barra de herramientas de Formularios y seleccionamos la barra de desplazamiento



Ilustración 4

La arrastramos y hacemos que coincida con el rango C3:C14. Luego abrimos el menú de formato del control

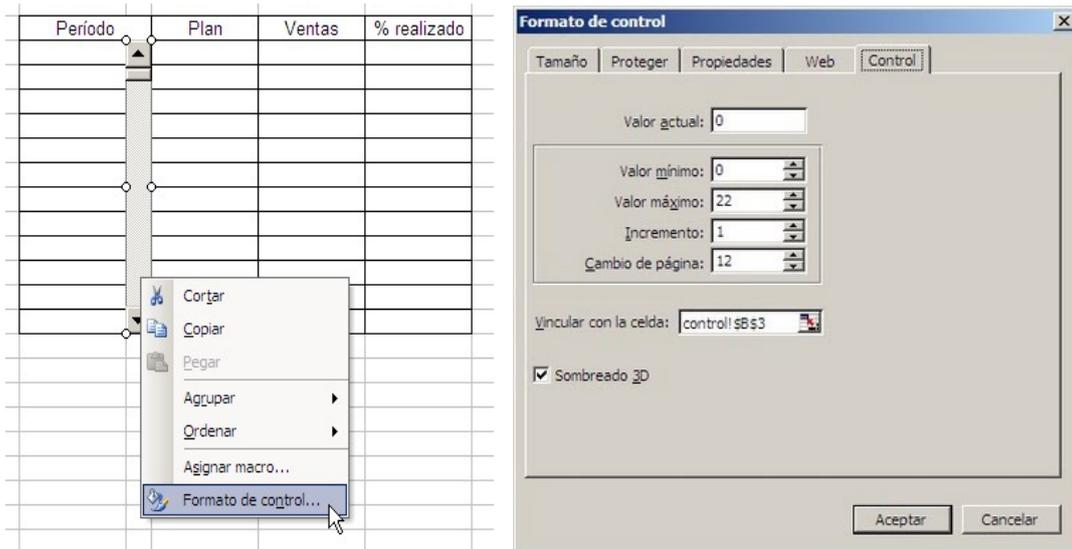


Ilustración 5

Ponemos estos valores:

- Valor actual = 0
- Valor Mínimo = 0
- Valor Máximo = 22
- Incremento = 1
- Cambio de Página = 12

Vincular con celda - control!\$B\$3 (o poner el nombre "inicio")

Seguimos. En la celda B3 ponemos esta fórmula

=DESREF('base de datos'!\$B\$3;inicio+FILA()-3;0)



Ilustración 6

y la copiamos hasta la celda B14. En la celda D3 ponemos la misma fórmula pero referenciando la columna C en la hoja "base de datos"

=DESREF('base de datos'!\$C\$3;inicio+FILA()-3;0)

y lo mismo en la celda E3, con referencia a la columna D en "base de datos"

D3		=DESREF('base de datos'!\$C\$3:inicio+FILA()-3:0)				
A	B	C	D	E	F	G
1						
2	Periodo		Plan	Ventas	% realizado	
3	Ene-2007		25 000	28 000	12.0%	
4	Feb-2007					
5	Mar-2007					
6	Abr-2007					
7	May-2007					
8	Jun-2007					
9	Jul-2007					
10	Ago-2007					
11	Sep-2007					
12	Oct-2007					
13	Nov-2007					
14	Dic-2007					
15						

Ilustración 7

En la celda F3 ponemos la fórmula =E3/D3-1

Copiamos todas las fórmulas al resto de la tabla. Todo lo que nos queda por hacer es crear el gráfico

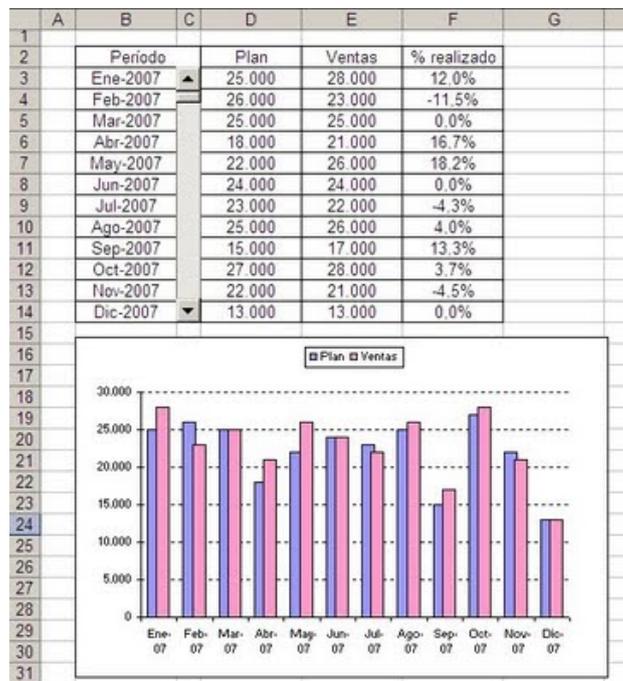


Ilustración 8

Este gráfico es absolutamente dinámico e irá cambiando a medida que cambien los datos en la tabla.

Este modelo tiene un serio inconveniente. Cada vez que agregamos o quitamos filas de la base de datos tendremos que corregir manualmente las definiciones de la barra de desplazamiento. Por ejemplo, si agregamos tres meses más y no corregimos la definición del valor máximo de la barra no veremos los datos que acabamos de agregar.

Sólo después de corregir la definición del valor a 25 veremos los nuevos datos.

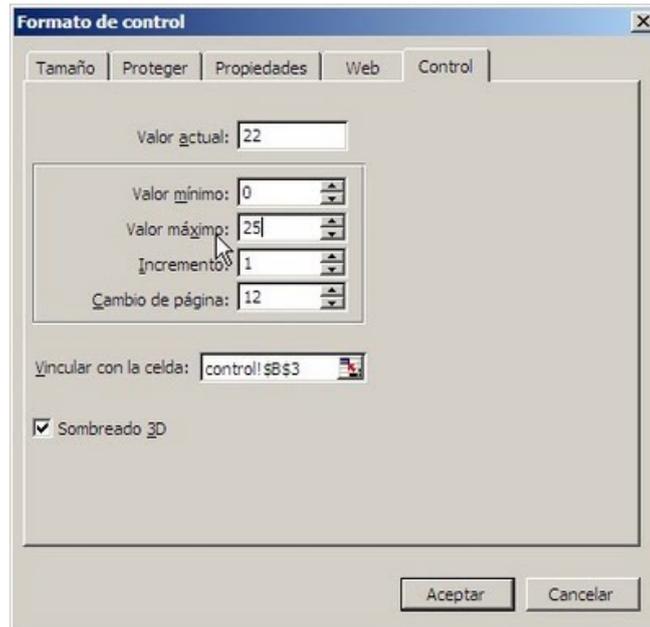


Ilustración 9

Podemos hacer que el modelo se actualice automáticamente usando los controles ActiveX en lugar de los de la barra de formularios. Este será el tema de la próxima nota.

1.1.2 Nota II

En la nota anterior vimos cómo construir un dashboard sencillo en Excel usando un control de la barra de Formularios y la función DESREF.

Vimos también que el control de la barra de Formularios tiene un serio inconveniente: no podemos establecer los parámetros dinámicamente. En nuestro ejemplo vimos que al agregar líneas a la base de datos tenemos que modificar manualmente la definición de valor máximo del control.

Para crear un modelo que se adapte automáticamente a los cambios en la base de datos tendremos que usar otro tipo de controles, los controles ActiveX.

Empezamos por reemplazar la barra de desplazamiento que pusimos ayer por el mismo control pero de la barra de herramienta Cuadro de Controles (ActiveX).

Para eso activamos la barra (Herramientas-Personalizar-Barra de Herramientas o mejor haciendo un clic con el botón derecho del mouse sobre el área de la barras de herramientas)

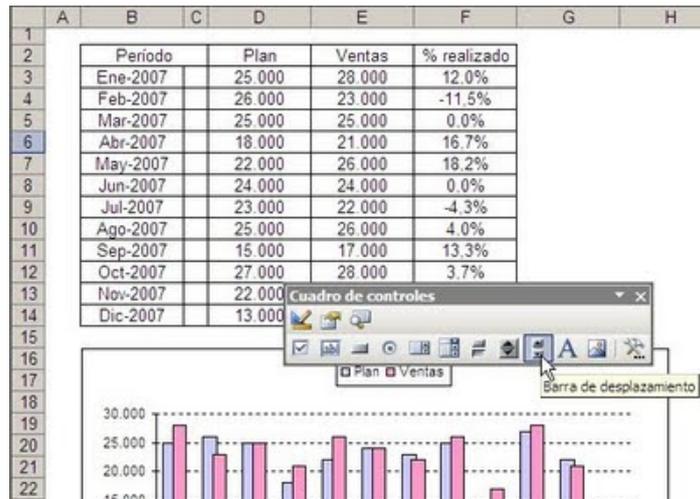


Ilustración 10

Dibujamos el control en la posición deseada (tip: mantenemos apretada la tecla ALT para que los bordes del control encuadren con los bordes de las celdas). Al dibujar el control Excel activa automáticamente el modo de Diseño (veremos que el icono con la escuadra y la regla aparece “apretado”). Estando en modo Diseño apretamos el icono Propiedades, establecemos estas propiedades:

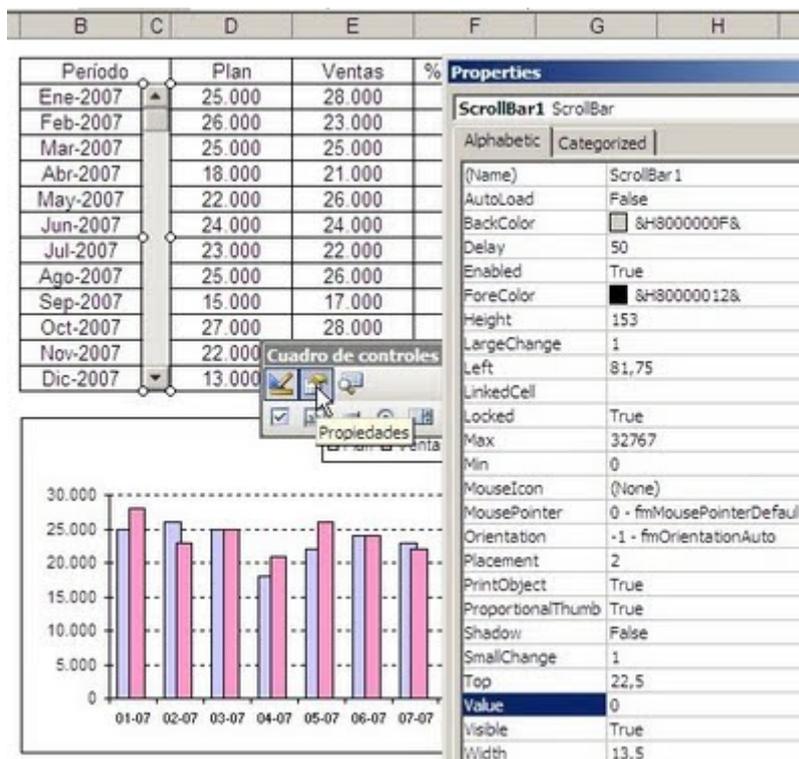


Ilustración 11

- **LinkedCell** = inicio (el rango control!C3 definido con ese nombre)
- **Min** = 0
- **SmallChange** = 1 (esto significa que cada vez que apretamos la flecha superior o inferior movemos una fila)
- **LargeChange** = 12 (esto hace que si apretamos en algún punto de la barra se muevan 12 filas)
- **Max** = por defecto aparece 32767. Este número lo manejaremos con una macro para evitar tener que editarlo con cada cambio en la base de datos.

Con el control seleccionado apretamos el icono “ver código”

Periodo	Plan	Ventas	% realizado
Ene-2007	25.000	28.000	12.0%
Feb-2007	26.000	23.000	-11.5%
Mar-2007	25.000	25.000	0.0%
Abr-2007	18.000	21.000	16.7%
May-2007	22.000	26.000	18.2%
Jun-2007	24.000	24.000	0.0%
Jul-2007	23.000	22.000	-4.3%
Ago-2007	25.000	26.000	4.0%
Sep-2007	15.000	17.000	13.3%
Oct-2007	27.000	28.000	3.7%
Nov-2007	22.000		
Dic-2007	13.000		

Ilustración 12

Esta acción abre un módulo de VBA en la hoja pertinente (la hoja Informe en nuestro caso)

En ese módulo ponemos esta macro

```
Private Sub ScrollBar1_Change()
    ScrollBar1.Max = Sheets("control").Range("D3")
End Sub
```

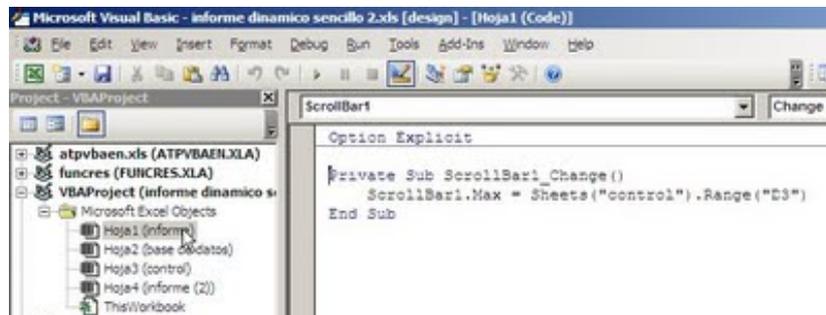


Ilustración 13

Antes de poder usar esta macro para controlar el valor de la propiedad Max tenemos que hacer algunos agregados a nuestro modelo:

definimos el nombre Periodos como rango dinámico con esta fórmula

```
=DESREF('base de datos'!$B$3;0;0;CONTARA('base de datos'!$B:$B)-1;1)
```

Esto nos permitirá saber en todo momento cuantas líneas de datos hay en nuestra tabla

en la hoja Control definimos tres celdas con nombres que contendrán las siguientes fórmulas:

- B3 (control_meses) : =CONTARA(periodos)
- C3 (inicio): 0 o en blanco; su valor es establecido por el control (LinkedCell)
- D3(max_periodo): =control_meses-12

	control_meses	=CONTARA(periodos)		
	A	B	C	D
1				
2		control_meses	inicio	max_periodo
3		34	0	22
4				

Ilustración 14

Ahora podemos ver que el valor Max está determinado por la cantidad de períodos en la base de datos menos 12 (la cantidad de líneas que aparecen en el dashboard).

La rutina ScrollBar1_Change es una macro (evento) que es disparada cada vez que accionamos la barra de desplazamiento.

Nuestro modelo se adapta ahora automáticamente a medida que vamos agregando o quitando datos en la base de datos.

2 Simulador presupuestario.xlsm: trabajando con controles ActiveX, Microsoft Office Web Components, ChartSpace, Spreadsheet, Spin, Conectar y bloquear textbox, enlazar textbox, etc..

2.1 Planteamiento del problema

	A	B	C	D	E	F	G	H
1	Simulador Presupuestario						Formulario	
2								
3	Presupuesto Nº							
4	Ingresos Previstos (Valor del Presupuesto)					133.844,00	Significación	
5							Calculada	Estandar
6	Costes Variables Estimados (Materia prima, tinta, etc.)					73.444,00	54,9%	21,3%
7								
8	Costes No Operativos Variables	2,95%				3.942,41 €		
9	Tot. Costes Fijos o Estructurales	254,0 hr	49,6 €/hr			12.601,78 €	9,4%	77,60%
10	Horas Estimadas de Trabajo							
11	<i>Nivel de Actividad</i>							
12	Dptos. Operativos	Nº de Hras	[100% / 76%]		[75% / 51%]		[- 50%]	
13	Taller I	100,00	Unitario	Total	Unitario	Total	Unitario	Total
14	Taller II	100,00	48 €/hr	4.775 €	59 €/hr	5.926 €	82 €/hr	8.230 €
15	Taller III	54,00	59 €/hr	5.937 €	72 €/hr	7.180 €	97 €/hr	9.666 €
16			35 €/hr	1.890 €	45 €/hr	2.430 €	55 €/hr	2.970 €
17								
18	Subtotal	254,0 hr	12.601,78 €	9,4%	15.536,49 €	11,6%	20.865,93 €	15,6%
19	= Resultado previsto		43.855,82 €	33%	40.921,10 €	31%	35.591,67 €	27%

Ilustración 15

Para sintetizar nuestro objetivo en este apartado presentamos a continuación, tal y como se muestra en la Ilustración 65, un caso en el que tenemos un conjunto de datos correspondientes a un simulador presupuestario. Las celdas en rojo son las variables independientes o aquellas que podemos cambiar mientras que el resto son dependientes o asociadas a una formula.

En este caso de análisis de sensibilidad consideramos oportuno contar con un formulario que nos permita el acceso a los distintos datos así como mostrar a través de un grafico incrustado los cambios en las principales variables relevantes del análisis.

Para ello hemos confeccionado un formulario al que accedemos a través del botón correspondiente de la ilustración anterior.

2.2 El formulario y análisis de los principales componentes

2.2.1 Primera aproximación al formulario "Facturas"

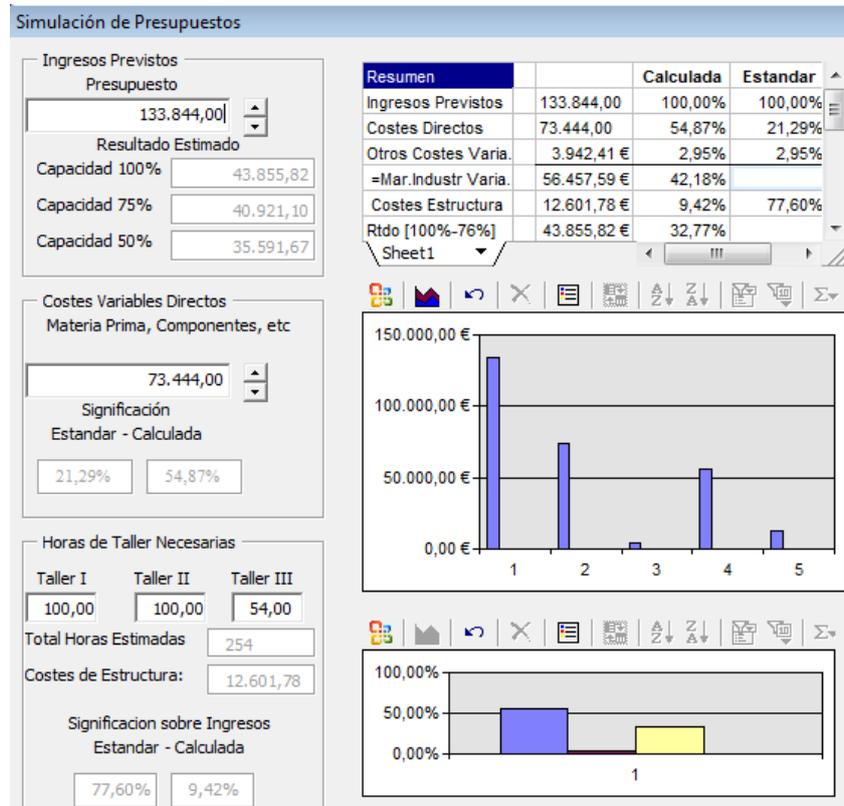


Ilustración 16

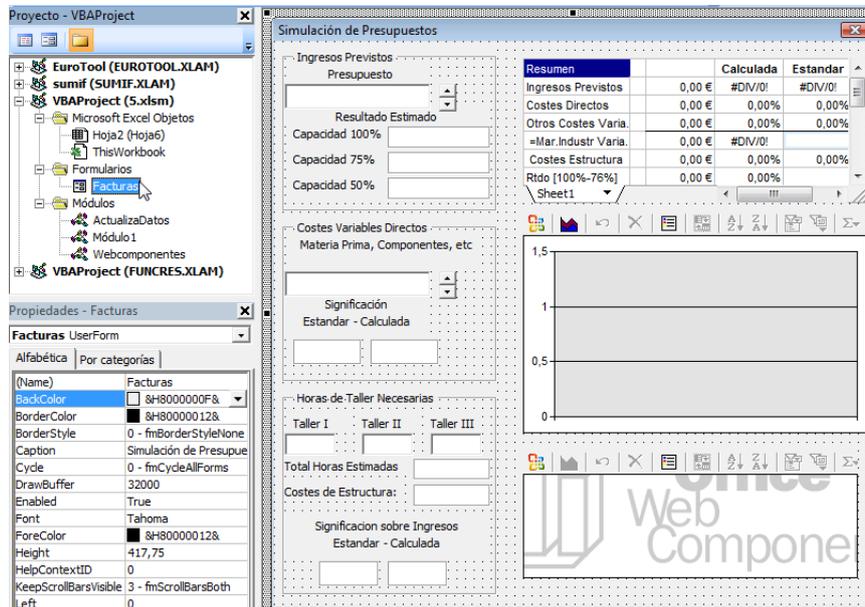


Ilustración 17

Este formulario central que hemos denominado "Facturas", contiene diferentes controles relacionados o vinculados con nuestro libro Excel y con diferentes hojas de cálculo del mismo, tal y como pasaremos a continuación a analizar.

Sin embargo destacan tipos de controles especiales, los primeros situados a la izquierda del formulario que son controles de Excel básicos o normales como control label o etiqueta, control cuadro de texto o text box, botón spin, etc. A la derecha nos encontramos dos controles diferenciados que forman parte del conjunto de controles denominados

Microsoft Office Web Components en concreto tenemos un Spreadsheet (u hoja de cálculo) y dos ChartSpace (o controles graficos), señalar que estos gráficos son dependientes de los datos disponibles en el control Spreadsheet.

2.2.2 *Análisis de los textbox del formulario. Características principales. Formato del cuadro de texto.*

Propiedades - Ingresos	
Ingresos TextBox	
Alfabética	Por categorías
(Name)	Ingresos
AutoSize	False
AutoTab	False
AutoWordSelect	True
BackColor	<input type="checkbox"/> &H80000005&
BackStyle	1 - fmBackStyleOpaque
BorderColor	<input checked="" type="checkbox"/> &H80000006&
BorderStyle	0 - fmBorderStyleNone
ControlSource	
ControlTipText	
DragBehavior	0 - fmDragBehaviorDisabled
Enabled	True
EnterFieldBehavior	0 - fmEnterFieldBehaviorSelectAll
EnterKeyBehavior	False
Font	Tahoma
ForeColor	<input checked="" type="checkbox"/> &H80000008&
Height	18
HelpContextID	0
HideSelection	True
IMEMode	0 - fmIMEModeNoControl
IntegralHeight	True
Left	0
Locked	False
MaxLength	0
MouseIcon	(Ninguno)
MousePointer	0 - fmMousePointerDefault
MultiLine	False
PasswordChar	
ScrollBars	0 - fmScrollBarsNone
SelectionMargin	True
SpecialEffect	2 - fmSpecialEffectSunken
TabIndex	1
TabKeyBehavior	False
TabStop	True
Tag	
Text	
TextAlign	3 - fmTextAlignRight
Top	18
Value	
Visible	True
Width	102
WordWrap	True

Ilustración 18

Retomando la Ilustración 66 en concreto el textbox que hemos denominado Presupuesto (aunque internamente su identificador es Ingresos) vamos a estudiar sus principales propiedades para entender uno de los aspectos básicos en los que hemos programado uno de los principales elementos que componen este formulario.

En la Ilustración 68, podemos observar las propiedades asociados al citado TextBox.

Ademas de la propiedad de este elemento queremos también en este apartado destacar otras cuestiones asociadas al mismo como son los eventos cambio y después de actualizar.

Es decir, cuando introducimos un valor en este cuadro de texto queremos que este cambio se vea reflejado o este vinculado a una celda concreta de la hoja de cálculo, para ello es necesario definir un procedimiento que realice la citada acción.

En este sentido hemos creado el procedimiento Ingresos_change

```
Private Sub Ingresos_Change()
Range("e4").Value = Ingresos.Value
' Actualizar Spreadsheet3, vuelve a importar los datos
Webcomponents.Importa_a_Spreadsheet
End Sub
```

Ilustración 19

Es decir se corresponde al evento cambio del textbox Ingreso y lo que hace es que cuando el valor del textbox cambie asigna este valor a la celda e4 de la hoja de cálculo activa.

También cuando esto ocurre los valores de la hoja de cálculo cambia y por tanto estos deben ser actualizados, en el Speedsheet, como justificaremos posteriormente.

También después de actualizar este elemento deseamos aplicar al mismo un formato así como otras acciones, para ello es necesario programar la citada acción asociado al evento AfterUpdate, tal y como mostramos en la Ilustración 70.

La función concreta es:

```
Ingresos.Text = Format(Ingresos.Text, "###0.00")
```

Conjuntamente realizamos otra serie de acciones o llamadas a módulos del libro que previamente hemos definido y explicaremos posteriormente como son actualizar los porcentajes, así como otros valores. Evidentemente al cambiar el valor de la variable Ingresos (TextBox) el resto de variables dependientes también cambiaran y por tanto será necesario actualizar los citados cambios en el formulario para que el mismo sea coincidente con los valores respectivos de las celdas de la hoja de cálculo.

```
Private Sub Ingresos_AfterUpdate()
Ingresos.Text = Format(Ingresos.Text, "#,##0.00")
'actualiza porcentajes llamamos al modulo ActualizaDatos
ActualizaDatos.Porcentajes
'actualiza los resultados al 100%, 75% y 50% llamamos al
ActualizaDatos.Resultados
'actualiza los costes de estructura
ActualizaDatos.Estructura
' Actualiza hoja de calculo Spreadsheet vuelve a importa
Webcomponentes.Importa_a_Spreadsheet
' Recargamos con datos el grafico1
Webcomponentes.grafico1actualiza
End Sub
```

Ilustración 20

Propiedades - T1hrs	
T1hrs TextBox	
Alfabética	Por categorías
(Name)	T1hrs
AutoSize	False
AutoTab	False
AutoWordSelect	True
BackColor	☐ &H80000005&
BackStyle	1 - fmBackStyleOpaque
BorderColor	☐ &H80000006&
BorderStyle	0 - fmBorderStyleNone
ControlSource	
ControlTipText	
DragBehavior	0 - fmDragBehaviorDisabled
Enabled	True
EnterFieldBehavior	0 - fmEnterFieldBehaviorSelectAll
EnterKeyBehavior	False
Font	Tahoma
ForeColor	☐ &H80000008&
Height	15,75
HelpContextID	0
HideSelection	True
IMEMode	0 - fmIMEModeNoControl
IntegralHeight	True
Left	0
Locked	False
MaxLength	0
MouseIcon	(Ninguno)
MousePointer	0 - fmMousePointerDefault
MultiLine	False
PasswordChar	
ScrollBars	0 - fmScrollBarsNone
SelectionMargin	True
SpecialEffect	2 - fmSpecialEffectSunken
TabIndex	1
TabKeyBehavior	False
TabStop	True
Tag	
Text	
TextAlign	1 - fmTextAlignLeft
Top	24
Value	
Visible	True
Width	36
WordWrap	True

Ilustración 21

De igual forma nos encontramos con las propiedades y el comportamiento del TextBox que hemos denominado “Taller I”. En este segundo ejemplo de programación del TextBox es muy similar al anterior.

En este caso en este control de cuadro de texto (y que internamente hemos llamado T1hrs) se estiman las horas necesarias de Taller I para llevar a cabo el trabajo o servicio y este elemento está asociado a la celda “B13” tal y como se puede ver en la Ilustración 65.

Igualmente asociado al mismo hemos programado dos acciones asociadas al evento Change así como al evento AfterUpdate

```
Private Sub T1hrs_Change()
Range("b13").Value = T1hrs.Value
' Actualizar Spreadsheet3, vuelve a importar los datos
Webcomponentes.Importa_a_Spreadsheet
End Sub
```

Ilustración 22

El objetivo de este evento Change es el mismo que en el caso analizado anteriormente y su significado por tanto es coincidente, comparar la Ilustración 69 y la Ilustración 72.

Igualmente tenemos asociado el evento AfterUpdate de la Ilustración 73 y que podemos comparar de igual manera con el descrito anteriormente correspondiente a la Ilustración 70

```
Private Sub T1hrs_AfterUpdate()
T1hrs.Text = Format(T1hrs.Text, "#,##0.00")
'actualiza porcentajes llamamos al modulo ActualizaDatos percent
ActualizaDatos.Porcentajes
'actualiza los resultados al 100%, 75% y 50% llamamos al modulo
ActualizaDatos.Resultados
'actualiza los costes de estructura
ActualizaDatos.Estructura
End Sub
```

Ilustración 23

2.2.3 Bloqueando los textbox

En algunos casos, en especial cuando las variables son dependientes nos puede interesar que el citado registro se muestre en el formulario pero que aparezca como bloqueado y el mismo se actualice en el momento de que cambien los valores del mismo. Este es el caso por ejemplo de los cuadros de texto correspondientes al resultado estimado para la capacidad del 100% o del total de horas estimadas necesarias de taller, tal y como podemos ver en la Ilustración 66 que aparecen con un color gris, síntoma de que no

pueden ser editados los valores en cuanto que son celdas resultantes de una fórmula o cálculo, es decir son variables dependientes.

Para analizar sus propiedades así como eventos principales asociados tomaremos como referencia la variable Resultado Estimado para la capacidad del 100% y que internamente hemos denominado al citado cuadro de texto como "Rtdo1".

Así tal y como se muestra en la Ilustración 75 y comparamos con la Ilustración 71 observamos que los únicos cambios se producen en la propiedad SpecialEffect que en este caso para que aparezca bloqueado le asignamos el valor a la citada propiedad "3 fm SpecialEffectEtched". Evidentemente también tenemos asignados un conjunto de eventos a este control concreto Change tal y como muestra la Ilustración 74 que simplemente lo que hace es cada vez que cambia su valor le vuelve a dar formato.

```
Private Sub Rtdo1_Change()  
Rtdo1.Text = Format(Rtdo1.Text, "#,##0.00")  
End Sub
```

Ilustración 24

Pero realmente para que el control quede bloqueado, hemos procedido a incorporar en el evento de inicialización del formulario la propiedad correspondiente, es decir el código:

```
Rtdo1.Enabled = False
```

Sin entrar a explicar detalladamente en este momento el código correspondiente al proceso de inicialización del formulario (ver Ilustración 76) exponemos a continuación el mismo para destacar simplemente como al inicializar el formulario de la Ilustración 66 procedemos a bloquear los Text Box o cuadros de texto deseados.

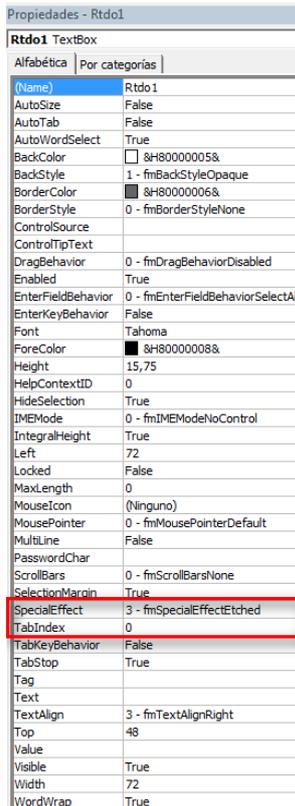


Ilustración 25

```
Private Sub UserForm_Initialize()
'Al inicializar el formulario Facturas hacemos lo siguiente
' Asignamos los valores por defecto a los campos del form y le damos forma
Ingresos.Text = Cells(4, 5)
Ingresos.Text = Format(Ingresos.Text, "#,##0.00")
GtosDirecVari.Text = Cells(6, 5)
GtosDirecVari.Text = Format(GtosDirecVari.Text, "#,##0.00")
GtosDirecVariPorEstandar.Text = Cells(6, 8)
GtosDirecVariPorEstandar.Text = Format(GtosDirecVariPorEstandar.Text, "#.0,0%")
GtosDirecVariPorCal.Text = Cells(6, 7)
GtosDirecVariPorCal.Text = Format(GtosDirecVariPorCal.Text, "#.0,0%")
T1hrs.Text = Cells(13, 2)
T1hrs.Text = Format(T1hrs.Text, "#,##0.00")
T2hrs.Text = Cells(14, 2)
T2hrs.Text = Format(T2hrs.Text, "#,##0.00")
T3hrs.Text = Cells(15, 2)
T3hrs.Text = Format(T3hrs.Text, "#,##0.00")
Rtdo1.Text = Cells(19, 3)
Rtdo1.Text = Format(Rtdo1.Text, "#,##0.00")
Rtdo2.Text = Cells(19, 5)
Rtdo2.Text = Format(Rtdo2.Text, "#,##0.00")
Rtdo3.Text = Cells(19, 7)
Rtdo3.Text = Format(Rtdo3.Text, "#,##0.00")
TotalHras.Text = Cells(9, 3)
TotalHras.Text = Format(TotalHras.Text, "#,##0.00")
TotalEstructura.Text = Cells(9, 5)
TotalEstructura.Text = Format(TotalEstructura.Text, "#,##0.00")
EstandarEstuc.Text = Cells(9, 8)
EstandarEstuc.Text = Format(EstandarEstuc.Text, "#.0,0%")
CalculadaEstuc.Text = Cells(9, 7)
CalculadaEstuc.Text = Format(CalculadaEstuc.Text, "#.0,0%")
' Bloqueamos los controles que deseamos aparecen en gris oscuro
' para que no se puedan meter datos son paneles exclusivamente informativos
GtosDirecVariPorEstandar.Enabled = False
GtosDirecVariPorCal.Enabled = False
Rtdo1.Enabled = False
Rtdo2.Enabled = False
Rtdo3.Enabled = False
TotalHras.Enabled = False
TotalEstructura.Enabled = False
EstandarEstuc.Enabled = False
CalculadaEstuc.Enabled = False
' Cargamos con datos el grafico1 y grafico 2
Webcomponentes.grafico1actualiza
Webcomponentes.grafico2actualiza
End Sub
```

Ilustración 26

El código de la Ilustración 76 está asociado al formulario facturas en concreto al evento inicializar, tal y como podemos leer en las líneas de comentarios (en color verde) en su penúltimo apartado procedemos al bloqueo de los controles de texto deseados asignándoles el valor False a la propiedad Enabled.

Junto a este conjunto de Text Box básicos que hemos analizado anteriormente nos encontramos con otros elementos que pasamos a continuación a analizar.

2.2.4 Control SpinButton (botón de número) asociado a Text Box.

El objeto de este elemento SpinButton es facilitar los cambios o variaciones de los valores numéricos asociados a una celda o TextBox. Lo que realmente hace es incrementar o disminuir el valor numérico asociado al control de cuadro de texto.

Por tanto vamos a analizar las propiedades y eventos asociados a este tipo de objeto para lo cual tomaremos como referencia los dos SpinButton usados en el formulario Facturas de la Ilustración 66, en concreto son el llamado SpinIngresos y SpinGtos.

Entre las propiedades de este tipo de objeto destacan básicamente tres, tal y como se muestra en la Ilustración 77 que son el valor máximo que deseamos asignar a este control, su valor mínimo así como el intervalo de cambio o variación (SmallChange) que deseamos asignar cada vez que pulsamos sobre el citado control, es decir es el valor de crecimiento o decrecimiento de valores.

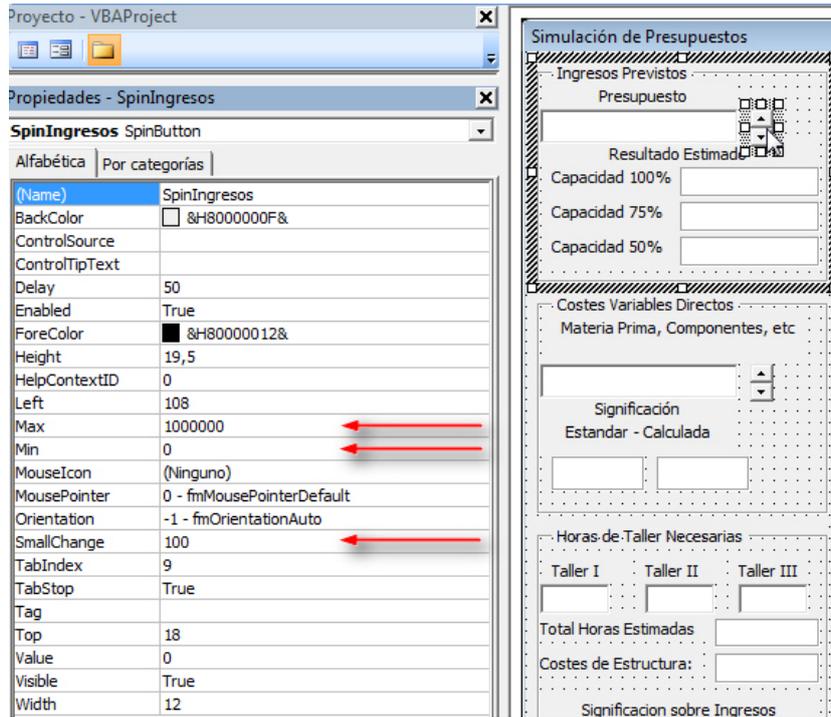


Ilustración 27

En referencia a los eventos asociados a este tipo de control son dos, el evento Enter y el Change tal y como se muestra en la Ilustración 78.

```

Private Sub SpinIngresos_Enter()
SpinIngresos.Value = Ingresos.Value
End Sub

Private Sub SpinIngresos_Change()
Ingresos.Value = SpinIngresos.Value
Range("e4").Value = Ingresos.Value
Ingresos.Text = Format(Ingresos.Text, "#,##0.00")
'actualiza porcentajes llamamos al modulo ActualizaDatos
ActualizaDatos.Porcentajes
'actualiza los resultados al 100%, 75% y 50% llamamos al
ActualizaDatos.Resultados
'actualiza los costes de estructura
ActualizaDatos.Estructura
End Sub
    
```

Ilustración 28

Así con el evento *SpinIngresos_Enter* lo que hace es que cada vez que se entre en este control el valor que toma por defecto es el que tiene el Text Box o Caja de Texto “Ingresos” del formulario activo.

Con el evento *SpinIngresos_Change*, es decir cada vez que cambia o pulsamos sobre el citado SpinButton se incrementa el valor de referencia y por tanto en primer lugar actualiza el mismo en la casilla de texto “Ingresos” del formulario y vuelve a dar formato al citado control de texto y además actualiza el valor de la celda que contiene esa referencia en la hoja de cálculo.

Como los cambios que se producen en este control afectan a variables dependientes, en concreto a la de ingresos, es necesario por tanto actualizar los datos de las mismas para

que queden correctamente reflejados en el formulario. Para ello recurriremos a un conjunto de módulos a los que invocamos, y que trataremos posteriormente.

Igualmente las propiedades así como los eventos configurados para el control SpinGtos son similares al tratado anteriormente tal y como se puede ver en la Ilustración 79 y su comparación con la Ilustración 78.

```
Private Sub SpinGtos_Enter()
SpinGtos.Value = GtosDirecVari.Value
End Sub

Private Sub SpinGtos_Change()
GtosDirecVari.Value = SpinGtos.Value
Range("e6").Value = GtosDirecVari.Value
GtosDirecVari.Text = Format(GtosDirecVari.Text, "#,##0.00")
'actualiza porcentajes llamamos al modulo ActualizaDatos porcentajes
ActualizaDatos.Porcentajes
'actualiza los resultados al 100%, 75% y 50% llamamos al modulo ActualizaDatos.Resultados
ActualizaDatos.Resultados
'actualiza los costes de estructura
ActualizaDatos.Estructura
End Sub
```

Ilustración 29

2.3 Principales eventos del formulario facturas y módulos del proyecto.

2.3.1 Evento al inicializar el formulario

Una vez diseñado y construido el formulario Facturas con todos sus elementos tal y como muestra la Ilustración 67 es necesario que cuando este se abra cargue todos los valores de los TextBox o Cuadros de Texto de las celdas de las hojas de cálculo a la que está vinculado y asigne el formato correspondiente. Para llevar a cabo esta acción, hacemos uso del evento asociado al citado formulario denominado UserForm_Initialize y en la que podemos distinguir las siguientes acciones (ver Ilustración 76):

- Cargamos los valores de las distintas cajas de texto con sus vínculos de cada una de las celdas correspondientes.
- Damos los formatos correspondientes a cada uno de estos controles.
- Bloquemos los controles deseados, especialmente los de aquellas variables dependientes para impedir su modificación.
- Cargamos los valores correspondientes a los controles ActiveX de Microsoft Office Web Components, especialmente el Spreadsheet así como los ChartSpace con el fin de dibujar los gráficos correspondientes.

En resumen, con este procedimiento cada vez que inicializamos el formulario se vuelven a cargar los datos de los distintos cuadros de texto con los valores vinculados de las celdas, se le da el formato numérico correspondiente, se bloquean aquellos valores correspondientes a las variables dependientes y finalmente se importan los datos y se dibujan los gráficos de los componentes Spreadsheet.

2.3.2 Botón inicializar/abrir el formulario. Módulo 1

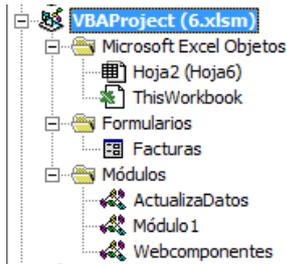


Ilustración 30

En nuestro libro Excel contamos con tres módulos (Ilustración 80), en concreto el Módulo1 es una macro o sentencia que lo que hace es abrir nuestro formulario “Facturas” y su código es tan simple como el que mostramos a continuación:

```
Sub MostrarFactura()
    Facturas.Show
End Sub
```

Por tanto lo que procede es asignar este código al botón Formulario de la Ilustración 65 a través de la asignación de la macro al citado botón “MostrarFactura” tal y como hemos denominado a este procedimiento.

2.3.3 Eventos especiales asociados a los textbox. Modulo Actualiza Datos.

En este modulo encontraremos tres procedimientos o rutinas (Ver Ilustración 81).

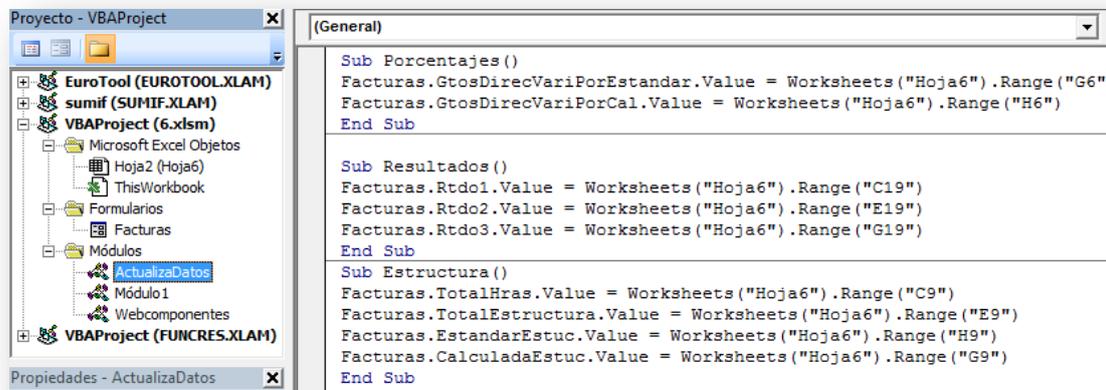


Ilustración 31

- ActualizaDatos.Porcentajes. Este procedimiento lo que hace es cargar o actualizar los valores del cuadro de porcentajes, según el siguiente código:

```
Sub Porcentajes()
    Facturas.GtosDirecVariPorEstandar.Value = Worksheets("Hoja6").Range("G6")
    Facturas.GtosDirecVariPorCal.Value = Worksheets("Hoja6").Range("H6")
End Sub
```

Es decir (Ilustración 82) asigna valores a los cuadros de porcentaje del formulario facturas contenidos en la Hoja 6 de nuestro libro y en el rango o celda correspondiente.

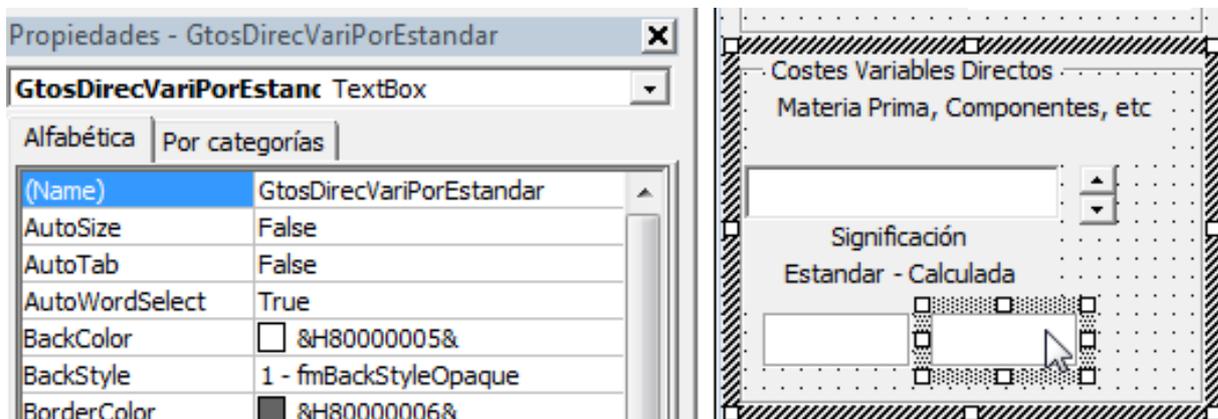


Ilustración 32

- ActualizaDatos.Resultados. En este caso el código asociados es:

```
Sub Resultados()
    Facturas.Rtdo1.Value = Worksheets("Hoja6").Range("C19")
    Facturas.Rtdo2.Value = Worksheets("Hoja6").Range("E19")
    Facturas.Rtdo3.Value = Worksheets("Hoja6").Range("G19")
End Sub
```

De forma similar al caso anterior, en esta ocasión actualizamos y asignamos valores a los cuadros de texto los valores contenidos en las hojas especificadas y en el rango correspondiente.

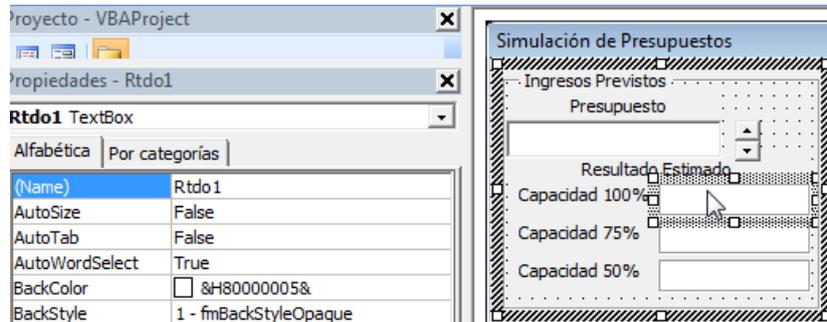


Ilustración 33

- ActualizarDatos.Estructura. El código asociado a esta parte del módulo es:

```
Sub Estructura()
    Facturas.TotalHras.Value = Worksheets("Hoja6").Range("C9")
    Facturas.TotalEstructura.Value = Worksheets("Hoja6").Range("E9")
    Facturas.EstandarEstuc.Value = Worksheets("Hoja6").Range("H9")
    Facturas.CalculadaEstuc.Value = Worksheets("Hoja6").Range("G9")
End Sub
```

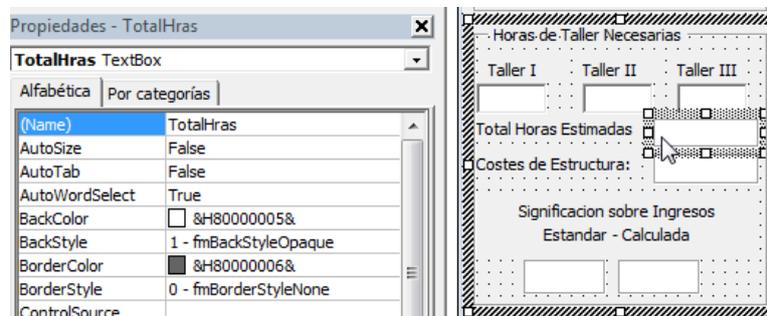


Ilustración 34

2.3.4 Llamada a los módulos desde los TextBox independientes.

Por tanto tomando en consideración estos módulos, podemos ver que cada vez que cambian los valores de los textbox independientes nuestro objetivo se centra en actualizar los valores en el formulario, para ello hacemos una llamada a los módulos anteriores.

Por ejemplo, cada vez que cambiamos el cuadro de texto "Ingresos" del formulario "Facturas" (Ilustración 69 e Ilustración 70) llama al método ActualizaDatos y sus distintos procedimientos para reflejar estos cambios en las variables dependientes en el formulario.

De igual forma por ejemplo cuando otra variable independiente como puede ser el TextBox "T1hrs" (Ilustración 71, Ilustración 72 e Ilustración 73) cambia esto afecta a otras variables dependientes y por tanto debemos llamar a este modulo para proceder a reflejar estas actualizaciones.

En definitiva, en todas las variables independientes del formulario factura y para el evento AfterUpdate hemos añadido las siguientes sentencias que ejecutan básicamente el módulo ActualizaDatos.

'actualiza porcentajes llamamos al modulo ActualizaDatos porcentajes
ActualizaDatos.Porcentajes

'actualiza los resultados al 100%, 75% y 50% llamamos al modulo ActualizaDatos resultados
ActualizaDatos.Resultados

'actualiza los costes de estructura
ActualizaDatos.Estructura

Señalar finalmente que el modulo que hemos denominado como webcomponente contiene las rutinas asociadas a los controles ActiveX de Microsoft Office Web Components, en concreto el Spreadsheet así como los ChartSpace, estos serán analizados con más detalle en el siguiente apartado.

2.4 Controles ActiveX, Spreadsheet y ChartSpace

2.4.1 Introducción

Retomando la Ilustración 66 observamos en el margen derecho del formulario la existencia de dos tipos de componentes especiales de la familia de Controles ActiveX Microsoft Office Web Components, en concreto Spreadsheet y dos ChartSpace.

Estos controles nos van a permitir interactuar con el resto del libro y de sus hojas de cálculo así como representar los valores mediante gráficos.

Destacar previamente que para poder tener estos disponibles en nuestra cuadro de herramienta de la cinta de programación VBA es necesario cargarlo ya que estos no vienen incorporados por defecto, para ello accederemos desde Microsoft Visual Basic a la opción Herramientas-Controles adicionales tal y como mostramos en la Ilustración 85.

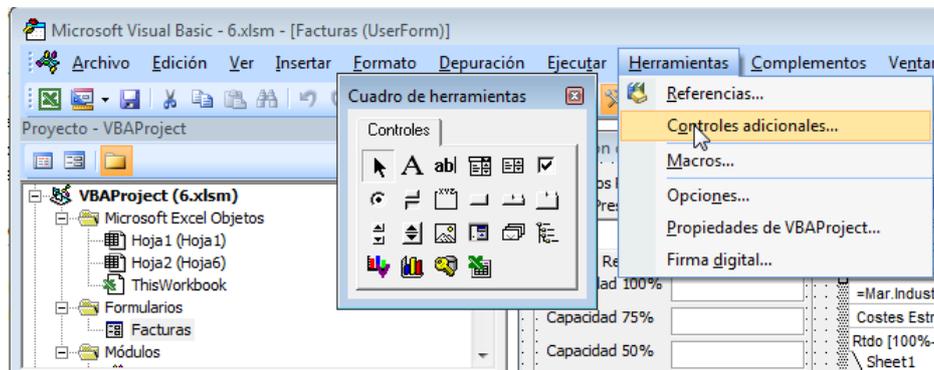


Ilustración 35

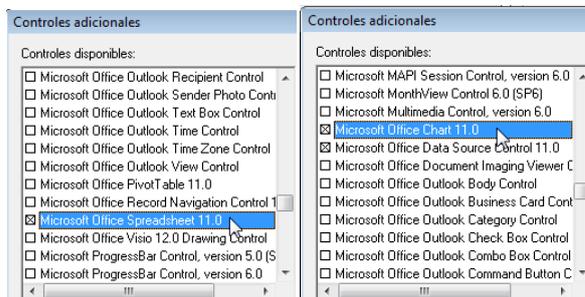


Ilustración 36

De esta forma accedemos a una pantalla adicional donde podremos seleccionar los controles necesarios que deseamos incorporar, en nuestro caso son los señalados en la Ilustración 86.

Como podemos observar de este cuadro de dialogo disponemos de multitud de controles que podemos añadir a nuestro cuadro de herramientas

En caso de que estos no esten disponibles en nuestro equipo deberemos descargarlo desde la web [“Herramienta de Office XP: Web Components”](#) o para la versión de [Office 2007](#).

Igualmente señalar también que si desarrollamos alguna aplicación excel con estos componentes el usuario final para poder usarlos se los deberá también que descargar.

1.1.1 Análisis del componente Spreadsheet.

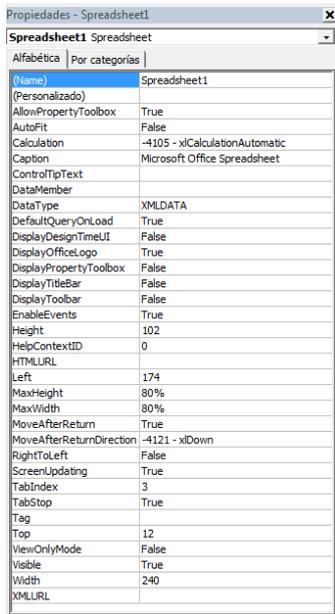


Ilustración 37

Partiendo de que el componente lo tenemos disponible, insertamos en nuestro formulario el citado Spreadsheet y podremos ver sus propiedades. Además si pulsamos sobre el mismo podremos acceder también a la configuración del mismo, para ello debemos situarnos dentro de una celda del Spreadsheet y con el botón derecho accedemos al conjunto de comandos y opciones disponibles, ver Ilustración 88.

Así podremos proteger las hojas de cálculo a mostrar, el número de ellas, así como los desplazamientos, etc. También nos permitirá dar formato a las celdas a través de la pestaña format.

Contamos además con la posibilidad de crear nuevas hojas (sheet), eliminarlas, modificarlas, etc.

Especial atención merece la pestaña fórmula a través de la cual podemos incorporar fórmulas y referencias a las celdas de forma similar a como trabajamos tradicionalmente con nuestra hoja de cálculo, evidentemente con ciertas peculiaridades.

Podremos ocultar los encabezados de fila y columna, disponer de un menú contextual asociado al citado componente.

Destacar que no podemos vincular directamente los datos de nuestra hoja de cálculo a nuestro Spreadsheet1 sino que deberemos importarlos a través de una rutina.

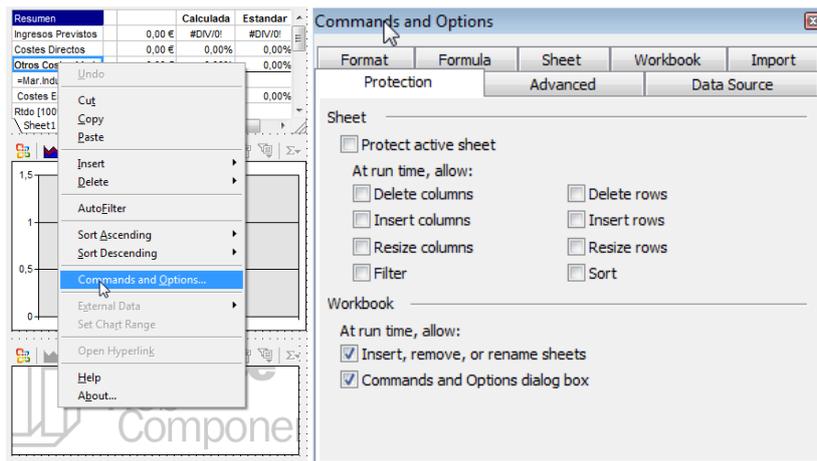


Ilustración 38

2.4.2 Análisis del Módulo Webcomponents. Caso Spreadsheet

Debemos tener en cuenta que con este componente no podemos leer de forma directa los datos de nuestra hoja de cálculo, pero si podemos importarlos para trabajar con ellos en nuestro Spreadsheet. Así para trabajar con los datos disponibles en nuestro libro de trabajo hemos establecido un procedimiento dentro del módulo Webcomponents que hemos denominado "Importa_a_SpreadSheet" tal y como se muestra en la Ilustración 89.

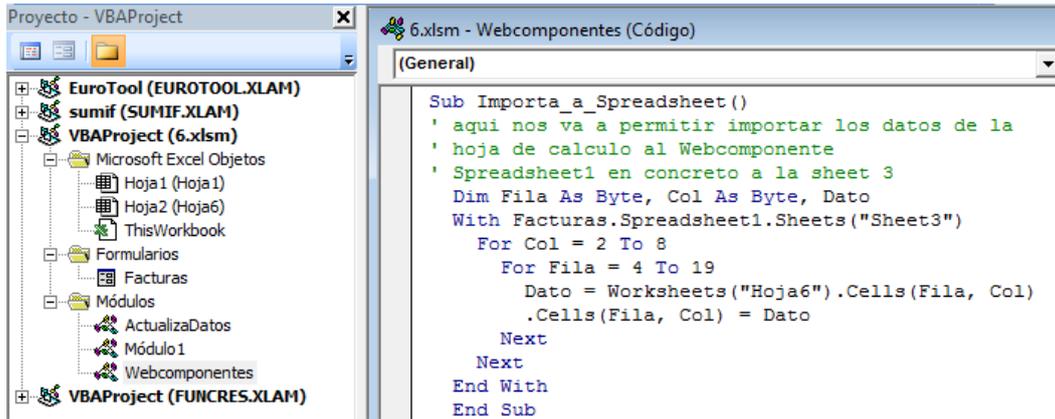


Ilustración 39

Lo que hace este procedimiento es leer los datos de nuestra “Hoja 6” (ver Ilustración 65) e incorporarlos a nuestro Spreadsheet en concreto a la hoja Sheet 3, tal y como vemos en la Ilustración 90 y posteriormente ocultamos para no estar visible para el usuario.

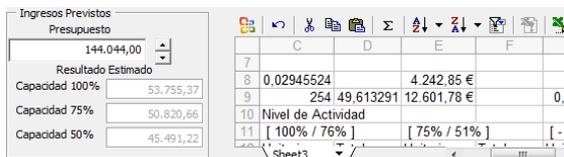


Ilustración 40

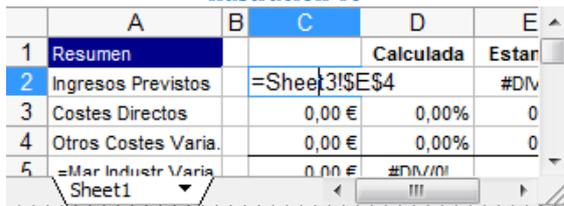


Ilustración 41

Una vez importados los datos necesarios podemos hacer referencia a los mismos dentro de cualquier Sheet del componente Spreadsheet con el nombre de la hoja y su celda de ubicación tal y como muestra la Ilustración 91.

Por ejemplo, leemos los Ingresos Previstos a través de la fórmula:

$$=Sheet3!E4$$

Y que corresponde a los datos importados anteriormente a través del procedimiento “Importa_a_SpreadSheet”.

Debemos tener en cuenta que cada vez que modificamos un valor en la hoja de calculo debemos invocar este procedimiento para actualizar los mismos en el Spreadsheet, es decir a través de una llamada como:

Webcomponentes.Importa_a_Spreadsheet

Así por ejemplo en las Ilustración 69, Ilustración 70 e Ilustración 72 tenemos varios ejemplos que al cambiar el valor de algunos cuadros de texto independientes del formulario automáticamente actualizamos los valores del componente Spreadsheet en a través de una llamada a la función anterior.

2.4.3 Análisis del Módulo Webcomponentes. Caso ChartSpace

Respecto al componente ChartSpace como podemos observar de nuestro formulario de ejemplo “Facturas” presentamos dos gráficos que hemos denominado ChartSpace1 y ChartSpace2. Los datos necesarios para representar los valores deben estar contenidos en otro componente Spreadsheet, así por ejemplo podemos preparar un nuevo sheet para representar los gráficos.

Evidentemente, en este caso también cada vez que cambiemos los valores de las celdas independientes queremos que el gráfico se actualiza, para ello hemos creado dentro del módulo Webcomponentes dos procedimientos uno para cada gráfico que hemos denominado “Grafico1actualiza” y “Grafico2actualiza” cuya única diferencia son los datos ha representar, ver Ilustración 92.

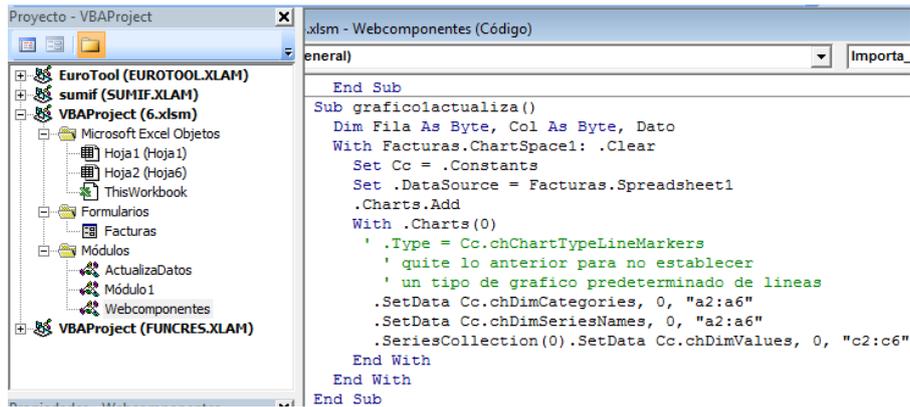


Ilustración 42

3 Bibliografía, ficheros y enlaces

Bibliografía

Caso simulador presupuestario totalmente autor: jggomez

Ficheros

Simulador Presupuestario.xlsm

Enlaces

<http://jldexcelsp.blogspot.com/2009/11/cuadro-de-control-dashboard-sencillo.html>

http://jldexcelsp.blogspot.com/2009/11/cuadro-de-control-dashboard-sencillo_17.html