

# **‘Anexo’**

**Programación Visual Basic  
con Excel 2010**

Ra-Ma<sup>®</sup>



# **‘Anexo’**

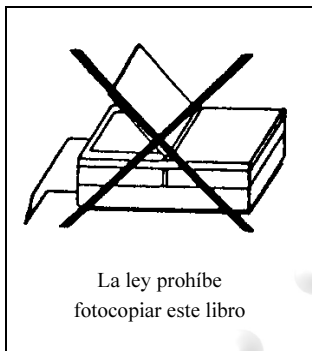
## **Programación Visual Basic con Excel 2010**

*Juan Antonio Gómez Gutiérrez*

# Ra-Ma<sup>®</sup>



**Ra-Ma<sup>®</sup>**



Programación Visual Basic con Excel 2010

© Juan Antonio Gómez Gutiérrez

© De la edición Ra-Ma 2013

**MARCAS COMERCIALES.** Las designaciones utilizadas por las empresas para distinguir sus productos (hardware, software, sistemas operativos, etc.) suelen ser marcas registradas. RA-MA ha intentado a lo largo de este libro distinguir las marcas comerciales de los términos descriptivos, siguiendo el estilo que utiliza el fabricante, sin intención de infringir la marca y sólo en beneficio del propietario de la misma. Los datos de los ejemplos y pantallas son ficticios a no ser que se especifique lo contrario.

RA-MA es marca comercial registrada.

Se ha puesto el máximo empeño en ofrecer al lector una información completa y precisa. Sin embargo, RA-MA Editorial no asume ninguna responsabilidad derivada de su uso ni tampoco de cualquier violación de patentes ni otros derechos de terceras partes que pudieran ocurrir. Esta publicación tiene por objeto proporcionar unos conocimientos precisos y acreditados sobre el tema tratado. Su venta no supone para el editor ninguna forma de asistencia legal, administrativa o de ningún otro tipo. En caso de precisarse asesoría legal u otra forma de ayuda experta, deben buscarse los servicios de un profesional competente.

Reservados todos los derechos de publicación en cualquier idioma.

Según lo dispuesto en el Código Penal vigente ninguna parte de este libro puede ser reproducida, grabada en sistema de almacenamiento o transmitida en forma alguna ni por cualquier procedimiento, ya sea electrónico, mecánico, reprográfico, magnético o cualquier otro sin autorización previa y por escrito de RA-MA; su contenido está protegido por la Ley vigente que establece penas de prisión y/o multas a quienes, intencionadamente, reprodujeren o plagiaran, en todo o en parte, una obra literaria, artística o científica.

Editado por:

RA-MA Editorial

Calle Jarama, 3A, Polígono Industrial Igarsa

28860 PARACUELLOS DE JARAMA, Madrid

Teléfono: 91 658 42 80

Fax: 91 662 81 39

Correo electrónico: [editorial@ra-ma.com](mailto:editorial@ra-ma.com)

Internet: [www.ra-ma.es](http://www.ra-ma.es) y [www.ra-ma.com](http://www.ra-ma.com)

Maquetación: Gustavo San Román Borruco

Ra-Ma®



# ÍNDICE

---

<b>CAPÍTULO 1. FUNCIONES VBA.....</b>	<b>13</b>
1.1 FUNCIONES PARA MANEJO DE CADENAS DE CARACTERES.....	13
1.1.1 Asc.....	15
1.1.2 AscW.....	15
1.1.3 Chr.....	16
1.1.4 ChrW.....	17
1.1.5 InStr.....	17
1.1.6 InStrRev.....	19
1.1.7 LCase.....	20
1.1.8 Left.....	21
1.1.9 Len.....	21
1.1.10 LTrim.....	22
1.1.11 Mid.....	23
1.1.12 Replace.....	23
1.1.13 Right.....	25
1.1.14 RTrim.....	26
1.1.15 Space.....	26
1.1.16 StrComp.....	27
1.1.17 StrConv.....	28
1.1.18 String.....	30
1.1.19 StrReverse.....	30
1.1.20 Trim.....	31
1.1.21 Ucase.....	32

1.2 FORMATEO DE VALORES.....	32
1.2.1 Format .....	33
1.2.2 FormatCurrency .....	43
1.2.3 FormatDateTime .....	44
1.2.4 FormatNumber .....	45
1.2.5 FormatPercent .....	46
1.2.6 Formatos numéricos.....	47
1.3 FUNCIONES DE ARRAY .....	48
1.3.1 Array.....	49
1.3.2 Filter .....	50
1.3.3 Join .....	51
1.3.4 LBound.....	52
1.3.5 Split .....	53
1.3.6 UBound .....	55
1.4 FUNCIONES DE FECHA.....	55
1.4.1 Intervalos de tiempo.....	57
1.4.2 Días de la semana.....	57
1.4.3 Date .....	58
1.4.4 DateAdd .....	59
1.4.5 DateDiff.....	60
1.4.6 DatePart.....	61
1.4.7 DateSerial .....	62
1.4.8 DateValue.....	62
1.4.9 Day .....	63
1.4.10 Hour.....	63
1.4.11 Minute .....	64
1.4.12 Month .....	65
1.4.13 MonthName.....	65
1.4.14 Now .....	66
1.4.15 Second .....	67
1.4.16 Time .....	67
1.4.17 Timer .....	68
1.4.18 TimeSerial .....	69
1.4.19 WeekDay.....	69
1.4.20 WeekdayName .....	70
1.4.21 Year .....	71
1.5 FUNCIONES MATEMÁTICAS .....	71

1.5.1 Abs .....	72
1.5.2 Atn .....	72
1.5.3 Cos .....	73
1.5.4 Exp .....	74
1.5.5 Log .....	74
1.5.6 Randomize .....	75
1.5.7 Rnd .....	76
1.5.8 Round .....	77
1.5.9 Sgn .....	78
1.5.10 Sin .....	79
1.5.11 Sqr .....	79
1.5.12 Tan .....	80
1.6 FUNCIONES FINANCIERAS .....	81
1.6.1 DDB .....	82
1.6.2 FV .....	83
1.6.3 IPmt .....	84
1.6.4 IRR .....	86
1.6.5 MIRR .....	87
1.6.6 NPer .....	88
1.6.7 NPV .....	90
1.6.8 Pmt .....	91
1.6.9 PPmt .....	92
1.6.10 PV .....	93
1.6.11 Rate .....	95
1.6.12 SLN .....	96
1.6.13 SYD .....	98
1.7 FUNCIONES DE COMPROBACIÓN .....	99
1.7.1 IsArray .....	100
1.7.2 IsDate .....	101
1.7.3 IsEmpty .....	102
1.7.4 IsError .....	103
1.7.5 IsNull .....	104
1.7.6 IsNumeric .....	105
1.7.7 IsObject .....	106
1.7.8 Nothing .....	107
1.8 FUNCIONES DE CONVERSIÓN DE TIPO .....	108
1.8.1 Cbool .....	109

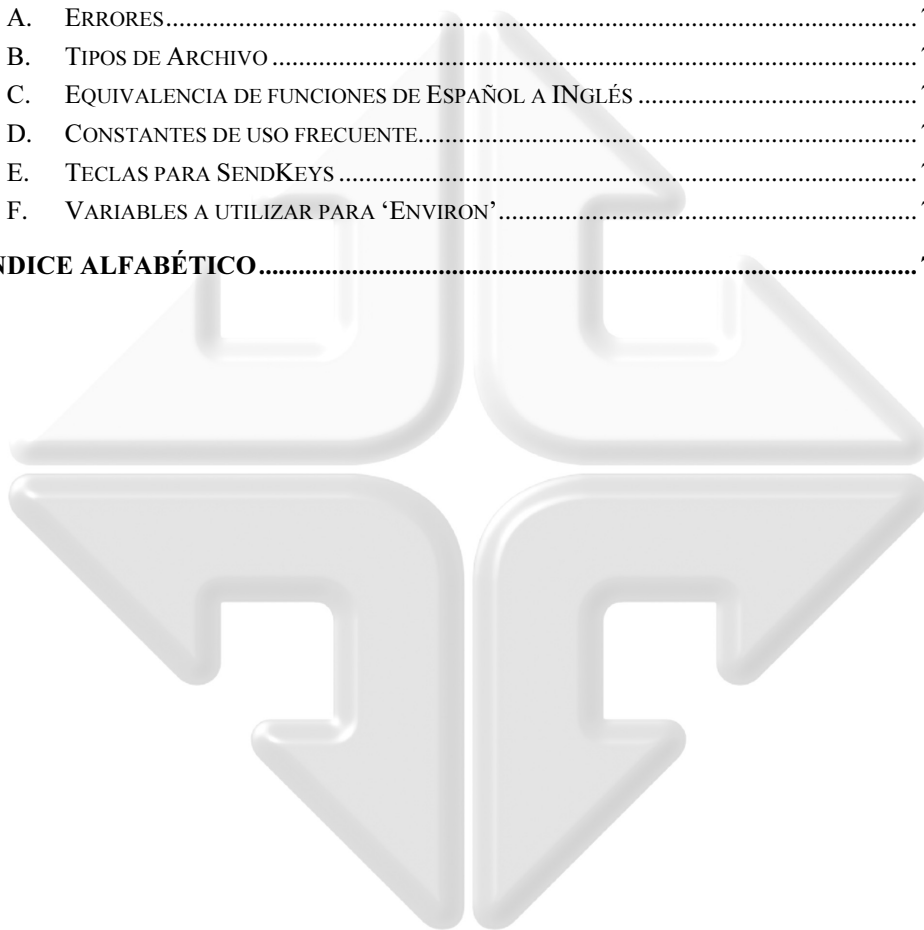
1.8.2 Cbyte .....	110
1.8.3 Ccur .....	110
1.8.4 Cdate.....	111
1.8.5 CDbI .....	112
1.8.6 Cdec.....	112
1.8.7 CInt.....	113
1.8.8 CLng.....	114
1.8.9 CLngLng .....	114
1.8.10 CSng.....	115
1.8.11 CStr.....	116
1.8.12 CVar .....	117
1.8.13 CVDate.....	117
1.8.14 CVer .....	118
1.8.15 Fix.....	119
1.8.16 Hex .....	120
1.8.17 Int .....	120
1.8.18 Oct .....	121
1.8.19 Str .....	122
1.8.20 Val .....	123
1.9 OTRAS FUNCIONES.....	124
1.9.1 AppActivate .....	125
1.9.2 Beep.....	127
1.9.3 CallByName.....	128
1.9.4 Choose.....	129
1.9.5 CreateObject.....	130
1.9.6 CurDir.....	133
1.9.7 DeleteSetting .....	133
1.9.8 Dir.....	135
1.9.9 DoEvents .....	136
1.9.10 Environ .....	137
1.9.11 EOF .....	138
1.9.12 Err.....	139
1.9.13 GetAllSettings .....	141
1.9.14 GetObject .....	142
1.9.15 GetSetting.....	143
1.9.16 IMESStatus.....	144
1.9.17 Partition .....	146

1.9.18 QBColor .....	147
1.9.19 RGB.....	148
1.9.20 SaveSetting.....	149
1.9.21 SendKeys.....	151
1.9.22 Shell.....	152
1.9.23 TypeName .....	153
1.9.24 VarType.....	155
<b>CAPÍTULO 2. EL MODELO DE OBJETOS.....</b>	<b>157</b>
2.1 EL OBJETO APLICACIÓN (APPLICATION) .....	157
2.1.1 Propiedades .....	157
2.1.2 Métodos.....	182
2.1.3 Eventos .....	233
2.2 EL OBJETO LIBRO (WORKBOOK). ....	237
2.2.1 Propiedades .....	237
2.2.2 Métodos.....	249
2.2.3 Eventos .....	308
2.3 EL OBJETO HOJA (WORKSHEET). ....	312
2.3.1 Propiedades .....	312
2.3.2 Métodos.....	320
2.3.3 Eventos .....	353
2.4 EL OBJETO RANGO (RANGE). ....	354
2.4.1 Propiedades .....	354
2.4.2 Métodos.....	406
<b>CAPÍTULO 3. CONTROLES .....</b>	<b>509</b>
3.1 DE FORMULARIO.....	509
3.1.1 CommandButton .....	509
3.1.2 ComboBox .....	510
3.1.3 CheckBox .....	512
3.1.4 ListBox .....	513
3.1.5 TextBox.....	514
3.1.6 ScrollBar.....	516
3.1.7 SpinButton.....	517
3.1.8 OptionButton.....	518
3.1.9 Label.....	519
3.1.10 Image.....	520
3.1.11 ToggleButton.....	521

3.1.12 Resumen de propiedades .....	522
3.1.13 Resumen de métodos.....	542
3.1.14 Resumen de eventos .....	543
<b>CAPÍTULO 4. FORMULARIOS .....</b>	<b>545</b>
4.1.1 Label.....	545
4.1.2 TextBox.....	546
4.1.3 ComboBox .....	548
4.1.4 ListBox .....	549
4.1.5 CheckBox .....	550
4.1.6 OptionButton.....	551
4.1.7 ToggleButton.....	552
4.1.8 Frame.....	553
4.1.9 CommandButton .....	554
4.1.10 TabStrip.....	555
4.1.11 MultiPage .....	556
4.1.12 ScrollBar.....	557
4.1.13 SpinButton.....	558
4.1.14 Image.....	559
4.2 RESUMEN DE PROPIEDADES .....	560
4.3 RESUMEN DE MÉTODOS .....	581
4.4 RESUMEN DE EVENTOS .....	582
<b>CAPÍTULO 5. CONTROLES DE FORMULARIO EN VBA.....</b>	<b>585</b>
5.1 CONTROLES. ....	585
5.1.1 Label.....	585
5.1.2 TextBox.....	586
5.1.3 ComboBox .....	588
5.1.4 ListBox .....	589
5.1.5 CheckBox .....	590
5.1.6 OptionButton.....	591
5.1.7 ToggleButton.....	592
5.1.8 Frame.....	593
5.1.9 CommandButton .....	594
5.1.10 TabStrip.....	595
5.1.11 MultiPage .....	596
5.1.12 ScrollBar.....	597
5.1.13 SpinButton.....	598

5.1.14 Image .....	599
5.2 RESUMEN DE PROPIEDADES .....	600
5.3 RESUMEN DE MÉTODOS .....	619
5.4 RESUMEN DE EVENTOS .....	620
<b>CAPÍTULO 6. EJEMPLOS .....</b>	<b>623</b>
6.1 HOLA MUNDO. ....	623
6.1.1 Ejecución lanzada mediante Ejecutar.....	626
6.1.2 Ejecución desde otro procedimiento. ....	626
6.1.3 Ejecución desde el Microsoft Excel.....	626
6.1.4 Ejecución desde un icono ubicado en la cinta de opciones.....	627
6.1.5 Ejecución al hacer clic sobre una imagen. ....	629
6.1.6 Ejecución desde un botón (Control de formulario).....	630
6.1.7 Ejecución desde un botón (Active X). ....	631
6.1.8 Ejecución desde opción de Backstage.....	633
6.1.9 Saludo con parámetro.....	635
6.2 EVENTOS Y CARACTERÍSTICAS DE LIBROS, HOJAS Y RANGOS.....	636
6.2.1 Al abrir y cerrar un libro. ....	636
6.2.2 Al activar una hoja. ....	640
6.2.3 Al cambiar la selección de un rango. ....	642
6.2.4 Análisis de algunas propiedades de libros, hojas y rangos .....	642
6.3 EJEMPLOS CON COLORES .....	645
6.3.1 Obtener un color.....	645
6.3.2 Mostrar una celda coloreada .....	647
6.3.3 Colorear aleatoriamente un rango .....	648
6.3.4 Ordenar por colores.....	649
6.3.5 Ordenar por colores a través de un procedimiento.....	650
6.4 'LOGS' AL ABRIR Y CERRAR LIBRO .....	651
6.5 FORMULARIOS .....	652
6.5.1 Utilidad 'Formulario' .....	652
6.5.2 Mantenimiento .....	655
6.5.3 Calculadora.....	665
6.5.4 Deduplicar filas .....	675
6.5.5 Buscador de archivos .....	685
6.5.6 Valoración equipo de trabajo .....	694
6.5.7 Insertar imágenes desde directorio.....	703
6.5.8 Ejecutar aplicaciones o programas.....	708
6.5.9 Menús .....	712

6.5.10 Animación .....	724
<b>APÉNDICES.....</b>	<b>733</b>
A. ERRORES.....	733
B. TIPOS DE ARCHIVO .....	739
C. EQUIVALENCIA DE FUNCIONES DE ESPAÑOL A INGLÉS .....	742
D. CONSTANTES DE USO FRECUENTE.....	751
E. TECLAS PARA SENDKEYS .....	751
F. VARIABLES A UTILIZAR PARA ‘ENVIRON’ .....	753
<b>ÍNDICE ALFABÉTICO.....</b>	<b>757</b>



Ra-Ma®



## FUNCIONES VBA

### 1.1 FUNCIONES PARA MANEJO DE CADENAS DE CARACTERES

Función	Descripción
Asc	Devuelve el código ASCII correspondiente al primer carácter de una cadena.
AscB	Devuelve el primer byte de la cadena.
AscW	Devuelve el código del primer carácter Unicode.
Chr	Obtiene el carácter correspondiente a un código ASCII.
ChrB	Obtiene el carácter correspondiente a un byte.
ChrW	Obtiene el carácter correspondiente a un código Unicode.
InStr	Devuelve la posición en la que se halla la primera ocurrencia de un string dentro de otro.
InStrRev	Devuelve la posición en la que se halla la primera ocurrencia de un string dentro de otro pero empezando por el final.

LCase	Convierte la cadena a minúsculas.
Left	Extrae un número determinado de caracteres de la izquierda.
Len	Devuelve el número de caracteres que contiene una cadena.
LTrim	Elimina los espacios que hubiese al principio de la cadena.
Mid	Extrae un número de caracteres de la cadena a partir de una posición.
Replace	Permite sustituir una cadena dentro de otra.
Right	Extrae un número determinado de caracteres de la derecha.
RTrim	Elimina los espacios existentes al final de la cadena.
Space	Fabrica una cadena con un número de espacios en blanco.
StrComp	Devuelve un valor que indica si un string es igual a otro o se ha de ordenar antes o después.
StrConv	Convierte una cadena (mayúsculas, minúsculas, Unicode, etc.) según la opción indicada.
String	Fabrica una cadena en la que se repite un carácter un determinado número de veces.
StrReverse	Invierte los caracteres de la cadena indicada.
Trim	Elimina los espacios de una cadena existentes al principio y al final.
Ucase	Convierte la cadena a mayúsculas.

Ra-Ma®


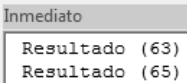
### 1.1.1 Asc

Devuelve el código ASCII correspondiente al primer carácter de una cadena.

Sintaxis:

Asc (cadena)	
Argumento	Comentario
cadena	Cadena de la que se tratará el primer carácter.

Ejemplo:

Muestra	Código	Resultado
	<pre>Sub muestra_Asc()     Dim cadena As String     cadena = Range("A1").Value     Debug.Print "Resultado (" &amp;     Asc(cadena) &amp; ")"     cadena = Range("A2").Value     Debug.Print "Resultado (" &amp;     Asc(cadena) &amp; ")" End Sub</pre>	

### 1.1.2 AscW

Devuelve el código del primer carácter Unicode.

Sintaxis:

AscW (cadena)	
Argumento	Comentario
cadena	Cadena de la que se tratará el primer carácter.

Ejemplo:

Muestra	Código	Resultado						
<table><tr><td></td><td>A</td></tr><tr><td>1</td><td>火</td></tr><tr><td>2</td><td>A</td></tr></table>		A	1	火	2	A	<pre>Sub muestra_AscW()   Dim cadena As String   cadena = Range("A1").Value   Debug.Print "Resultado (" &amp; AscW(cadena) &amp; ")"   cadena = Range("A2").Value   Debug.Print "Resultado (" &amp; AscW(cadena) &amp; ")" End Sub</pre>	<div>Inmediato</div> <div>Resultado (12117) Resultado (65)</div>
	A							
1	火							
2	A							

1.1.3 Chr

Obtiene el carácter del código ASCII indicado.

Sintaxis:

Chr(código)	
Argumento	Comentario
código	Código a representar.

Ejemplo:

Muestra	Código	Resultado				
<table><tr><td></td><td>A</td></tr><tr><td>1</td><td>A</td></tr></table>		A	1	A	<pre>Sub muestra_Chr()   Dim cadena As String   cadena = Chr(65)   Range("A1").Value = cadena   Debug.Print "Resultado (" &amp; cadena &amp; ")" End Sub</pre>	<div>Inmediato</div> <div>Resultado (A)</div>
	A					
1	A					


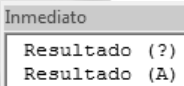
### 1.1.4 ChrW

Obtiene el carácter correspondiente a un código Unicode.

Sintaxis:

ChrW(código)	
Argumento	Comentario
código	Valor numérico del carácter a representar.

Ejemplo:

Muestra	Código	Resultado
	<pre>Sub muestra_ChrW()   Dim cadena As String   cadena = ChrW(12117)   Range("A1").Value = cadena   Debug.Print "Resultado (" &amp; cadena &amp;   ") "   cadena = ChrW(65)   Range("A2").Value = cadena   Debug.Print "Resultado (" &amp; cadena &amp;   ") " End Sub</pre>	

### 1.1.5 InStr

Devuelve la posición en la que se halla la primera ocurrencia de un string dentro de otro.

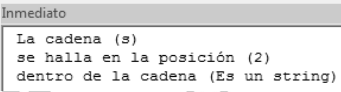
Sintaxis:

InStr( [inicio], string1, string2, [compare] )
--

Ra-Ma®

Argumento	Comentario								
<b>[inicio]</b>	Posición inicial a partir de la cual se comienza la búsqueda. Es opcional y si se omite, se empieza a buscar desde el carácter 1.								
<b>string1</b>	Cadena sobre la que se busca.								
<b>string2</b>	Cadena a buscar.								
<b>[compare]</b>	<p>Especifica el tipo de comparación a utilizar. Si no se indica nada, se utilizará lo establecido por la cláusula <b>Option Compare</b>.</p> <p>Los valores posibles son:</p> <table> <tr> <th>Valor</th><th>Comentario</th></tr> <tr> <td><b>vbUseCompareOption</b></td><td>Compara en base a la instrucción Option Compare.</td></tr> <tr> <td><b>vbBinaryCompare</b></td><td>Ejecuta una comparación binaria.</td></tr> <tr> <td><b>vbTextCompare</b></td><td>Ejecuta una comparación textual.</td></tr> </table>	Valor	Comentario	<b>vbUseCompareOption</b>	Compara en base a la instrucción Option Compare.	<b>vbBinaryCompare</b>	Ejecuta una comparación binaria.	<b>vbTextCompare</b>	Ejecuta una comparación textual.
Valor	Comentario								
<b>vbUseCompareOption</b>	Compara en base a la instrucción Option Compare.								
<b>vbBinaryCompare</b>	Ejecuta una comparación binaria.								
<b>vbTextCompare</b>	Ejecuta una comparación textual.								

Ejemplo:

Código	Resultado
<pre> Sub muestra_InStr()     Dim s As String     Dim b As String     Dim i As Integer     s = "Es un string"     b = "s"     i = InStr(1, s, b)     Debug.Print "La cadena (" &amp; b &amp; ") " &amp; Chr(13) _     &amp; "se halla en la posición (" &amp; i &amp;     ") " &amp; Chr(13) _     &amp; "dentro de la cadena (" &amp; s &amp; ") " End Sub </pre>	

### 1.1.6 InStrRev

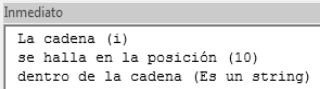
Devuelve la posición en la que se halla la primera ocurrencia de un string dentro de otro pero empezando por el final.

#### Sintaxis:

```
InStrRev(string1, string2 [, inicio[, compare ]])
```

Argumento	Comentario								
<b>string1</b>	Cadena sobre la que se busca.								
<b>string2</b>	Cadena a buscar.								
<b>[inicio]</b>	Indica la posición a partir de la cual se revisará la cadena hacia atrás. Si no se indica nada se utiliza -1 que significa empezar desde el último carácter de la cadena sobre la que se está buscando hacia atrás.								
<b>[compare]</b>	<p>Especifica el tipo de comparación a utilizar. Si no se indica nada, se utilizará lo establecido por la cláusula <b>Option Compare</b>.</p> <p>Los valores posibles son:</p> <table> <tr> <th>Valor</th><th>Comentario</th></tr> <tr> <td><b>vbUseCompareOption</b></td><td>Compara en base a la instrucción Option Compare.</td></tr> <tr> <td><b>vbBinaryCompare</b></td><td>Ejecuta una comparación binaria.</td></tr> <tr> <td><b>vbTextCompare</b></td><td>Ejecuta una comparación textual.</td></tr> </table>	Valor	Comentario	<b>vbUseCompareOption</b>	Compara en base a la instrucción Option Compare.	<b>vbBinaryCompare</b>	Ejecuta una comparación binaria.	<b>vbTextCompare</b>	Ejecuta una comparación textual.
Valor	Comentario								
<b>vbUseCompareOption</b>	Compara en base a la instrucción Option Compare.								
<b>vbBinaryCompare</b>	Ejecuta una comparación binaria.								
<b>vbTextCompare</b>	Ejecuta una comparación textual.								

Ejemplo:

Código	Resultado
<pre> Sub muestra_InstrRev()     Dim s As String     Dim b As String     Dim i As Integer     s = "Es un string"     b = "s"     i = InstrRev(s, b, 5)     Debug.Print "La cadena (" &amp; b &amp; ") " &amp; Chr(13) _     &amp; "se halla en la posición (" &amp; i &amp; ")" &amp; Chr(13) _     &amp; "dentro de la cadena (" &amp; s &amp; ")" End Sub </pre>	

### 1.1.7 LCase

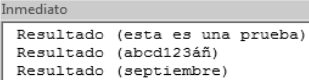
Convierte la cadena a minúsculas.

Sintaxis:

LCase(cadena)
---------------

Argumento	Comentario
Cadena	Cadena a convertir a minúsculas.

Ejemplo:

Código	Resultado
<pre> Sub muestra_LCase()     Debug.Print "Resultado (" &amp; LCase("Esta Es Una Prueba") &amp; ") "     Debug.Print "Resultado (" &amp; LCase("AbCd123ÑÑ") &amp; ") "     Debug.Print "Resultado (" &amp; LCase(MonthName(9)) &amp; ") " End Sub </pre>	



### 1.1.8 Left

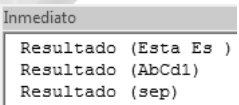
Extrae un número determinado de caracteres de la izquierda.

Sintaxis:

```
Left (cadena, numcar)
```

Argumento	Comentario
cadena	Cadena a la que extraer los caracteres.
numcar	Número de caracteres a extraer.

**Ejemplo:**

Código	Resultado
<pre>Sub muestra_Left()     Debug.Print "Resultado (" &amp; Left("Esta Es Una PruEBA", 8) &amp; ")"     Debug.Print "Resultado (" &amp; Left("AbCd123ÁÑ", 5) &amp; ")"     Debug.Print "Resultado (" &amp; Left(MonthName(9), 3) &amp; ")" End Sub</pre>	

### 1.1.9 Len

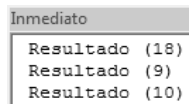
Devuelve el número de caracteres que contiene una cadena.

Sintaxis:

```
Len (cadena)
```

Argumento	Comentario
cadena	Cadena a la que se cuenta el número de caracteres.

Ejemplo:

Código	Resultado
<pre>Sub muestra_Len()     Debug.Print "Resultado (" &amp; Len("Esta Es Una PruEBA") &amp; ")"     Debug.Print "Resultado (" &amp; Len("AbCd123ÁÑ") &amp; ")"     Debug.Print "Resultado (" &amp; Len(MonthName(9)) &amp; ")" End Sub</pre>	

### 1.1.10 LTrim

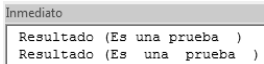
Elimina los espacios que hubiese al principio de la cadena.

Sintaxis:

LTrim(cadena)
---------------

Argumento	Comentario
cadena	Cadena a la que se le eliminan los espacios.

Ejemplo:

Código	Resultado
<pre>Sub muestra_LTrim()     Debug.Print "Resultado (" &amp; LTrim(" Es una prueba ") &amp; ")"     Debug.Print "Resultado (" &amp; LTrim(" Es una prueba ") &amp; ")" End Sub</pre>	

Ra-Ma®

### 1.1.11 Mid

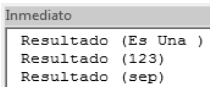
Extrae un número de caracteres de la cadena a partir de una posición.

Sintaxis:

```
Mid(cadena,posini,numcar)
```

Argumento	Comentario
cadena	Cadena a la que extraer caracteres.
posini	Posición a partir de la cual se extraerán los caracteres.
numcar	Número de caracteres a extraer.

Ejemplo:

Código	Resultado
<pre>Sub muestra_Mid()     Debug.Print "Resultado (" &amp; Mid("Esta Es Una PruEBA", 6, 7) &amp; ")"     Debug.Print "Resultado (" &amp; Mid("AbCd123Ñ", 5, 3) &amp; ")"     Debug.Print "Resultado (" &amp; Mid(MonthName(9), 1, 3) &amp; ")" End Sub</pre>	

### 1.1.12 Replace

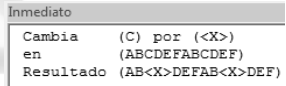
Permite sustituir una cadena dentro de otra un número de terminado de veces a partir de una posición inicial.

Sintaxis:

```
Replace(cadOri, cadAntigua, cadNueva [, posini[, numocu[, compare]])
```

Argumento	Comentario								
cadOri	Cadena a la que se efectuará la sustitución de caracteres.								
cadAntigua	Cadena que se desea reemplazar.								
cadNueva	Cadena a poner en lugar de la cadena antigua.								
posini	Posición a partir de la cual se iniciará la búsqueda. Por defecto es 1.								
numocu	Número de veces que se desea realizar la sustitución. Si no se indica nada asume el valor -1 lo cual significa reemplazar todas las ocurrencias.								
compare	Especifica el tipo de comparación a utilizar. Si no se indica nada, se utilizará lo establecido por la cláusula <b>Option Compare</b> .								
	Los valores posibles son:								
	<table><tr><th>Valor</th><th>Comentario</th></tr><tr><td><b>vbUseCompareOption</b></td><td>Compara en base a la instrucción Option Compare.</td></tr><tr><td><b>vbBinaryCompare</b></td><td>Ejecuta una comparación binaria.</td></tr><tr><td><b>vbTextCompare</b></td><td>Ejecuta una comparación textual.</td></tr></table>	Valor	Comentario	<b>vbUseCompareOption</b>	Compara en base a la instrucción Option Compare.	<b>vbBinaryCompare</b>	Ejecuta una comparación binaria.	<b>vbTextCompare</b>	Ejecuta una comparación textual.
	Valor	Comentario							
	<b>vbUseCompareOption</b>	Compara en base a la instrucción Option Compare.							
<b>vbBinaryCompare</b>	Ejecuta una comparación binaria.								
<b>vbTextCompare</b>	Ejecuta una comparación textual.								

Ejemplo:

Código	Resultado
<pre> Sub muestra_Replace()     Dim s, A, n, r As String     s = "ABCDEFGHABCDEFGH"     A = "C"     n = "&lt;X&gt;"     r = Replace(s, A, n)     Debug.Print "Cambia (" &amp; A &amp; ") por (" &amp; n &amp; ") "     Debug.Print "en (" &amp; s &amp; ") "     "     Debug.Print "Resultado (" &amp; r &amp; ") " End Sub </pre>	 <pre> Inmediato Cambia (C) por (&lt;X&gt;) en (ABCDEFGHABCDEFGH) Resultado (AB&lt;X&gt;DEFAB&lt;X&gt;DEF) </pre>

### 1.1.13 Right

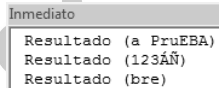
Extrae un número determinado de caracteres de la derecha.

Sintaxis:

Right (cadena,numcar)
-----------------------

Argumento	Comentario
cadena	Cadena a la que extraer los caracteres.
numcar	Número de caracteres a extraer.

Ejemplo:

Código	Resultado
<pre> Sub muestra_Right()     Debug.Print "Resultado (" &amp; Right("Esta Es Una PruEBA", 8) &amp; ") "     Debug.Print "Resultado (" &amp; Right("AbCd123ÁÑ", 5) &amp; ") "     Debug.Print "Resultado (" &amp; Right(MonthName(9), 3) &amp; ") " End Sub </pre>	 <pre> Inmediato Resultado (a PruEBA) Resultado (123ÁÑ) Resultado (bre) </pre>

### 1.1.14 RTrim

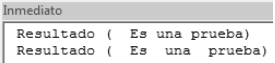
Elimina los espacios existentes al final de la cadena.

Sintaxis:

RTrim(cadena)
---------------

Argumento	Comentario
cadena	Cadena a la que se le eliminan los espacios.

Ejemplo:

Código	Resultado
<pre>Sub muestra_RTrim()     Debug.Print "Resultado (" &amp; RTrim("Es una prueba ") &amp; ")"     Debug.Print "Resultado (" &amp; RTrim("Es una prueba ") &amp; ")" End Sub</pre>	

### 1.1.15 Space

Fabrica una cadena con un número de espacios en blanco.

Sintaxis:

Space(número)
---------------

Argumento	Comentario
número	Número de espacios a incluir en la cadena.

Ejemplo:

Código	Resultado
<pre>Sub muestra_Space()     Debug.Print "Resultado -1(" &amp; Space(1)     &amp; ")2(" &amp; Space(2) &amp; _     ")3(" &amp; Space(3) &amp; ")-"     Debug.Print "Resultado (Es una" &amp; Space(1) &amp; "prueba)" End Sub</pre>	<div>Inmediato</div> <div>Resultado -1( )2( )3( )- Resultado (Es una prueba)</div>

### 1.1.16 StrComp

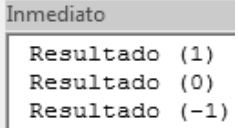
Devuelve un valor que indica si un string es igual a otro o se ha de ordenar antes o después.

Sintaxis:

```
StrComp(cadena1, cadena2 [, compare])
```

Argumento	Comentario								
cadena1	Primera cadena a comparar.								
cadena2	Segunda cadena a comparar.								
[, compare]	<p>Especifica el tipo de comparación a utilizar. Si no se indica nada, se utilizará lo establecido por la cláusula <b>Option Compare</b>.</p> <p>Los valores posibles son:</p> <table> <tr> <th>Valor</th><th>Comentario</th></tr> <tr> <td><b>vbUseCompareOption</b></td><td>Compara en base a la instrucción Option Compare.</td></tr> <tr> <td><b>vbBinaryCompare</b></td><td>Ejecuta una comparación binaria.</td></tr> <tr> <td><b>vbTextCompare</b></td><td>Ejecuta una comparación textual.</td></tr> </table>	Valor	Comentario	<b>vbUseCompareOption</b>	Compara en base a la instrucción Option Compare.	<b>vbBinaryCompare</b>	Ejecuta una comparación binaria.	<b>vbTextCompare</b>	Ejecuta una comparación textual.
Valor	Comentario								
<b>vbUseCompareOption</b>	Compara en base a la instrucción Option Compare.								
<b>vbBinaryCompare</b>	Ejecuta una comparación binaria.								
<b>vbTextCompare</b>	Ejecuta una comparación textual.								

Ejemplo:

Código	Resultado
<pre> Sub muestra_StrComp()   Dim A, b As String   A = "uno": b = "dos"   Debug.Print "Resultado (" &amp; StrComp(A, b) &amp; ")"   A = "uno": b = "uno"   Debug.Print "Resultado (" &amp; StrComp(A, b) &amp; ")"   A = "dos": b = "uno"   Debug.Print "Resultado (" &amp; StrComp(A, b) &amp; ")" End Sub </pre>	

### 1.1.17 StrConv

Convierte una cadena (mayúsculas, minúsculas, Unicode, etc.) según la opción indicada.

Sintaxis:

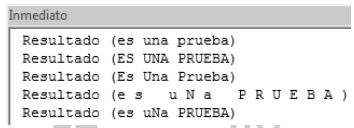
```
StrConv (string, conversión, LocalID)
```

Argumento	Comentario	
string	Cadena a convertir.	
conversión	Tipo de conversión a realizar. Puede ser alguna de las siguientes:	
	<b>Tipo</b>	<b>Comentario</b>
	vbUpperCase	Convierte la cadena a mayúsculas.
	vbLowerCase	Convierte la cadena a minúsculas.
	vbProperCase	Convierte la cadena de forma que la primera letra de cada palabra se pone en mayúscula y el resto en minúsculas.



	vbWide	Convierte los caracteres de un único byte a caracteres de doble byte.
	vbNarrow	Convierte los caracteres de doble byte a caracteres de un único byte.
	vbKatakana	Convierte los caracteres Hiragana a caracteres Katakana.
	vbHiragana	Convierte los caracteres Katakana a caracteres Hiragana.
	vbUnicode	Convierte la cadena a Unicode según la página de códigos por defecto del sistema.
	vbFromUnicode	Convierte la cadena Unicode según la página de códigos por defecto del sistema.
LocalID	Opcional. ID de la configuración regional del sistema (LCID).	

Ejemplo:

Código	Resultado
<pre> Sub muestra_StrConv()     Dim s, u, f As String     s = "es uNa PRUEBA"     Debug.Print "Resultado (" &amp;     StrConv(s, vbLowerCase) &amp; ")"     Debug.Print "Resultado (" &amp;     StrConv(s, vbUpperCase) &amp; ")"     Debug.Print "Resultado (" &amp;     StrConv(s, vbProperCase) &amp; ")"     u = StrConv(s, vbUnicode)     Debug.Print "Resultado (" &amp; u &amp;     ")"     f = StrConv(u, vbFromUnicode)     Debug.Print "Resultado (" &amp; f &amp;     ")" End Sub </pre>	 <pre> Inmediato Resultado (es una prueba) Resultado (ES UNA PRUEBA) Resultado (Es Una Prueba) Resultado (e s  u N a  P R U E B A ) Resultado (es uNa PRUEBA) </pre>

### 1.1.18 String

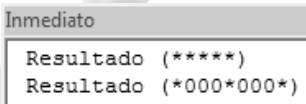
Fabrica una cadena en la que se repite un carácter un determinado número de veces.

#### Sintaxis:

```
String(número, carácter)
```

Argumento	Comentario
número	Número de caracteres a incluir.
carácter	Carácter a incluir.

#### Ejemplo:

Código	Resultado
<pre>Sub muestra_String()     Debug.Print "Resultado (" &amp; String(5,     "*") &amp; ")"     Debug.Print "Resultado (*" &amp; String(3,     "0") &amp; _     "*" &amp; String(3, "0") &amp; ")" End Sub</pre>	

### 1.1.19 StrReverse

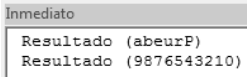
Invierte los caracteres de la cadena indicada.

#### Sintaxis:

```
StrReverse(cadena)
```

Argumento	Comentario
cadena	Cadena a la que se invertirán sus caracteres.

**Ejemplo:**

Código	Resultado
<pre>Sub muestra_StrReverse()     Debug.Print "Resultado (" &amp;     StrReverse("Prueba") &amp; ")"     Debug.Print "Resultado (" &amp;     StrReverse("0123456789") &amp; ")" End Sub</pre>	

**1.1.20 Trim**

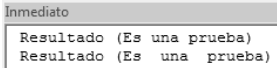
Elimina los espacios de una cadena existentes al principio y al final.

Sintaxis:

Trim(cadena)
--------------

Argumento	Comentario
cadena	Cadena a la que se eliminan los espacios del principio y del final.

Ejemplo:

Código	Resultado
<pre>Sub muestra_Trim()     Debug.Print "Resultado (" &amp; Trim(" Es     una prueba ") &amp; ")"     Debug.Print "Resultado (" &amp; Trim(" Es     una prueba ") &amp; ")" End Sub</pre>	

Ra-Ma®

### 1.1.21 Ucase

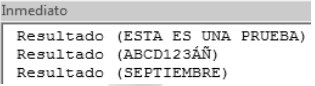
Convierte la cadena a mayúsculas.

Sintaxis:

UCase (cadena)
----------------

Argumento	Comentario
Cadena	Cadena a convertir a mayúsculas.

Ejemplo:

Código	Resultado
<pre>Sub muestra_UCase()     Debug.Print "Resultado (" &amp;     UCase("Esta Es Una PruEBA") &amp; ") "     Debug.Print "Resultado (" &amp;     UCase("AbCd123ÁÑ") &amp; ") "     Debug.Print "Resultado (" &amp;     UCase(MonthName(9)) &amp; ") " End Sub</pre>	

## 1.2 FORMATEO DE VALORES

Formato	Descripción
Format	Devuelve una expresión de tipo Variant con una expresión formateada.
FormatCurrency	Devuelve expresión con un número formateado con formato moneda.
FormatDateTime	Devuelve expresión con formato de fecha/hora.
FormatNumber	Devuelve expresión con formato número.
FormatPercent	Devuelve expresión con formato porcentaje.

## 1.2.1 Format

La función Format fabrica una expresión formateando un valor según una serie de especificaciones.

Sintaxis:

```
Format(valor [, formato[, pdíaSemana[, psemAño]])
```

Argumento	Comentario																		
valor	Valor o expresión a formatear.																		
formato	Expresión a utilizar para el formateo.																		
pdíaSemana	<p>Constante que indica cual es el primer día de la semana.</p> <p>Los posibles valores a utilizar son:</p> <table> <tr> <th>Valor</th><th>Comentario</th></tr> <tr> <td>vbUseSystem</td><td>Usa el valor del NLS del sistema.</td></tr> <tr> <td>vbSunday</td><td>Domingo.</td></tr> <tr> <td>vbMonday</td><td>Lunes.</td></tr> <tr> <td>vbTuesday</td><td>Martes.</td></tr> <tr> <td>vbWednesday</td><td>Miércoles.</td></tr> <tr> <td>vbThursday</td><td>Jueves.</td></tr> <tr> <td>vbFriday</td><td>Viernes.</td></tr> <tr> <td>vbSaturday</td><td>Sábado.</td></tr> </table>	Valor	Comentario	vbUseSystem	Usa el valor del NLS del sistema.	vbSunday	Domingo.	vbMonday	Lunes.	vbTuesday	Martes.	vbWednesday	Miércoles.	vbThursday	Jueves.	vbFriday	Viernes.	vbSaturday	Sábado.
Valor	Comentario																		
vbUseSystem	Usa el valor del NLS del sistema.																		
vbSunday	Domingo.																		
vbMonday	Lunes.																		
vbTuesday	Martes.																		
vbWednesday	Miércoles.																		
vbThursday	Jueves.																		
vbFriday	Viernes.																		
vbSaturday	Sábado.																		
psemAño	<p>Constante que indica cual es la primera semana del año.</p> <p>Los posibles valores a utilizar son:</p>																		

	Valor	Comentario
	vbUseSystem	Usa el valor del NLS del sistema.
	vbFirstJan1	Semana en la que se halle el 1 de enero. Opción por defecto.
	vbFirstFourDays	Primera semana del año que tenga 4 ó mas días.
	vbFirstFullWeek	Primera semana completa del año.

Los formatos definidos por el usuario pueden contener los siguientes caracteres:

### 1.1.1.1 NUMÉRICOS

Carácter	Comentario / Ejemplo						
Ninguno	<p>Muestra el número sin aplicar ningún formato.</p> <table> <tr> <td>Format (123)</td><td>(123)</td></tr> <tr> <td>Format (12345678)</td><td>(12345678)</td></tr> <tr> <td>Format (1234.78)</td><td>(1234,56)</td></tr> </table>	Format (123)	(123)	Format (12345678)	(12345678)	Format (1234.78)	(1234,56)
Format (123)	(123)						
Format (12345678)	(12345678)						
Format (1234.78)	(1234,56)						
0	<p>Muestra un dígito o un 0 en la posición en la que se halle.</p> <table> <tr> <td>Format (123, "00000")</td><td>(00123)</td></tr> <tr> <td>Format (12345678, "00000")</td><td>(12345678)</td></tr> <tr> <td>Format (1234.78, "00000.000")</td><td>(01234,780)</td></tr> </table>	Format (123, "00000")	(00123)	Format (12345678, "00000")	(12345678)	Format (1234.78, "00000.000")	(01234,780)
Format (123, "00000")	(00123)						
Format (12345678, "00000")	(12345678)						
Format (1234.78, "00000.000")	(01234,780)						
#	Muestra un dígito o nada en la posición en la que se halle.						

	<table> <tr> <td>Format (123, "#####")</td><td>(123)</td></tr> <tr> <td>Format (1234.5678, "#####.#")</td><td>(12345678)</td></tr> <tr> <td>Format (1234.5678, "####.#####")</td><td>(1234,56)</td></tr> </table>	Format (123, "#####")	(123)	Format (1234.5678, "#####.#")	(12345678)	Format (1234.5678, "####.#####")	(1234,56)
Format (123, "#####")	(123)						
Format (1234.5678, "#####.#")	(12345678)						
Format (1234.5678, "####.#####")	(1234,56)						
.	<p>Indica dónde se sitúa la posición decimal (coma o punto)</p> <table> <tr> <td>Format (1234.5678, "#####.###")</td><td>(1234,568)</td></tr> <tr> <td>Format (1234.5678, "#####.#")</td><td>(1234,6)</td></tr> <tr> <td>Format (1234.5678, "####.#####")</td><td>(1234,5678)</td></tr> </table>	Format (1234.5678, "#####.###")	(1234,568)	Format (1234.5678, "#####.#")	(1234,6)	Format (1234.5678, "####.#####")	(1234,5678)
Format (1234.5678, "#####.###")	(1234,568)						
Format (1234.5678, "#####.#")	(1234,6)						
Format (1234.5678, "####.#####")	(1234,5678)						
%	<p>La expresión se multiplica por 100 y el símbolo % se pone donde se indica en la máscara de formato.</p> <table> <tr> <td>Format (0.56789, "####.###%")</td><td>(56,789%)</td></tr> <tr> <td>Format (1234.5678, "#####.###%")</td><td>(123456,78%)</td></tr> <tr> <td>Format (-12.345, "####.##%")</td><td>(-1234,5%)</td></tr> </table>	Format (0.56789, "####.###%")	(56,789%)	Format (1234.5678, "#####.###%")	(123456,78%)	Format (-12.345, "####.##%")	(-1234,5%)
Format (0.56789, "####.###%")	(56,789%)						
Format (1234.5678, "#####.###%")	(123456,78%)						
Format (-12.345, "####.##%")	(-1234,5%)						
,	<p>Indica que se desea utilizar el separador de millar. Si se indica inmediatamente a la izquierda del separador decimal, significa que el valor se ha de dividir por 1000. Si se indican 2 comas, se divide por 1000000 y así sucesivamente.</p> <table> <tr> <td>Format (0.56789, "#####.###")</td><td>(,568)</td></tr> <tr> <td>Format (1234.5678, "#,#####.##")</td><td>(1.234,57)</td></tr> <tr> <td>Format (-12.345, "#,###.#")</td><td>(-12,3)</td></tr> </table>	Format (0.56789, "#####.###")	(,568)	Format (1234.5678, "#,#####.##")	(1.234,57)	Format (-12.345, "#,###.#")	(-12,3)
Format (0.56789, "#####.###")	(,568)						
Format (1234.5678, "#,#####.##")	(1.234,57)						
Format (-12.345, "#,###.#")	(-12,3)						
E- E+ e- e+	Expresa el valor en notación científica.						

	<table> <tr> <td><code>Format (0.56789, "#####.###E+00")</code></td><td><code>(567890,E-06)</code></td></tr> <tr> <td><code>Format (12345678, "#,####.### E+")</code></td><td><code>(12.345,678 E+3)</code></td></tr> <tr> <td><code>Format (-12.345, "#,###e+000")</code></td><td><code>(-1.235e-002)</code></td></tr> </table>	<code>Format (0.56789, "#####.###E+00")</code>	<code>(567890,E-06)</code>	<code>Format (12345678, "#,####.### E+")</code>	<code>(12.345,678 E+3)</code>	<code>Format (-12.345, "#,###e+000")</code>	<code>(-1.235e-002)</code>
<code>Format (0.56789, "#####.###E+00")</code>	<code>(567890,E-06)</code>						
<code>Format (12345678, "#,####.### E+")</code>	<code>(12.345,678 E+3)</code>						
<code>Format (-12.345, "#,###e+000")</code>	<code>(-1.235e-002)</code>						
- + \$ ( )	<p>Muestra un literal. Para escribir un carácter diferente a los indicados en la izquierda, hay que anteponer la barra invertida “\”.</p> <table> <tr> <td><code>Format (123456.7, "+#,####.###")</code></td><td><code>(+123.456,7)</code></td></tr> <tr> <td><code>Format (123456.7, "\*#,####.###")</code></td><td><code>(*123.456,7)</code></td></tr> <tr> <td><code>Format (-12.345, "#,###.#\$")</code></td><td><code>(-12,3\$)</code></td></tr> </table>	<code>Format (123456.7, "+#,####.###")</code>	<code>(+123.456,7)</code>	<code>Format (123456.7, "\*#,####.###")</code>	<code>(*123.456,7)</code>	<code>Format (-12.345, "#,###.#\$")</code>	<code>(-12,3\$)</code>
<code>Format (123456.7, "+#,####.###")</code>	<code>(+123.456,7)</code>						
<code>Format (123456.7, "\*#,####.###")</code>	<code>(*123.456,7)</code>						
<code>Format (-12.345, "#,###.#\$")</code>	<code>(-12,3\$)</code>						
"ABC"	<p>Permite introducir un literal como parte del formato. Las comillas dobles se sustituyen por chr(34).</p> <table> <tr> <td><code>Debug.Print "(" &amp; Format(0.56789, _ Chr(34) &amp; "Saldo " &amp; Chr(34) _ &amp; "#####.###") &amp; ")"</code></td><td><code>(Saldo ,568)</code></td></tr> <tr> <td><code>Debug.Print "(" &amp; Format (1234.5678, _ Chr(34) &amp; "-&gt;&gt; " &amp; Chr(34) _ &amp; "#,####.##") &amp; ")"</code></td><td><code>(-&gt;&gt; 1.234,57)</code></td></tr> <tr> <td><code>Debug.Print "(" &amp; Format (-12.345, _ "#,###.#" _ &amp; Chr(34) &amp; " !!" &amp; Chr(34)) &amp; _ )"</code></td><td><code>(-12,3 !!)</code></td></tr> </table>	<code>Debug.Print "(" &amp; Format(0.56789, _ Chr(34) &amp; "Saldo " &amp; Chr(34) _ &amp; "#####.###") &amp; ")"</code>	<code>(Saldo ,568)</code>	<code>Debug.Print "(" &amp; Format (1234.5678, _ Chr(34) &amp; "-&gt;&gt; " &amp; Chr(34) _ &amp; "#,####.##") &amp; ")"</code>	<code>(-&gt;&gt; 1.234,57)</code>	<code>Debug.Print "(" &amp; Format (-12.345, _ "#,###.#" _ &amp; Chr(34) &amp; " !!" &amp; Chr(34)) &amp; _ )"</code>	<code>(-12,3 !!)</code>
<code>Debug.Print "(" &amp; Format(0.56789, _ Chr(34) &amp; "Saldo " &amp; Chr(34) _ &amp; "#####.###") &amp; ")"</code>	<code>(Saldo ,568)</code>						
<code>Debug.Print "(" &amp; Format (1234.5678, _ Chr(34) &amp; "-&gt;&gt; " &amp; Chr(34) _ &amp; "#,####.##") &amp; ")"</code>	<code>(-&gt;&gt; 1.234,57)</code>						
<code>Debug.Print "(" &amp; Format (-12.345, _ "#,###.#" _ &amp; Chr(34) &amp; " !!" &amp; Chr(34)) &amp; _ )"</code>	<code>(-12,3 !!)</code>						

# Ra-Ma®



**1.1.1.2 DE CADENA:**

Carácter	Comentario / Ejemplo	
@	Se muestra un carácter si lo hay o bien un espacio en blanco si no lo hay. En caso de que el valor tenga más caracteres que la máscara, se mostrará el valor solo hasta la longitud de la máscara. Si se indica el símbolo de admiración "!" en cuyo caso, el valor aparecerá truncado.	
	<code>Format("Prueba", "#####")</code>	( Prueba)
	<code>Format("Prueba", "###")</code>	(Prueba)
	<code>Format("Prueba", "!###")</code>	(eba)
&	Se muestra un carácter si lo hay o bien nada si no lo hay. En caso de que el valor tenga más caracteres que la máscara, se mostrará el valor solo hasta la longitud de la máscara. Si se indica el símbolo de admiración "!" en cuyo caso, el valor aparecerá truncado.	
	<code>Format("Prueba", "#####")</code>	(Prueba)
	<code>Format("Prueba", "###")</code>	(Prueba)
	<code>Format("Prueba", "!###")</code>	(eba)
<	Muestra los caracteres en minúsculas.	
	<code>Format("PRUEBA", "&lt;&lt;&lt;&lt;&lt;&lt;&lt;&lt;")</code>	(prueba)
	<code>Format("PrUEBa", "&lt;&lt;&lt;")</code>	(prueba)
>	Muestra los caracteres en mayúsculas.	
	<code>Format("prueba", "&gt;&gt;&gt;&gt;&gt;&gt;&gt;&gt;")</code>	(PRUEBA)
	<code>Format("PrUEBa", "&gt;&gt;&gt;")</code>	(PRUEBA)
!	Fuerza que el llenado sea de izquierda a derecha en lugar de derecha a izquierda (opción por defecto).	
	<code>Format("Prueba", "!#####")</code>	(Prueba )
	<code>Format("Prueba", "#####")</code>	( Prueba)

**1.1.1.3 DE FECHA:**

Carácter	Comentario / Ejemplo	
:	Separador de hora.	
	Format ("3:7", "h:m")	(3:7)
	Format ("3:7", "hh:m")	(03:7)
	Format ("3:7", "h:mm")	(3:07)
/	Separador de fecha.	
	Format ("7-01-2011", "d/m")	(7/1)
	Format ("7-01-2011", "d/m/yy")	(7/1/11)
	Format ("7-01-2011", "dd/mm")	(07/01)
	Format ("7-01-2011", "dd/mm/yy")	(07/01/11)
	Format ("7-01-2011", "dd/mm/yyyy")	(07/01/2011)
c	Muestra la fecha si no hay parte fraccionaria y la hora si no hay parte entera.	
	Format ("7-01-2011 15:35", "c")	(07/01/2011 15:35:00)
	Format (Now, "c")	(06/10/2012 20:50:15)
	Format ("15:35", "c")	(15:35:00)
d	Muestra el día sin poner cero a la izquierda si éste es menor que 10.	
	Format ("7-01-2011", "d")	(7)
dd	Muestra el día poniendo cero a la izquierda si éste es menor que 10.	
	Format ("7-01-2011", "dd")	(07)
ddd	Muestra el día abreviado.	
	Format ("7-01-2011", "ddd") & " "	(vie)

dddd	<p>Muestra el día con el nombre completo.</p> <table border="1" data-bbox="302 278 1137 309"> <tr> <td>Format ("7-01-2011", "dddd")</td><td>(viernes)</td></tr> </table>	Format ("7-01-2011", "dddd")	(viernes)				
Format ("7-01-2011", "dddd")	(viernes)						
ddddd	<p>Muestra la fecha completa.</p> <table border="1" data-bbox="302 440 1137 471"> <tr> <td>Format ("7-01-2011", "ddddd")</td><td>(07/01/2011)</td></tr> </table>	Format ("7-01-2011", "ddddd")	(07/01/2011)				
Format ("7-01-2011", "ddddd")	(07/01/2011)						
dddddd	<p>Muestra la fecha completa en formato de 'fecha larga'.</p> <table border="1" data-bbox="302 604 1137 660"> <tr> <td>Format ("7-01-2011", "dddddd")</td><td>(viernes, 07 de enero de 2011)</td></tr> </table>	Format ("7-01-2011", "dddddd")	(viernes, 07 de enero de 2011)				
Format ("7-01-2011", "dddddd")	(viernes, 07 de enero de 2011)						
w	<p>Indica cual es el número de día de la semana teniendo en cuenta que el domingo es el 1 y el sábado el 7.</p> <table border="1" data-bbox="302 893 1137 953"> <tr> <td>Format ("7-01-2011", "w")</td><td>(6)</td></tr> <tr> <td>Format ("27-01-2011", "w")</td><td>(5)</td></tr> </table>	Format ("7-01-2011", "w")	(6)	Format ("27-01-2011", "w")	(5)		
Format ("7-01-2011", "w")	(6)						
Format ("27-01-2011", "w")	(5)						
ww	<p>Indica el número de semana dentro del año.</p> <table border="1" data-bbox="302 1086 1137 1146"> <tr> <td>Format ("7-01-2011", "ww")</td><td>(2)</td></tr> <tr> <td>Format ("27-08-2011", "ww")</td><td>(35)</td></tr> </table>	Format ("7-01-2011", "ww")	(2)	Format ("27-08-2011", "ww")	(35)		
Format ("7-01-2011", "ww")	(2)						
Format ("27-08-2011", "ww")	(35)						
m	<p>Muestra el mes como un número sin ceros a la izquierda. Si acompaña a la opción h, entonces interpreta que se trata de un formato de tiempo y muestra los minutos.</p> <table border="1" data-bbox="302 1346 1137 1434"> <tr> <td>Format ("7-12-2011", "m")</td><td>(12)</td></tr> <tr> <td>Format ("27-01-2011", "m")</td><td>(1)</td></tr> <tr> <td>Format ("08:20", "h:m")</td><td>(8:20)</td></tr> </table>	Format ("7-12-2011", "m")	(12)	Format ("27-01-2011", "m")	(1)	Format ("08:20", "h:m")	(8:20)
Format ("7-12-2011", "m")	(12)						
Format ("27-01-2011", "m")	(1)						
Format ("08:20", "h:m")	(8:20)						

mm	<p>Muestra el mes mostrando un cero a la izquierda hasta ajustarlo a 2 dígitos. Si acompaña a la opción h, entonces interpreta que se trata de un formato de tiempo y muestra los minutos.</p> <table border="1" data-bbox="302 347 1138 433"> <tr> <td>Format ("7-12-2011", "mm")</td><td>(12)</td></tr> <tr> <td>Format ("27-01-2011", "mm")</td><td>(01)</td></tr> <tr> <td>Format ("08:5", "h:mm")</td><td>(8:05)</td></tr> </table>	Format ("7-12-2011", "mm")	(12)	Format ("27-01-2011", "mm")	(01)	Format ("08:5", "h:mm")	(8:05)
Format ("7-12-2011", "mm")	(12)						
Format ("27-01-2011", "mm")	(01)						
Format ("08:5", "h:mm")	(8:05)						
mmm	<p>Muestra el nombre del mes abreviado a los 3 primeros caracteres.</p> <table border="1" data-bbox="302 566 1138 624"> <tr> <td>Format (Now, "mmm") ' 01/08/2011</td><td>(ago)</td></tr> <tr> <td>Format ("27-01-2011", "mmm")</td><td>(ene)</td></tr> </table>	Format (Now, "mmm") ' 01/08/2011	(ago)	Format ("27-01-2011", "mmm")	(ene)		
Format (Now, "mmm") ' 01/08/2011	(ago)						
Format ("27-01-2011", "mmm")	(ene)						
mmmm	<p>Muestra el nombre del mes completo.</p> <table border="1" data-bbox="302 757 1138 815"> <tr> <td>Format (Now, "mmmm") ' 01/08/2011</td><td>(agosto)</td></tr> <tr> <td>Format ("27-01-2011", "mmmm")</td><td>(enero)</td></tr> </table>	Format (Now, "mmmm") ' 01/08/2011	(agosto)	Format ("27-01-2011", "mmmm")	(enero)		
Format (Now, "mmmm") ' 01/08/2011	(agosto)						
Format ("27-01-2011", "mmmm")	(enero)						
q	<p>Muestra el número de trimestre del año.</p> <table border="1" data-bbox="302 948 1138 1006"> <tr> <td>Format (Now, "q") ' 01/08/2011</td><td>(3)</td></tr> <tr> <td>Format ("27-01-2011", "q")</td><td>(1)</td></tr> </table>	Format (Now, "q") ' 01/08/2011	(3)	Format ("27-01-2011", "q")	(1)		
Format (Now, "q") ' 01/08/2011	(3)						
Format ("27-01-2011", "q")	(1)						
y	<p>Muestra el número de día del año (de 1 a 366).</p> <table border="1" data-bbox="302 1139 1138 1197"> <tr> <td>Format (Now, "y") ' 01/08/2011</td><td>(243)</td></tr> <tr> <td>Format ("27-01-2011", "y")</td><td>(27)</td></tr> </table>	Format (Now, "y") ' 01/08/2011	(243)	Format ("27-01-2011", "y")	(27)		
Format (Now, "y") ' 01/08/2011	(243)						
Format ("27-01-2011", "y")	(27)						
yy	<p>Muestra el año con 2 dígitos.</p> <table border="1" data-bbox="302 1330 1138 1388"> <tr> <td>Format (Now, "yy") ' 01/08/2011</td><td>(11)</td></tr> <tr> <td>Format ("27-01-1969", "yy")</td><td>(69)</td></tr> </table>	Format (Now, "yy") ' 01/08/2011	(11)	Format ("27-01-1969", "yy")	(69)		
Format (Now, "yy") ' 01/08/2011	(11)						
Format ("27-01-1969", "yy")	(69)						
yyyy	<p>Muestra el año con 4 dígitos.</p> <table border="1" data-bbox="302 1521 1138 1579"> <tr> <td>Format (Now, "yyyy") ' 01/08/2011</td><td>(2011)</td></tr> <tr> <td>Format ("27-01-1969", "yyyy")</td><td>(1969)</td></tr> </table>	Format (Now, "yyyy") ' 01/08/2011	(2011)	Format ("27-01-1969", "yyyy")	(1969)		
Format (Now, "yyyy") ' 01/08/2011	(2011)						
Format ("27-01-1969", "yyyy")	(1969)						

h	<p>Muestra la hora con uno o dos dígitos según si el valor es menor a 10 o no.</p> <table border="1" data-bbox="302 311 1138 371"> <tr> <td>Format (Now, "h") ' 21:30</td><td>(21)</td></tr> <tr> <td>Format ("08:20", "h")</td><td>(8)</td></tr> </table>	Format (Now, "h") ' 21:30	(21)	Format ("08:20", "h")	(8)
Format (Now, "h") ' 21:30	(21)				
Format ("08:20", "h")	(8)				
hh	<p>Muestra la hora mostrando un cero a la izquierda hasta ajustarla a 2 dígitos.</p> <table border="1" data-bbox="302 538 1138 598"> <tr> <td>Format (Now, "hh") ' 21:30</td><td>(21)</td></tr> <tr> <td>Format ("08:20", "hh")</td><td>(08)</td></tr> </table>	Format (Now, "hh") ' 21:30	(21)	Format ("08:20", "hh")	(08)
Format (Now, "hh") ' 21:30	(21)				
Format ("08:20", "hh")	(08)				
n	<p>Muestra los minutos con uno o dos dígitos según si el valor es menor a 10 o no.</p> <table border="1" data-bbox="302 766 1138 826"> <tr> <td>Format (Now, "n") ' 21:05</td><td>(5)</td></tr> <tr> <td>Format ("08:20", "n")</td><td>(20)</td></tr> </table>	Format (Now, "n") ' 21:05	(5)	Format ("08:20", "n")	(20)
Format (Now, "n") ' 21:05	(5)				
Format ("08:20", "n")	(20)				
nn	<p>Muestra los minutos mostrando un cero a la izquierda hasta ajustarla a 2 dígitos.</p> <table border="1" data-bbox="302 993 1138 1053"> <tr> <td>Format (Now, "nn") ' 21:05</td><td>(05)</td></tr> <tr> <td>Format ("08:20", "nn")</td><td>(20)</td></tr> </table>	Format (Now, "nn") ' 21:05	(05)	Format ("08:20", "nn")	(20)
Format (Now, "nn") ' 21:05	(05)				
Format ("08:20", "nn")	(20)				
s	<p>Muestra los segundos con uno o dos dígitos según si el valor es menor a 10 o no.</p> <table border="1" data-bbox="302 1221 1138 1281"> <tr> <td>Format (Now, "s") ' 21:05:34</td><td>(34)</td></tr> <tr> <td>Format ("08:20:05", "s")</td><td>(5)</td></tr> </table>	Format (Now, "s") ' 21:05:34	(34)	Format ("08:20:05", "s")	(5)
Format (Now, "s") ' 21:05:34	(34)				
Format ("08:20:05", "s")	(5)				
ss	<p>Muestra los segundos mostrando un cero a la izquierda hasta ajustarla a 2 dígitos.</p> <table border="1" data-bbox="302 1448 1138 1508"> <tr> <td>Format (Now, "ss") ' 21:05:34</td><td>(34)</td></tr> <tr> <td>Format ("08:20:05", "ss")</td><td>(05)</td></tr> </table>	Format (Now, "ss") ' 21:05:34	(34)	Format ("08:20:05", "ss")	(05)
Format (Now, "ss") ' 21:05:34	(34)				
Format ("08:20:05", "ss")	(05)				

tttt	<p>Muestra la hora completa (hora, minutos y segundos).</p> <table border="1" data-bbox="302 278 1138 338"> <tr> <td>Format (Now, "tttttt") ' 21:05:34</td><td>(21:05:34)</td></tr> <tr> <td>Format ("08:20:05", "tttttt")</td><td>(8:20:05)</td></tr> </table>	Format (Now, "tttttt") ' 21:05:34	(21:05:34)	Format ("08:20:05", "tttttt")	(8:20:05)
Format (Now, "tttttt") ' 21:05:34	(21:05:34)				
Format ("08:20:05", "tttttt")	(8:20:05)				
AM/PM	<p>Muestra los indicadores AM o PM según si es antes o después del mediodía en mayúsculas.</p> <table border="1" data-bbox="302 505 1138 566"> <tr> <td>Format (Now, "AM/PM") ' 21:05:34</td><td>(PM)</td></tr> <tr> <td>Format ("08:20:05", "AM/PM")</td><td>(AM)</td></tr> </table>	Format (Now, "AM/PM") ' 21:05:34	(PM)	Format ("08:20:05", "AM/PM")	(AM)
Format (Now, "AM/PM") ' 21:05:34	(PM)				
Format ("08:20:05", "AM/PM")	(AM)				
am/ap	<p>Muestra los indicadores am o pm según si es antes o después del mediodía en minúsculas.</p> <table border="1" data-bbox="302 733 1138 793"> <tr> <td>Format (Now, "am/pm") ' 21:05:34</td><td>(pm)</td></tr> <tr> <td>Format ("08:20:05", "am/pm")</td><td>(am)</td></tr> </table>	Format (Now, "am/pm") ' 21:05:34	(pm)	Format ("08:20:05", "am/pm")	(am)
Format (Now, "am/pm") ' 21:05:34	(pm)				
Format ("08:20:05", "am/pm")	(am)				
A/P	<p>Muestra los indicadores A o P según si es antes o después del mediodía en mayúsculas.</p> <table border="1" data-bbox="302 960 1138 1021"> <tr> <td>Format (Now, "A/P") ' 21:05:34</td><td>(P)</td></tr> <tr> <td>Format ("08:20:05", "A/P")</td><td>(A)</td></tr> </table>	Format (Now, "A/P") ' 21:05:34	(P)	Format ("08:20:05", "A/P")	(A)
Format (Now, "A/P") ' 21:05:34	(P)				
Format ("08:20:05", "A/P")	(A)				
a/p	<p>Muestra los indicadores a o p según si es antes o después del mediodía en minúsculas.</p> <table border="1" data-bbox="302 1188 1138 1248"> <tr> <td>Format (Now, "a/p") ' 21:05:34</td><td>(p)</td></tr> <tr> <td>Format ("08:20:05", "a/p")</td><td>(a)</td></tr> </table>	Format (Now, "a/p") ' 21:05:34	(p)	Format ("08:20:05", "a/p")	(a)
Format (Now, "a/p") ' 21:05:34	(p)				
Format ("08:20:05", "a/p")	(a)				

Ra-Ma<sup>®</sup>

## 1.2.2 FormatCurrency

Devuelve una cadena mostrando un número con el formato de moneda según la configuración definida en el sistema.

Sintaxis:

```
FormatCurrency (expresión[, numDec [, digIni [, parNeg [, sepMiles]]]])
```

Argumento	Comentario
Expresión	Expresión a formatear.
numDec	Número de decimales a mostrar
digIni	Mostrar 0 o no en caso de que el número empiece con 0.
parNeg	Si el número es negativo, aparece entre paréntesis o no.
sepMiles	Indica el separador de miles o no.

Los parámetros **digini**, **parNeg** y **sepMiles** pueden tener los siguientes valores:

Valor	Comentario
vbFalse	Falso. No mostrará el elemento al que se refiera.
vbTrue	Verdadero. Mostrará el elemento al que se refiera.
vbUseDefault	Dependerá de la configuración regional del equipo.

Ejemplo:

Muestra	Resultado
FormatCurrency("123456,7890")	(123.456,79 €)
FormatCurrency("123456,7890", 1)	(123.456,8 €)
FormatCurrency("-123456,7890", 1, vbTrue, vbTrue, vbFalse)	((123456,8 €))
FormatCurrency("0,7890", 1, vbFalse, vbTrue, vbFalse)	(,8 €)

### 1.2.3 FormatDateTime

Devuelve una cadena mostrando una fecha u hora.

Sintaxis:

<code>FormatDateTime (fechaHora[, nombreFormato])</code>
--

Argumento	Comentario												
fechaHora	Expresión a formatear.												
nombreFormato	<div>Tipo de formato a utilizar. Puede ser alguno de los siguientes:<table><tr><th>Valor</th><th>Comentario</th></tr><tr><td><code>vbGeneralDate</code></td><td>Se muestra la fecha y/o la hora si están presentes con un formato corto y largo respectivamente.</td></tr><tr><td><code>vbLongDate</code></td><td>Muestra la fecha en formato largo.</td></tr><tr><td><code>vbLongTime</code></td><td>Muestra la hora en formato largo.</td></tr><tr><td><code>vbShortDate</code></td><td>Muestra la fecha en formato corto.</td></tr><tr><td><code>vbShortTime</code></td><td>Muestra la hora en formato corto.</td></tr></table></div>	Valor	Comentario	<code>vbGeneralDate</code>	Se muestra la fecha y/o la hora si están presentes con un formato corto y largo respectivamente.	<code>vbLongDate</code>	Muestra la fecha en formato largo.	<code>vbLongTime</code>	Muestra la hora en formato largo.	<code>vbShortDate</code>	Muestra la fecha en formato corto.	<code>vbShortTime</code>	Muestra la hora en formato corto.
Valor	Comentario												
<code>vbGeneralDate</code>	Se muestra la fecha y/o la hora si están presentes con un formato corto y largo respectivamente.												
<code>vbLongDate</code>	Muestra la fecha en formato largo.												
<code>vbLongTime</code>	Muestra la hora en formato largo.												
<code>vbShortDate</code>	Muestra la fecha en formato corto.												
<code>vbShortTime</code>	Muestra la hora en formato corto.												

Ra-Ma®



Ejemplo:

Muestra	Resultado
<code>FormatDateTime(Now) ' 01/08/2011 - 21:05:34</code>	(01/08/2011 21:05:34)
<code>FormatDateTime("01/08/2011 21:05:34", vbGeneralDate)</code>	(01/08/2011 21:05:34)
<code>FormatDateTime("01/08/2011 21:05:34", vbLongDate)</code>	(lunes, 01 de agosto de 2011)
<code>FormatDateTime("01/08/2011 21:05:34", vbLongTime)</code>	(21:05:34)
<code>FormatDateTime("01/08/2011 21:05:34", vbShortDate)</code>	(01/08/2011)
<code>FormatDateTime("01/08/2011 21:05:34", vbShortTime)</code>	(21:05)

### 1.2.4 FormatNumber

Devuelve una cadena mostrando un número.

Sintaxis:

```
FormatNumber (expresión[, numDec [, digIni [, parNeg [, sepMiles]]]])
```

Argumento	Comentario
Expresión	Expresión a formatear.
numDec	Número de decimales a mostrar
digIni	Mostrar 0 o no en caso de que el número empiece con 0.
parNeg	Si el número es negativo, aparece entre paréntesis o no.
sepMiles	Indica el separador de miles o no.

Los parámetros **digini**, **parNeg** y **sepMiles** pueden tener los siguientes valores:

Valor	Comentario
<b>vbFalse</b>	Falso. No mostrará el elemento al que se refiera.
<b>vbTrue</b>	Verdadero. Mostrará el elemento al que se refiera.
<b>vbUseDefault</b>	Dependerá de la configuración regional del equipo.

Ejemplo:

Muestra	Resultado
<code>FormatNumber("123456,7890")</code>	<code>(123.456,79)</code>
<code>FormatNumber("123456,7890", 1)</code>	<code>(123.456,8)</code>
<code>FormatNumber("-123456,7890", 1, vbTrue, vbTrue, vbFalse)</code>	<code>((123456,8))</code>
<code>FormatNumber("0,7890", 1, vbFalse, vbTrue, vbFalse)</code>	<code>(,8)</code>

## 1.2.5 FormatPercent

Devuelve una cadena mostrando un número en forma de porcentaje al que multiplica por 100 y le añade al final el símbolo %.

Sintaxis:

<code>FormatPercent (expresión[, numDec [, digIni [, parNeg [, sepMiles]]]])</code>
---

Argumento	Comentario
Expresión	Expresión a formatear.
numDec	Número de decimales a mostrar
digIni	Mostrar 0 o no en caso de que el número empiece con 0.
parNeg	Si el número es negativo, aparece entre paréntesis o no.
sepMiles	Indica el separador de miles o no.

Los parámetros **digini**, **parNeg** y **sepMiles** pueden tener los siguientes valores:

Valor	Comentario
<code>vbFalse</code>	Falso. No mostrará el elemento al que se refiera.
<code>vbTrue</code>	Verdadero. Mostrará el elemento al que se refiera.
<code>vbUseDefault</code>	Dependerá de la configuración regional del equipo.

Ejemplo:

Código	Resultado
<code>FormatPercent("123,7890")</code>	(12.378,90%)
<code>FormatPercent("123,7890", 1)</code>	(12.378,9%)
<code>FormatPercent("-123,7890", 1, vbTrue, vbTrue, vbFalse)</code>	((12378,9%))
<code>FormatPercent("0,7890", 1, vbFalse, vbTrue, vbFalse)</code>	(78,9%)

## 1.2.6 Formatos numéricos

Para dar formatos a valores numéricos, existen una serie de formatos predefinidos que pueden simplificar en un momento dado el uso de la función. A continuación se muestra cada uno de estos formatos y un ejemplo de los mismos.

Formato	Descripción
“General Number”	Muestra el número sin separador de millares.
“Currency”	Muestra el número con el separador de millares, con dos decimales y con el símbolo de la moneda el cual irá en función de la configuración del sistema.
“Fixed”	Muestra dos dígitos a la derecha del separador decimal y uno a la izquierda si no hay indicado ninguno.
“Standard”	Muestra el número con separador de miles, con dos dígitos a la derecha del separador decimal y uno a la izquierda si no hay indicado ninguno.

“Percent”	Muestra el número multiplicado por 100 con el signo de tanto por ciento (%) añadido al final y con dos dígitos a la derecha del separador decimal.
“Scientific”	Muestra el número en coma flotante (notación científica estándar).
“Yes/No”	Muestra “No” si el número es 0 o “Yes” en cualquier otro caso.
“True/False”	Muestra “False” si el número es 0 o “True” en cualquier otro caso.
“On/Off”	Muestra “Off” si el número es 0 o “On” en cualquier otro caso.

Ejemplo:

Código	Resultado
Format("123456,7890", "General Number")	(123456,789)
Format("123456,7890", "Currency")	(123.456,79 €)
Format("123456,7890", "Fixed")	(123456,79)
Format("123456,7890", "Standard")	(123.456,79)
Format("123456,7890", "Percent")	(12345678,90%)
Format("123456,7890", "Scientific")	(1,23E+05)
Format("123456,7890", "Yes/No")	(Sí)
Format("123456,7890", "True/False")	(Verdadero)
Format("123456,7890", "On/Off")	(Activado)

### 1.3 FUNCIONES DE ARRAY

Función	Descripción
Array	Devuelve un array.
Filter	Devuelve una matriz con las cadenas que cumplen un determinado filtro aplicado a una matriz inicial dada.

Join	Devuelve una expresión en la que se combinan las cadenas contenidas en una matriz.
LBound	Devuelve el número de posición más pequeño de un array.
Split	Devuelve una matriz de una dimensión con las subcadenas resultantes de trocear un string en base a un separador determinado.
UBound	Devuelve el número de posición más grande de un array.

### 1.3.1 Array

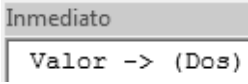
Esta función permite devolver una matriz a partir de una lista de valores o expresiones.

Sintaxis:

<code>Array(listaValores)</code>
----------------------------------

Argumento	Comentario
listaValores	Lista de valores o expresiones a utilizar para fabricar cada uno de los elementos de la matriz resultante.

Ejemplo:

Código	Resultado
<pre>Sub muestra_Array()     Dim miArray As Variant     miArray = Array("Uno", "Dos", "Tres")     Debug.Print "Valor -&gt; (" &amp; miArray(1) &amp; ")" End Sub</pre>	

Ra-Ma®

### 1.3.2 Filter

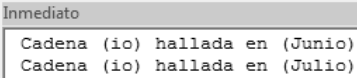
Esta función devuelve una matriz con el resultado de aplicar un determinado filtro a otra matriz inicial.

Sintaxis:

```
Filter(matrizOri, cadenaBuscar[, incluido[, compare]])
```

Argumento	Comentario								
matrizOri	Matriz sobre la cual se aplica el filtro o la búsqueda que se desea realizar.								
cadenaBuscar	Cadena de texto que se desea buscar en la matriz.								
Incluido	Flag que indica si queremos seleccionar el elemento que contiene la cadena que estamos buscando (True) o queremos incluir el que no la contiene (False).								
compare	<p>Especifica el tipo de comparación a utilizar. Si no se indica nada, se utilizará lo establecido por la cláusula <b>Option Compare</b>.</p> <p>Los valores posibles son:</p> <table><tr><th>Valor</th><th>Comentario</th></tr><tr><td><b>vbUseCompareOption</b></td><td>Compara en base a la instrucción Option Compare.</td></tr><tr><td><b>vbBinaryCompare</b></td><td>Ejecuta una comparación binaria.</td></tr><tr><td><b>vbTextCompare</b></td><td>Ejecuta una comparación textual.</td></tr></table>	Valor	Comentario	<b>vbUseCompareOption</b>	Compara en base a la instrucción Option Compare.	<b>vbBinaryCompare</b>	Ejecuta una comparación binaria.	<b>vbTextCompare</b>	Ejecuta una comparación textual.
Valor	Comentario								
<b>vbUseCompareOption</b>	Compara en base a la instrucción Option Compare.								
<b>vbBinaryCompare</b>	Ejecuta una comparación binaria.								
<b>vbTextCompare</b>	Ejecuta una comparación textual.								

Ejemplo:

Código	Resultado
<pre> Sub muestra_filter()     Dim matrizOri, matrizFin, elemento     Dim cadenaBus As String     cadenaBus = "io"     matrizOri = Array("Junio", "Julio", "Agosto")     matrizFin = Filter(matrizOri, cadenaBus)     For Each elemento In matrizFin         Debug.Print "Cadena (" &amp; cadenaBus &amp; _         ") hallada en (" &amp; elemento &amp; ")"     Next End Sub </pre>	

### 1.3.3 Join

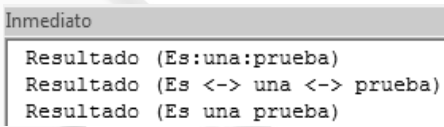
Esta función devuelve una cadena con el resultado de unir los elementos existentes en una matriz concatenando cada uno de los elementos de la misma mediante un determinado separador.

Sintaxis:

```
Join(matriz[, separador])
```

Argumento	Comentario
matriz	Matriz que contiene los elementos a concatenar.
separador	Separador a utilizar en la concatenación de cada elemento. Si no se indica nada, por defecto se utiliza el espacio como separador (" "). Si no se quiere dejar ningún espacio en blanco, puede indicarse una cadena vacía ("").

Ejemplo:

Código	Resultado
<pre>Sub muestra_Join()     Dim matrizOri     Dim s As String     matrizOri = Array("Es", "una", "prueba")     s = Join(matrizOri, ":")     Debug.Print "Resultado (" &amp; s &amp; ")"     s = Join(matrizOri, " &lt;-&gt; ")     Debug.Print "Resultado (" &amp; s &amp; ")"     s = Join(matrizOri)     Debug.Print "Resultado (" &amp; s &amp; ")" End Sub</pre>	

### 1.3.4 LBound

Mediante esta función podemos obtener el índice más bajo de una matriz o de una dimensión de la misma.

Sintaxis:

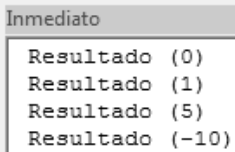
```
LBound(matriz[, dimensión])
```

Argumento	Comentario
Matriz	Matriz a tratar.
dimensión	Dimensión que se quiere analizar.

Ra-Ma®



Ejemplo:

Código	Resultado
<pre>Sub muestra_LBound()   Dim m(10) As Integer   Debug.Print "Resultado (" &amp; LBound(m) &amp; ")"   Dim n(1 To 31, 5 To 15, -10 To 20)   Debug.Print "Resultado (" &amp; LBound(n, 1) &amp;   ")"   Debug.Print "Resultado (" &amp; LBound(n, 2) &amp;   ")"   Debug.Print "Resultado (" &amp; LBound(n, 3) &amp;   ")" End Sub</pre>	 <pre> Inmediato Resultado (0) Resultado (1) Resultado (5) Resultado (-10) </pre>

### 1.3.5 Split

La función Split permite devolver una matriz con los elementos resultantes de trocear una cadena en base a un determinado separador.

Sintaxis:

Split(expresión[, separador[, numCad[, compare]])
---

Argumento	Comentario
expresión	Cadena o expresión a trocear.
Separador	Separador a utilizar para determinar por donde hay que cortar la cadena. Por defecto, se utiliza el espacio como delimitador (" ").
numCad	Número de trozos máximos que queremos obtener.
compare	Especifica el tipo de comparación a utilizar. Si no se indica nada, se utilizará lo establecido por la cláusula <b>Option Compare</b> .

	Los valores posibles son:	
	Valor	Comentario
	<b>vbUseCompareOption</b>	Compara en base a la instrucción Option Compare.
	<b>vbBinaryCompare</b>	Ejecuta una comparación binaria.
	<b>vbTextCompare</b>	Ejecuta una comparación textual.

Ejemplo:

Código	Resultado
<pre>Sub muestra_Split()   Dim s As String   Dim i As Integer   Dim m, n, o As Variant    s = "Es una prueba de Split"    m = Split(s)   Debug.Print "LBound (" &amp; LBound(m) &amp; -   ") UBound (" &amp; UBound(m) &amp; ")"   For i = LBound(m) To UBound(m)     Debug.Print "m(" &amp; i &amp; ") : " &amp; m(i)   Next i   Debug.Print    n = Split(s, " ", 3)   Debug.Print "LBound (" &amp; LBound(n) &amp; -   ") UBound (" &amp; UBound(n) &amp; ")"   For i = LBound(n) To UBound(n)     Debug.Print "n(" &amp; i &amp; ") : " &amp; n(i)   Next i End Sub</pre>	<div>Inmediato</div> <div>LBound (0) UBound (4) m(0) : Es m(1) : una m(2) : prueba m(3) : de m(4) : Split  LBound (0) UBound (2) n(0) : Es n(1) : una n(2) : prueba de Split</div>

### 1.3.6 UBound

Mediante esta función podemos obtener el índice más alto de una matriz o de una dimensión de la misma.

Sintaxis:

```
UBound(matriz[, dimensión])
```

Argumento	Comentario
Matriz	Matriz a tratar.
dimensión	Dimensión que se quiere analizar.

Ejemplo:

Código	Resultado
<pre>Sub muestra_UBound()   Dim m(10) As Integer   Debug.Print "Resultado (" &amp; UBound(m) &amp; ")"   Dim n(1 To 31, 5 To 15, -10 To 20)   Debug.Print "Resultado (" &amp; UBound(n, 1) &amp; ")"   Debug.Print "Resultado (" &amp; UBound(n, 2) &amp; ")"   Debug.Print "Resultado (" &amp; UBound(n, 3) &amp; ")" End Sub</pre>	<p>Inmediato</p> <pre>Resultado (10) Resultado (31) Resultado (15) Resultado (20)</pre>

## 1.4 FUNCIONES DE FECHA

Función	Descripción
Date	Devuelve la fecha actual del sistema.
DateAdd	Suma un intervalo de tiempo a una determinada fecha.
DateDiff	Devuelve el intervalo de tiempo existente entre dos fechas expresado en la unidad de medida que le indiquemos.

DatePart	Devuelve una parte de una fecha en función de la información solicitada (semana, trimestre, etc).
DateSerial	Devuelve una fecha a partir de la concatenación de año, mes y día.
DateValue	Devuelve una fecha a partir de una cadena que contiene una expresión de tipo fecha.
Day	Obtiene el día de una fecha.
Hour	Obtiene la hora.
Minute	Obtiene los minutos.
Month	Obtiene el mes de una fecha.
MonthName	Devuelve un string con el nombre del mes indicado.
Now	Devuelve la fecha y hora actuales.
Second	Obtiene los segundos.
Time	Devuelve la hora actual del sistema.
Timer	Devuelve el número de segundos transcurridos desde la medianoche.
TimeSerial	Compone una cadena con una hora a partir de los parámetros indicados.
WeekDay	Devuelve el día de la semana correspondiente.
WeekdayName	Devuelve el nombre de un día de la semana.
Year	Devuelve el año de una fecha determinada.

### 1.4.1 Intervalos de tiempo

El tipo de información (o intervalos de tiempo) que se requiere para las funciones **DateAdd**, **DateDiff** y **DatePart** son los siguientes:

Cadena	Unidad
“yyyy”	Año.
“q”	Trimestre.
“m”	Mes.
“y”	Día de año.
“d”	Día del mes.
“w”	Día de la semana.
“ww”	Semana.
“h”	Hora.
“n”	Minuto.
“s”	Segundo.

### 1.4.2 Días de la semana

Existen una serie de constantes que determinan cual es el día de la semana a utilizar en algunos de los parámetros solicitados por algunas funciones. Dichas constantes son las siguientes:

Ra-Ma®

Valor	Comentario
<b>vbFriday</b>	Viernes.
<b>vbMonday</b>	Lunes.
<b>vbSaturday</b>	Sábado.
<b>vbSunday</b>	Domingo.
<b>vbThursday</b>	Jueves.
<b>vbTuesday</b>	Martes.
<b>vbUseSystemDayOfWeek</b>	Día indicado en la configuración del sistema por defecto.
<b>vbWednesday</b>	Miércoles.

### 1.4.3 Date

Devuelve la fecha actual del sistema.

Sintaxis:

Date
------

Argumento	Comentario
ninguno	No tiene.

Ejemplo:

Código	Resultado
<pre>Sub muestra Date()   Dim f As Date   f = Date ' 03/09/2011   Debug.Print "Resultado (" &amp; f &amp; ")" End Sub</pre>	<p>Inmediato</p> <p>Resultado (03/09/2011)</p>

### 1.4.4 DateAdd

Suma un intervalo de tiempo a una determinada fecha.

Sintaxis:

```
DateAdd(intervalo, número, fecha)
```

Argumento	Comentario
intervalo	Indica qué tipo de información es la que se tiene que añadir (días, meses, horas, etc.). Ver tabla <b>Intervalos de tiempo</b> .
número	Número de unidades a añadir.
fecha	Fecha sobre la que añadir la información indicada en número.

Ejemplo:

Código	Resultado
<pre>Sub muestra_DateAdd()     Dim fechaHora, dias, horas, res As String     fechaHora = "03/09/2011 7:57:19"     dias = 30     res = DateAdd("d", dias, fechaHora)     Debug.Print "Resultado :"     Debug.Print "    (" &amp; fechaHora &amp; ")"     Debug.Print " + (" &amp; dias &amp; " días)"     Debug.Print " = (" &amp; res &amp; ")"     horas = 4     res = DateAdd("h", horas, fechaHora)     Debug.Print "Resultado :"     Debug.Print "    (" &amp; fechaHora &amp; ")"     Debug.Print " + (" &amp; horas &amp; " horas)"     Debug.Print " = (" &amp; res &amp; ")" End Sub</pre>	<pre>Inmediato Resultado :     (03/09/2011 7:57:19) + (30 días) = (03/10/2011 7:57:19) Resultado :     (03/09/2011 7:57:19) + (4 horas) = (03/09/2011 11:57:19)</pre>

### 1.4.5 DateDiff

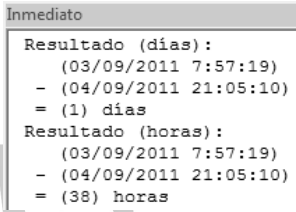
Devuelve el intervalo de tiempo existente entre dos fechas expresado en la unidad de medida que le indiquemos.

Sintaxis:

DateDiff(intervalo, fechaIni, fechaFin[, pDiaSem[, pSemAño]])
---

Argumento	Comentario
<b>Intervalo</b>	Indica qué tipo de información es la que se tiene que añadir (días, meses, horas, etc.). Ver tabla <b>Intervalos de tiempo</b> .
<b>fechaIni</b>	Fecha inicial del rango sobre el que se calcula el intervalo de tiempo.
<b>fechaFin</b>	Fecha final del rango sobre el que se calcula el intervalo de tiempo.
<b>pDiaSem</b>	Indica cual es el primer día de la semana. Por defecto se entiende que es el domingo. Ver tabla <b>Días de la semana</b> .
<b>pSemAño</b>	Indica cual es la primera semana del año. Por defecto se entiende que es aquella en la que se halla el día 1.

Ejemplo:

Código	Resultado
<pre> Sub muestra_DateDiff()     Dim fechaIni, fechaFin, dias, horas, res As String     fechaIni = "03/09/2011 7:57:19"     fechaFin = "04/09/2011 21:05:10"     res = DateDiff("d", fechaIni, fechaFin)     Debug.Print "Resultado (días):"     Debug.Print "    (" &amp; fechaIni &amp; ")"     Debug.Print " - (" &amp; fechaFin &amp; ")"     Debug.Print " = (" &amp; res &amp; ") días"     res = DateDiff("h", fechaIni, fechaFin)     Debug.Print "Resultado (horas):"     Debug.Print "    (" &amp; fechaIni &amp; ")"     Debug.Print " - (" &amp; fechaFin &amp; ")"     Debug.Print " = (" &amp; res &amp; ") horas" End Sub </pre>	 <pre> Inmediato Resultado (días):     (03/09/2011 7:57:19)  - (04/09/2011 21:05:10)  = (1) días Resultado (horas):     (03/09/2011 7:57:19)  - (04/09/2011 21:05:10)  = (38) horas </pre>



### 1.4.6 DatePart

Devuelve una parte de una fecha en función de la información solicitada (semana, trimestre, etc.).

Sintaxis:

```
DatePart(información, fecha[,pDiaSem[, pSemAño]])
```

Argumento	Comentario
información	Indica qué tipo de información es la que se tiene que añadir (días, meses, horas, etc.). Ver tabla <b>Intervalos de tiempo</b> .
fecha	Fecha a tratar.
pDiaSem	Indica cual es el primer día de la semana. Por defecto se entiende que es el domingo. Ver tabla <b>Días de la semana</b> .
pSemAño	Indica cual es la primera semana del año. Por defecto se entiende que es aquella en la que se halla el día 1.

Ejemplos:

Código	Resultado
DatePart("yyyy", "03/09/2011 7:57:19")	Res (2011)
DatePart("q", "03/09/2011 7:57:19")	Res (3)
DatePart("m", "03/09/2011 7:57:19")	Res (9)
DatePart("y", "03/09/2011 7:57:19")	Res (246)
DatePart("d", "03/09/2011 7:57:19")	Res (3)
DatePart("w", "03/09/2011 7:57:19")	Res (7)
DatePart("ww", "03/09/2011 7:57:19")	Res (36)
DatePart("h", "03/09/2011 7:57:19")	Res (7)
DatePart("n", "03/09/2011 7:57:19")	Res (57)
DatePart("s", "03/09/2011 7:57:19")	Res (19)

### 1.4.7 DateSerial

Devuelve una fecha a partir de la concatenación de año, mes y día.

Sintaxis:

DateSerial(año, mes, día)
---------------------------

Argumento	Comentario
año	Año a utilizar para formar la fecha.
mes	Mes a utilizar para formar la fecha.
día	Día a utilizar para formar la fecha.

Ejemplo:

Código	Resultado
<pre>Sub muestra_DateSerial()   Dim fecha As String   fecha = DateSerial("2011", "09", "03")   Debug.Print "Res (" &amp; fecha &amp; ")" End Sub</pre>	

### 1.4.8 DateValue

Devuelve una fecha a partir de una cadena que contiene una expresión de tipo fecha.

Sintaxis:

DateValue(fecha)
------------------

Argumento	Comentario
fecha	Expresión a la que se desea obtener su valor

Ejemplo:

Código	Resultado
<pre>Sub muestra_DateValue()     Debug.Print "Res (" &amp; DateValue("03/09/2011")     &amp; ")" End Sub</pre>	<div>Inmediato</div> <div>Res (03/09/2011)</div>

### 1.4.9 Day

Obtiene el día de una fecha.

Sintaxis:

Day (fecha)
-------------

Argumento	Comentario
fecha	Expresión a la que se desea obtener el día.

Ejemplo:

Código	Resultado
<pre>Sub muestra_Day()     Debug.Print "Res (" &amp; Day("03/09/2011") &amp; ")" End Sub</pre>	<div>Inmediato</div> <div>Res (3)</div>

### 1.4.10 Hour

Obtiene la hora.

Sintaxis:

Hour (hora)
-------------

Argumento	Comentario
hora	Expresión de la que se desea obtener la hora.

Ejemplo:

Código	Resultado
<pre>Sub muestra_Hour()     Debug.Print "Res (" &amp; Hour("07:35:15") &amp; ")" End Sub</pre>	<div>Inmediato</div> <div>Res (7)</div>

### 1.4.11 Minute

Obtiene los minutos.

Sintaxis:

Minute(hora)
--------------

Argumento	Comentario
Hora	Expresión de la que se desea obtener los minutos.

Ejemplo:

Código	Resultado
<pre>Sub muestra_Minute()     Debug.Print "Res (" &amp; Minute("07:35:15") &amp; ")" End Sub</pre>	<div>Inmediato</div> <div>Res (35)</div>

### 1.4.12 Month

Obtiene el mes de una fecha.

Sintaxis:

```
Month(fecha)
```

Argumento	Comentario
fecha	Expresión de la que se desea obtener el mes.

Ejemplo:

Código	Resultado
<pre>Sub muestra_Month()   Debug.Print "Res (" &amp; Month("03/09/2011") &amp; ")" End Sub</pre>	<div>Inmediato</div> <div>Res (9)</div>

### 1.4.13 MonthName

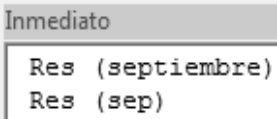
Devuelve un string con el nombre del mes indicado.

Sintaxis:

```
MonthName(numMes [,abreviado])
```

Argumento	Comentario
numMes	Número de mes del que se desea obtener el nombre.
abreviado	Indica si se desea abreviar o no ( <b>True</b> o <b>False</b> ). Por defecto es <b>False</b> y no se abrevia.

Ejemplo:

Código	Resultado
<pre>Sub muestra_MonthName()     Dim s As String     Dim res As String     s = "03/09/2011"     res = MonthName(DatePart("m", s))     Debug.Print "Res (" &amp; res &amp; ")"     res = MonthName(DatePart("m", s), True)     Debug.Print "Res (" &amp; res &amp; ")" End Sub</pre>	

### 1.4.14 Now

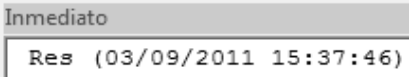
Devuelve la fecha y hora actuales.

Sintaxis:

Now
-----

Argumento	Comentario
ninguno	No tiene.

Ejemplo:

Código	Resultado
<pre>Sub muestra_Now()     Debug.Print "Res (" &amp; Now &amp; ")" End Sub</pre>	

Ra-Ma®

### 1.4.15 Second

Obtiene los segundos.

Sintaxis:

Second (hora)

Argumento	Comentario
Hora	Expresión de la que se desea obtener los segundos.

Ejemplo:

Código	Resultado
<pre>Sub muestra_Second()     Debug.Print "Res (" &amp; Second("07:35:15") &amp; ") " End Sub</pre>	<div>Inmediato</div> <div>Res (15)</div>

### 1.4.16 Time


Devuelve la hora actual del sistema.

Sintaxis:

Time

Argumento	Comentario
ninguno	No tiene.

Ejemplo:

Código	Resultado
<pre>Sub muestra_Time()   Dim f As Date   f = Time ' 07:57:19   Debug.Print "Resultado (" &amp; f &amp; ")" End Sub</pre>	

### 1.4.17 Timer

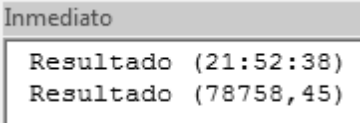
Devuelve el número de segundos transcurridos desde la medianoche.

Sintaxis:

Timer
-------

Argumento	Comentario
ninguno	No tiene.

Ejemplo:

Código	Resultado
<pre>Sub muestra_Timer()   Dim f As String   f = Timer ' 21:52:38   Debug.Print "Resultado (" &amp; Time &amp;   ")"   Debug.Print "Resultado (" &amp; Timer &amp;   ")" End Sub</pre>	

Ra-Ma®



### 1.4.18 TimeSerial

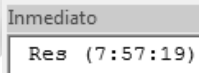
Compone una cadena con una hora a partir de los parámetros indicados.

Sintaxis:

```
TimeSerial(hora, minuto, segundo)
```

Argumento	Comentario
hora	Hora a incluir en la expresión de hora.
minuto	Minuto a incluir en la expresión.
segundo	Segundos a incluir en la expresión.

Ejemplo:

Código	Resultado
<pre>Sub muestra_TimeSerial()     Debug.Print "Res (" &amp; TimeSerial("07", "57",     "19") &amp; ")" End Sub</pre>	

### 1.4.19 WeekDay

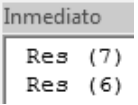
Devuelve el día de la semana correspondiente.

Sintaxis:

```
Weekday(fecha [, pDiaSem])
```

Argumento	Comentario
fecha	Fecha a la que se desea averiguar el día de la semana que representa.
pdiaSem	Indica cual es el primer día de la semana. Ver tabla <b>Días de la semana.</b>

Ejemplo:

Código	Resultado
<pre>Sub muestra_WeekDay()     Debug.Print "Res (" &amp; Weekday("03/09/2011") &amp;     ") "     Debug.Print "Res (" &amp; Weekday("03/09/2011",     vbMonday) &amp; ") " End Sub</pre>	

### 1.4.20 WeekdayName

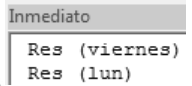
Devuelve el nombre de un día de la semana.

Sintaxis:

WeekdayName(día, abreviar, pDiaSem)
-------------------------------------

Argumento	Comentario
día	Día al que se desea obtener al nombre.
abreviar	Indica si el nombre ha de estar abreviado (True) o no (False). Por defecto no se abrevia.
pDiaSem	Indica cual es el primer día de la semana. Ver tabla <b>Días de la semana</b> .

Ejemplo:

Código	Resultado
<pre>Sub muestra_WeekdayName()     Debug.Print "Res (" &amp; WeekdayName(5) &amp; ") "     Debug.Print "Res (" &amp; WeekdayName(2, True,     vbSunday) &amp; ") " End Sub</pre>	

### 1.4.21 Year

Devuelve el año de una fecha determinada.

Sintaxis:

Year (fecha)
--------------

Argumento	Comentario
<b>fecha</b>	Expresión de la que se desea obtener el año.

Ejemplo:

Código	Resultado
<pre>Sub muestra_Year()   Debug.Print "Res (" &amp; Year("03/09/2011") &amp; ") " End Sub</pre>	<div>Inmediato</div> <div>Res (2011)</div>

## 1.5 FUNCIONES MATEMÁTICAS

Función	Descripción
Abs	Devuelve el valor absoluto del número indicado.
Atn	Devuelve la arcotangente del número indicado.
Cos	Devuelve el coseno del número (ángulo) indicado.
Exp	Devuelve el número 'e' elevado a una potencia.
Log	Devuelve el logaritmo de un número.
Randomize	Inicializa el generador de números aleatorios.
Rnd	Devuelve un valor comprendido entre 0 y 1.

Round	Devuelve un número redondeado a un número especificado de decimales.
Sgn	Devuelve un valor que indica el signo de un determinado valor.
Sin	Devuelve el seno del número (ángulo) indicado.
Sqr	Devuelve la raíz cuadrada del número indicado.
Tan	Devuelve la tangente del número (ángulo) indicado.

### 1.5.1 Abs

Devuelve el valor absoluto del número indicado.

Sintaxis:

Abs (número)
--------------

Argumento	Comentario
<b>número</b>	Expresión al que se elimina el signo.

Ejemplo:

Código	Resultado			
<pre>Sub muestra_Abs()     Debug.Print "Res (" &amp; Abs(123.45) &amp; ")"     Debug.Print "Res (" &amp; Abs(-123.45) &amp; ")" End Sub</pre>	<table><tr><th>Inmediato</th></tr><tr><td>Res (123,45)</td></tr><tr><td>Res (123,45)</td></tr></table>	Inmediato	Res (123,45)	Res (123,45)
Inmediato				
Res (123,45)				
Res (123,45)				

### 1.5.2 Atn

Devuelve la arcotangente del número indicado.

Sintaxis:

Atn (número)
--------------

Argumento	Comentario
<b>número</b>	Expresión a la que se le calcula la arcotangente.

Ejemplo:

Código	Resultado
<pre>Sub muestra_Atn()     Debug.Print "Res (" &amp; Atn(123.45) &amp; ")"     Debug.Print "Res (" &amp; Atn(1) &amp; ")"     Debug.Print "Res (" &amp; 4 * Atn(1) &amp; ")" ' Pi End Sub</pre>	<div>Inmediato</div> <pre>Res (1,56269605843965) Res (0,785398163397448) Res (3,14159265358979)</pre>

### 1.5.3 Cos

Devuelve el coseno del número (ángulo expresado en radianes) indicado.

Sintaxis:

Cos (número)
--------------

Argumento	Comentario
<b>número</b>	Expresión a la que se le calcula el coseno.

Ejemplo:

Código	Resultado
<pre>Sub muestra_Cos()     Dim Pi As Double     Pi = (4 * Atn(1))     Debug.Print "Res ( 0°) -&gt; (" &amp; Round(Cos(0 * Pi), 5) &amp; ")"     Debug.Print "Res ( 90°) -&gt; (" &amp; Round(Cos(Pi / 2), 5) &amp; ")"     Debug.Print "Res (180°) -&gt; (" &amp; Round(Cos(1 * Pi), 5) &amp; ")"     Debug.Print "Res (270°) -&gt; (" &amp; Round(Cos(1.5 * Pi), 5) &amp; ")"     Debug.Print "Res (360°) -&gt; (" &amp; Round(Cos(2 * Pi), 5) &amp; ")" End Sub</pre>	<div>Inmediato</div> <pre>Res ( 0°) -&gt; (1) Res ( 90°) -&gt; (0) Res (180°) -&gt; (-1) Res (270°) -&gt; (0) Res (360°) -&gt; (1)</pre>

### 1.5.4 Exp

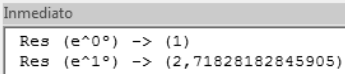
Devuelve el número ‘e’ elevado a una potencia.

Sintaxis:

Exp (número)
--------------

Argumento	Comentario
<b>número</b>	Expresión a la que elevar el número ‘e’.

Ejemplo:

Código	Resultado
<pre>Sub muestra_Exp()     Debug.Print "Res (e^0°) -&gt; (" &amp; Exp(0)     &amp; ")"     Debug.Print "Res (e^1°) -&gt; (" &amp; Exp(1)     &amp; ")" End Sub</pre>	

### 1.5.5 Log

Devuelve el logaritmo natural de un número. El logaritmo de un número en una base determinada, es el valor al que hay que elevar dicha base para obtener el número. Por ejemplo el logaritmo de 100 en base 10 es 2.

Sintaxis:

Log (número)
--------------

Argumento	Comentario
<b>número</b>	Expresión a la que calcular el logaritmo natural.

Ejemplo:

Código	Resultado
<pre>Sub muestra_Log()     Debug.Print "Res Log(2.71828182845905) 'e' -&gt; ("     &amp; Log(2.71828182845905) &amp; ")" End Sub</pre>	

Para obtener el logaritmo de un número (x) en una base determinada (BaseN) podemos emplear la siguiente fórmula:

$\text{LogBaseN}(x) = \text{Log}(x) / \text{Log}(\text{BaseN})$
---

Así pues, para calcular el logaritmo de 100 en base 10 y confirmar el ejemplo comentado en la explicación de esta función podríamos escribir el siguiente ejemplo:

Código	Resultado
<pre>Sub muestra_LogDecimal()     Dim ld As Double     ld = Log(100) / Log(10)     Debug.Print "Log decimal de 100 =&gt; (" &amp;     ld &amp; ")" End Sub</pre>	

## 1.5.6 Randomize

Inicializa el generador de números aleatorios con un valor de semilla indicado. Si no se indica ningún valor, la semilla se basa en el reloj del sistema.

Sintaxis:

Randomize [número]
--------------------

Argumento	Comentario
<b>número</b>	Semilla a utilizar para inicializar la secuencia de números aleatorios.

Ejemplo:

Código	Resultado
<pre>Sub muestra_Randomize()   Dim i, num As Double   Randomize   For i = 1 To 5     Debug.Print "(" &amp; Rnd &amp; ")"   Next i End Sub</pre>	<div>Inmediato</div> <div>(0,6272395)</div> <div>(0,5594857)</div> <div>(0,2529206)</div> <div>(0,6395959)</div> <div>(0,26613)</div>

### 1.5.7 Rnd

Devuelve un valor comprendido entre 0 y 1.

Sintaxis:

Rnd [(número)]

Argumento	Comentario										
número	Se trata de un argumento opcional y, en función de su valor, provoca que la función devuelva unos números aleatorios u otros.										
	Hay que considerar lo siguiente:										
	<table><tr><th>Valor</th><th>Comentario</th></tr><tr><td>&lt; 0</td><td>Se repite el mismo número utilizando el valor como valor de semilla.</td></tr><tr><td>&gt; 0</td><td>Devuelve el siguiente número aleatorio.</td></tr><tr><td>= 0</td><td>Devuelve el último número aleatorio generado.</td></tr><tr><td>ninguno</td><td>Devuelve el siguiente número aleatorio.</td></tr></table>	Valor	Comentario	< 0	Se repite el mismo número utilizando el valor como valor de semilla.	> 0	Devuelve el siguiente número aleatorio.	= 0	Devuelve el último número aleatorio generado.	ninguno	Devuelve el siguiente número aleatorio.
	Valor	Comentario									
	< 0	Se repite el mismo número utilizando el valor como valor de semilla.									
	> 0	Devuelve el siguiente número aleatorio.									
= 0	Devuelve el último número aleatorio generado.										
ninguno	Devuelve el siguiente número aleatorio.										
Si se utiliza un valor de semilla concreto, siempre se generará la misma secuencia de números aleatorios.											



Ejemplo:

Código	Resultado
<pre>Sub muestra_Rnd()   Dim i As Double   For i = 1 To 5     Debug.Print "(" &amp; Rnd &amp; ")"   Next i   Debug.Print "(" &amp; Rnd(0) &amp; ")" ' Repite último número End Sub</pre>	<div>Inmediato</div> <pre>(0,7594203) (0,3724771) (0,7478452) (0,6656253) (0,7181321) (0,7181321)</pre>

El siguiente ejemplo muestra la obtención de 5 números aleatorios comprendidos entre 0 y 100:

Código	Resultado
<pre>Sub muestra_Rnd_Rango()   Dim i, ini, fin, res As Double   ini = 0: fin = 100   ' Busca 5 números aleatorios entre ini y fin   For i = 1 To 5     res = Int((fin - ini + 1) * Rnd + ini)     Debug.Print "(" &amp; res &amp; ")"   Next i End Sub</pre>	<div>Inmediato</div> <pre>(6) (30) (20) (37) (15)</pre>

## 1.5.8 Round

Devuelve un número redondeado a un número especificado de decimales. El redondeo se realiza de forma que si el dígito a redondear es menor a 5, se devuelve un 0 y el dígito existente inmediatamente a su izquierda se queda intacto. Si el dígito a redondear es mayor o igual a 5, se devuelve un 0 también pero, el dígito que posee inmediatamente a la izquierda se incrementa en una unidad.

Sintaxis:

Round(expresión [, numDec])
-----------------------------

Argumento	Comentario
<b>expresión</b>	Expresión a redondear.
<b>numDec</b>	Número de decimales a usar para el redondeo. Si se omite este argumento, la función devuelve un entero.

Ejemplo:

Código	Resultado
<pre>Sub muestra_Round()   Dim numero As Double   numero = 12345.6789   Debug.Print "(" &amp; Round(numero) &amp; ")"   Debug.Print "(" &amp; Round(numero, 1) &amp; ")"   Debug.Print "(" &amp; Round(numero, 2) &amp; ")" End Sub</pre>	<div>Inmediato</div> <pre>(12346) (12345,7) (12345,68)</pre>

### 1.5.9 Sgn

Devuelve un valor que indica el signo de una determinada expresión. Los valores devueltos pueden ser:

Valor	Comentario
1	Indica que el resultado de la expresión es mayor que 0.
0	Indica que el resultado de la expresión es igual a 0.
-1	Indica que el resultado de la expresión es menor que 0.

Sintaxis:

Sgn (expresión)
-----------------

Argumento	Comentario
<b>expresión</b>	Expresión a analizar.

Ejemplo:

Código	Resultado
<pre>Sub muestra_Sgn()   Debug.Print "(" &amp; Sgn(12345.6789) &amp; ")"   Debug.Print "(" &amp; Sgn(0) &amp; ")"   Debug.Print "(" &amp; Sgn(-12345.6789) &amp; ")" End Sub</pre>	<div>Inmediato</div> <pre>(1) (0) (-1)</pre>

### 1.5.10 Sin

Devuelve el seno del número (ángulo expresado en radianes) indicado.

Sintaxis:

Sin(número)
-------------

Argumento	Comentario
número	Expresión a la que se le calcula el seno.

Ejemplo:

Código	Resultado						
<pre>Sub muestra_Sin()     Dim Pi As Double     Pi = (4 * Atn(1))     Debug.Print "Res ( 0°) -&gt; (" &amp; Round(Sin(0 * Pi), 5) &amp; ")"     Debug.Print "Res ( 90°) -&gt; (" &amp; Round(Sin(Pi / 2), 5) &amp; ")"     Debug.Print "Res (180°) -&gt; (" &amp; Round(Sin(1 * Pi), 5) &amp; ")"     Debug.Print "Res (270°) -&gt; (" &amp; Round(Sin(1.5 * Pi), 5) &amp; ")"     Debug.Print "Res (360°) -&gt; (" &amp; Round(Sin(2 * Pi), 5) &amp; ")" End Sub</pre>	<table><tr><th>Inmediato</th></tr><tr><td>Res ( 0°) -&gt; (0)</td></tr><tr><td>Res ( 90°) -&gt; (1)</td></tr><tr><td>Res (180°) -&gt; (0)</td></tr><tr><td>Res (270°) -&gt; (-1)</td></tr><tr><td>Res (360°) -&gt; (0)</td></tr></table>	Inmediato	Res ( 0°) -> (0)	Res ( 90°) -> (1)	Res (180°) -> (0)	Res (270°) -> (-1)	Res (360°) -> (0)
Inmediato							
Res ( 0°) -> (0)							
Res ( 90°) -> (1)							
Res (180°) -> (0)							
Res (270°) -> (-1)							
Res (360°) -> (0)							

### 1.5.11 Sqr

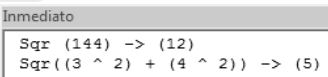
Devuelve la raíz cuadrada del número indicado.

Sintaxis:

Sqr(número)
-------------

Argumento	Comentario
<b>número</b>	Expresión a la que se le calcula la raíz cuadrada.

Ejemplo:

Código	Resultado
<pre>Sub muestra_Sqr()     Debug.Print "Sqr (144) -&gt; (" &amp; Sqr(144)     &amp; ")"     Debug.Print "Sqr((3 ^ 2) + (4 ^ 2)) -&gt;     (" -     &amp; Sqr((3 ^ 2) + (4 ^ 2)) &amp; ")" End Sub</pre>	

### 1.5.12 Tan

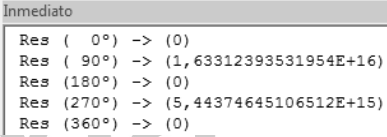
Devuelve la tangente del número (ángulo expresado en radianes) indicado.

Sintaxis:

Tan(número)
-------------

Argumento	Comentario
<b>número</b>	Expresión a la que se le calcula la tangente.

Ejemplo:

Código	Resultado
<pre>Sub muestra_Tan()     Dim Pi As Double     Pi = (4 * Atn(1))     Debug.Print "Res ( 0°) -&gt; (" &amp;     Round(Tan(0 * Pi), 5) &amp; ")"     Debug.Print "Res ( 90°) -&gt; (" &amp;     Round(Tan(Pi / 2), 5) &amp; ")"     Debug.Print "Res (180°) -&gt; (" &amp;     Round(Tan(1 * Pi), 5) &amp; ")"     Debug.Print "Res (270°) -&gt; (" &amp;     Round(Tan(1.5 * Pi), 5) &amp; ")"     Debug.Print "Res (360°) -&gt; (" &amp;     Round(Tan(2 * Pi), 5) &amp; ")" End Sub</pre>	

## 1.6 FUNCIONES FINANCIERAS

Función	Comentario
DDB	Calcula la amortización de un activo dado un período.
FV	Calcula el valor futuro de una anualidad en función de unos pagos periódicos y un tipo de interés constantes.
IPmt	Calcula el pago de intereses durante un periodo determinado de una anualidad en función de unos pagos periódicos y un tipo de interés constantes.
IRR	Calcula la tasa interna de retorno de una serie de flujos de efectivo periódicos (pagos y cobros).
MIRR	Calcula la tasa interna de retorno modificada para una serie de flujos de efectivo periódicos (pagos y cobros).
NPer	Calcula el número de periodos de una anualidad en función de unos pagos periódicos y un tipo de interés constantes.
NPV	Calcula el valor actual neto de una inversión basada en una serie de flujos de efectivo periódicos (pagos y cobros) y un tipo de descuento.
Pmt	Calcula el pago de una anualidad en función de unos pagos periódicos y un tipo de interés constantes.
PPmt	Calcula el pago nominal correspondiente a un periodo determinado de una anualidad en función de unos pagos periódicos y un tipo de interés constantes.
PV	Calcula el valor actual de una anualidad basada en pagos periódicos constantes que se van a realizar en el futuro, con un tipo de interés también constante.
Rate	Calcula el tipo de interés por periodo de una anualidad.
SLN	Calcula la depreciación lineal de un bien durante un único periodo.
SYD	Calcula la depreciación de un bien en dígitos de suma de años durante un periodo especificado.

## 1.6.1 DDB

Calcula la amortización de un activo dado un período

Sintaxis:

```
DDB(coste, valResidual, duración, perCalDeprec [, factor])
```

Argumento	Comentario
<b>coste</b>	Coste inicial del bien.
<b>valResidual</b>	Valor del bien al final de su vida útil.
<b>duración</b>	Duración de la vida útil del bien.
<b>perCalDeprec</b>	Periodo a considerar para la depreciación del bien.
<b>factor</b>	Tasa de disminución de saldo. Por defecto se utiliza 2 (método de doble disminución).

Ejemplo:

Código	Resultado										
<pre>Sub muestra_DDB()     Dim depr, coste, valResidual, duracion,     perCalDeprec, valor     Dim mask, s, cadena As String      mask = "#,##0.00"      coste = 10000     valResidual = 10     duracion = 5     perCalDeprec = 3      depr = DDB(coste, valResidual, duracion,     perCalDeprec)      s = "Coste": valor = coste: GoSub Fmt     s = "Valor residual": valor = valResidual: GoSub Fmt</pre>	<div> <div>Inmediato</div> <table> <tr> <td>Coste</td><td>: 10.000,00</td></tr> <tr> <td>Valor residual</td><td>: 10,00</td></tr> <tr> <td>Duración</td><td>: 5,00</td></tr> <tr> <td>Per.deprec.</td><td>: 3,00</td></tr> <tr> <td>DEPRECIACION</td><td>: 1.440,00</td></tr> </table> </div>	Coste	: 10.000,00	Valor residual	: 10,00	Duración	: 5,00	Per.deprec.	: 3,00	DEPRECIACION	: 1.440,00
Coste	: 10.000,00										
Valor residual	: 10,00										
Duración	: 5,00										
Per.deprec.	: 3,00										
DEPRECIACION	: 1.440,00										

```

s = "Duración": valor = duracion: GoSub Fmt
s = "Per.deprec. ": valor = perCalDeprec:
GoSub Fmt
s = "DEPRECIACION ": valor = depr: GoSub Fmt

Exit Sub
Fmt:
cadena = Format(s, "!@@@@@@@@@@@@@")
Debug.Print cadena & " : " _
& Format(valor, mask), "@@@@@@@@@")
Return
End Sub

```

## 1.6.2 FV

Calcula el valor futuro de una anualidad en función de unos pagos periódicos y un tipo de interés constantes.

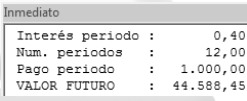
Sintaxis:

```
FV(intPer, numPer, pago[, valorActual[, cuandoVencenPagos]])
```

Argumento	Comentario
<b>intPer</b>	Interés por período. Por ejemplo un 20% anual correspondería a $0,2/12 = 0,016$ .
<b>numPer</b>	Número de periodos de pago.
<b>pago</b>	Pago a realizar en cada periodo.
<b>valorActual</b>	Saldo actual o importe inicial.
<b>cuanadoVencenPagos</b>	Flag que indica si los pagos vencen al final del período (0) o al inicio (1).

Ra-Ma®

Ejemplo:

Código	Resultado
<pre> Sub muestra_FV()     Dim valf, intPer, numPer, Pago, valorActual,     cuandoVencenPagos, valor     Dim mask, s, cadena As String      mask = "#,##0.00"      intPer = 0.4     numPer = 12     Pago = 1000     valorActual = 20000     cuandoVencenPagos = 1      valf = FV(intPer / 12, numPer, -Pago, -     valorActual, cuandoVencenPagos)      s = "Interés periodo": valor = intPer: GoSub Fmt     s = "Num. periodos": valor = numPer: GoSub Fmt     s = "Pago periodo": valor = Pago: GoSub Fmt     s = "VALOR FUTURO ": valor = valf: GoSub Fmt      Exit Sub Fmt:     cadena = Format(s, "!@@@@@@@@@@@@@@@@")     Debug.Print cadena &amp; " : " _     &amp; Format(Format(valor, mask), "@@@@@@@@")     Return End Sub </pre>	 <pre> Inmediato Interés periodo :      0,40 Num. periodos   :      12,00 Pago periodo    :    1.000,00 VALOR FUTURO    :   44.588,45 </pre>

### 1.6.3 IPmt

Calcula el pago de intereses durante un periodo determinado de una anualidad en función de unos pagos periódicos y un tipo de interés constantes.

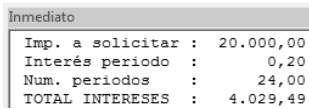
Sintaxis:

```
IPmt(intPer, perPago, numPer, valorActual[, valorFuturo[,
cuandoVencenPagos]])
```



Argumento	Comentario
<b>intPer</b>	Interés por período. Por ejemplo un 20% anual correspondería a $0,2/12 = 0,016$ .
<b>perPago</b>	Período de pago comprendido entre 1 y numPer.
<b>numPer</b>	Número de periodos de pago.
<b>valorActual</b>	Saldo actual o importe inicial.
<b>valorFuturo</b>	Valor futuro que se desea tener una vez realizado el pago final.
<b>cuandoVencenPagos</b>	Flag que indica si los pagos vencen al final del período (0) o al inicio (1).

Ejemplo:3

Código	Resultado								
<pre> Sub muestra_IPmt()     Dim valInt, intPer, perPago, numPer As Double     Dim valorActual, valorFuturo, cuandoVencenPagos     Dim totalInt, valor As Double     Dim mask, s, cadena As String      mask = "#,##0.00"      intPer = 0.2     numPer = 24     valorActual = 20000     valorFuturo = 0     cuandoVencenPagos = 1     For perPago = 1 To numPer         valInt = IPmt(intPer / 12, perPago, numPer, -valorActual, valorFuturo, cuandoVencenPagos)          totalInt = totalInt + valInt     Next perPago      s = "Imp. a solicitar": valor = valorActual: GoSub Fmt </pre>	 <p>Immediate</p> <table border="1"> <tr> <td>Imp. a solicitar :</td> <td>20.000,00</td> </tr> <tr> <td>Interés periodo :</td> <td>0,20</td> </tr> <tr> <td>Num. periodos :</td> <td>24,00</td> </tr> <tr> <td>TOTAL INTERESES :</td> <td>4.029,49</td> </tr> </table>	Imp. a solicitar :	20.000,00	Interés periodo :	0,20	Num. periodos :	24,00	TOTAL INTERESES :	4.029,49
Imp. a solicitar :	20.000,00								
Interés periodo :	0,20								
Num. periodos :	24,00								
TOTAL INTERESES :	4.029,49								

```

s = "Interés periodo": valor = intPer:
GoSub Fmt
s = "Num. periodos": valor = numPer:
GoSub Fmt
s = "TOTAL INTERESES ": valor =
totalInt: GoSub Fmt

Exit Sub
Fmt:
cadena = Format(s, "!@@@@@@@@@@@@@@@@")
Debug.Print cadena & " : "
& Format(Format(valor, mask),
"@@@@@@@@@@@@")
Return
End Sub

```

### 1.6.4 IRR

Calcula la tasa interna de retorno de una serie de flujos de efectivo periódicos (pagos y cobros).

Sintaxis:

```
IRR(valores() [, estIRR])
```

Argumento	Comentario
<b>valores()</b>	Valores de los flujos de efectivo. Los flujos negativos representan costes. Los positivos representan los cobros.
<b>estIRR</b>	Tasa interna de retorno estimada.

Ejemplo:

Código	Resultado												
<pre> Sub muestra_IRR() Dim valores(4) As Double Dim estIRR, intRet, valor As Double Dim mask, s, cadena As String Dim i As Integer mask = "#,##0.00" estIRR = 0.1 valores(0) = -15000 </pre>	<p>(R)</p> <table> <tr> <td>Inmediato</td><td></td></tr> <tr> <td>Valor (0)</td><td>: -15.000,00</td></tr> <tr> <td>Valor (1)</td><td>: 6.500,00</td></tr> <tr> <td>Valor (2)</td><td>: 8.000,00</td></tr> <tr> <td>Valor (3)</td><td>: 9.500,00</td></tr> <tr> <td>Tasa int. retorno</td><td>: 0,26</td></tr> </table>	Inmediato		Valor (0)	: -15.000,00	Valor (1)	: 6.500,00	Valor (2)	: 8.000,00	Valor (3)	: 9.500,00	Tasa int. retorno	: 0,26
Inmediato													
Valor (0)	: -15.000,00												
Valor (1)	: 6.500,00												
Valor (2)	: 8.000,00												
Valor (3)	: 9.500,00												
Tasa int. retorno	: 0,26												

```

valores(1) = 6500
valores(2) = 8000
valores(3) = 9500
intRet = IRR(valores, estIRR)
For i = 0 To UBound(valores) - 1
    s = "Valor (" & i & ")": valor =
valores(i)
    GoSub Fmt
Next i
s = "Tasa int. retorno": valor = intRet:
GoSub Fmt
Exit Sub
Fmt:
cadena = Format(s, "!" & String(18,
"@"))
Debug.Print cadena & " : " _
& Format(valor, mask),
"@@@@@@@@@@")
Return
End Sub

```

### 1.6.5 MIRR

Calcula la tasa interna de retorno modificada para una serie de flujos de efectivo periódicos (pagos y cobros).

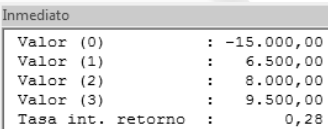
Sintaxis:

```
MIRR(valores(), intPagado, intRecibido)
```

Argumento	Comentario
<b>valores()</b>	Valores de los flujos de efectivo. Los flujos negativos representan costes. Los negativos representan los cobros.
<b>intPagado</b>	Interés pagado como coste de financiación.
<b>intRecibido</b>	Interés recibido como ganancia por la inversión en efectivo.

Ra-Ma®

Ejemplo:

Código	Resultado
<pre> Sub muestra_MIRR()     Dim valores(4) As Double     Dim intPagado, intRecibido, intRet, valor As Double     Dim mask, s, cadena As String     Dim i As Integer     mask = "#,##0.00"     intPagado = 0.1     intRecibido = 0.3     valores(0) = -15000     valores(1) = 6500     valores(2) = 8000     valores(3) = 9500     intRet = MIRR(valores, intPagado, intRecibido)     For i = 0 To UBound(valores) - 1         s = "Valor (" &amp; i &amp; ")": valor = valores(i)         GoSub Fmt     Next i     s = "Tasa int. retorno": valor = intRet: GoSub Fmt     Exit Sub Fmt:     cadena = Format(s, "!" &amp; String(18, "@"))     Debug.Print cadena &amp; " : "         &amp; Format(Format(valor, mask), "@@@@@@@@@@@@")     Return End Sub </pre>	 <pre> Inmediato Valor (0)      : -15.000,00 Valor (1)      :  6.500,00 Valor (2)      :  8.000,00 Valor (3)      :  9.500,00 Tasa int. retorno : 0,28 </pre>

## 1.6.6 NPer

Calcula el número de periodos de una anualidad en función de unos pagos periódicos y un tipo de interés constantes.

Sintaxis:

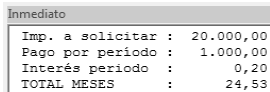
```

NPer(intPer, pagoPer, valorActual[, valorFuturo[,
cuandoVencenPagos]])

```

Argumento	Comentario
<b>intPer</b>	Interés por período. Por ejemplo un 20% anual correspondería a $0,2/12 = 0,016$ .
<b>pagoPer</b>	Pago a realizar en cada periodo.
<b>valorActual</b>	Saldo actual o importe inicial.
<b>valorFuturo</b>	Valor futuro que se desea tener una vez realizado el pago final.
<b>cuandoVencenPagos</b>	Flag que indica si los pagos vencen al final del período (0) o al inicio (1).

Ejemplo:

Código	Resultado
<pre> Sub muestra_NPer()     Dim intPer, pagoPer As Double     Dim valorActual, valorFuturo, cuandoVencenPagos     Dim nMeses, valor As Double     Dim mask, s, cadena As String      mask = "#,##0.00"      intPer = 0.2     pagoPer = 1000     valorActual = 20000     valorFuturo = 0     cuandoVencenPagos = 1     nMeses = NPer(intPer / 12, -pagoPer, valorActual, cuandoVencenPagos)     s = "Imp. a solicitar": valor = valorActual: GoSub Fmt     s = "Pago por período": valor = pagoPer: GoSub Fmt     s = "Interés periodo": valor = intPer: GoSub Fmt     s = "TOTAL MESES ": valor = nMeses: GoSub Fmt      Exit Sub Fmt:     cadena = Format(s, "!@@@@@@@@@@@@@@@@")     Debug.Print cadena &amp; " : "     &amp; Format(Format(valor, mask), "@@@@@@@@")     Return End Sub </pre>	 <pre> Immediate Imp. a solicitar : 20.000,00 Pago por periodo : 1.000,00 Interés periodo : 0,20 TOTAL MESES : 24,53 </pre>

### 1.6.7 NPV

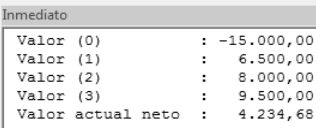
Calcula el valor actual neto de una inversión basada en una serie de flujos de efectivo periódicos (pagos y cobros) y un tipo de descuento.

Sintaxis:

NPV(tipoDes, valores())
-------------------------

Argumento	Comentario
<b>tipoDes</b>	Tipo de descuento a lo largo del período.
<b>valores()</b>	Valores de los flujos de efectivo. Los flujos negativos representan costes. Los negativos representan los cobros.

Ejemplo:

Código	Resultado
<pre> Sub muestra_NPV()     Dim valores(4) As Double     Dim tipoDes, valorNeto, valor As Double     Dim mask, s, cadena As String     Dim i As Integer     mask = "#,##0.00"     tipoDes = 0.1     valores(0) = -15000     valores(1) = 6500     valores(2) = 8000     valores(3) = 9500     valorNeto = NPV(tipoDes, valores)     For i = 0 To UBound(valores) - 1         s = "Valor (" &amp; i &amp; ")": valor =     valores(i)         GoSub Fmt     Next i     s = "Valor actual neto": valor =     valorNeto: GoSub Fmt     Exit Sub Fmt:     cadena = Format(s, "!" &amp; String(18, "@"))     Debug.Print cadena &amp; " : " &amp; _ &amp; Format(valor, mask), "@@@@@@@@@@@@")     Return End Sub </pre>	 <pre> Inmediato Valor (0)      : -15.000,00 Valor (1)      :  6.500,00 Valor (2)      :  8.000,00 Valor (3)      :  9.500,00 Valor actual neto :  4.234,68 </pre>

### 1.6.8 Pmt

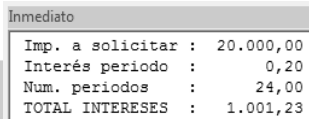
Calcula el pago de una anualidad en función de unos pagos periódicos y un tipo de interés constantes.

Sintaxis:

```
Pmt(intPer, numPer, valorActual[, valorFuturo[, cuandoVencenPagos]])
```

Argumento	Comentario
<b>intPer</b>	Interés por período. Por ejemplo un 20% anual correspondería a $0,2/12 = 0,016$ .
<b>numPer</b>	Número de periodos de pago.
<b>valorActual</b>	Saldo actual o importe inicial.
<b>valorFuturo</b>	Valor futuro que se desea tener una vez realizado el pago final.
<b>cuandoVencenPagos</b>	Flag que indica si los pagos vencen al final del período (0) o al inicio (1).

Ejemplo:

Código	Resultado
<pre>Sub muestra_Pmt()     Dim valPago, intPer, numPer As Double     Dim valorActual, valorFuturo,     cuandoVencenPagos     Dim valor As Double     Dim mask, s, cadena As String      mask = "#,##0.00"      intPer = 0.2     numPer = 24     valorActual = 20000     valorFuturo = 0     cuandoVencenPagos = 1     valPago = Pmt(intPer / 12, numPer, -     valorActual, _</pre>	 <pre>Immediate Imp. a solicitar : 20.000,00 Interés periodo : 0,20 Num. periodos : 24,00 TOTAL INTERESES : 1.001,23</pre>

<pre>         valorFuturo, cuandoVencenPagos)         s = "Imp. a solicitar": valor = valorActual: GoSub Fmt         s = "Interés periodo": valor = intPer: GoSub Fmt         s = "Num. periodos": valor = numPer: GoSub Fmt         s = "TOTAL INTERESES ": valor = valPago: GoSub Fmt          Exit Sub Fmt:         cadena = Format(s, "!@@@@@@@@@@@@@@@@")         Debug.Print cadena &amp; " : " &amp;         &amp; Format(Format(valor, mask), "@@@@@@@@@@@@")         Return End Sub </pre>	
---	--

### 1.6.9 PPmt

Calcula el pago nominal correspondiente a un periodo determinado de una anualidad en función de unos pagos periódicos y un tipo de interés constantes.

Sintaxis:

```
PPmt(intPer, perPago, numPer, valorActual[, valorFuturo[,
cuandoVencenPagos]])
```

Argumento	Comentario
<b>intPer</b>	Interés por período. Por ejemplo un 20% anual correspondería a $0,2/12 = 0,016$ .
<b>perPago</b>	Período de pago comprendido entre 1 y numPer.
<b>numPer</b>	Número de periodos de pago.
<b>valorActual</b>	Saldo actual o importe inicial.
<b>valorFuturo</b>	Valor futuro que se desea tener una vez realizado el pago final.
<b>cuandoVencenPagos</b>	Flag que indica si los pagos vencen al final del período (0) o al inicio (1).



Ejemplo:

Código	Resultado																																							
<pre>Sub muestra_PPmt()     Dim valInt, intPer, perPago, numPer As Double     Dim valorActual, valorFuturo, cuandoVencenPagos     Dim totalInt, valor As Double     Dim mask, s, cadena As String      mask = "#,##0.00"      intPer = 0.2     numPer = 12     valorActual = 20000     valorFuturo = 0     cuandoVencenPagos = 1      s = "Imp. a solicitar": valor = valorActual: GoSub Fmt     s = "Interés periodo": valor = intPer: GoSub Fmt     s = "Num. periodos": valor = numPer: GoSub Fmt     For perPago = 1 To numPer         valInt = PPmt(intPer / 12, perPago, numPer, -valorActual, valorFuturo, cuandoVencenPagos)          s = "Pago (" &amp; perPago &amp; ")": valor = valInt: GoSub Fmt     Next perPago      Exit Sub Fmt:     cadena = Format(s, "!@@@@@@@@@@@@@@@@")     Debug.Print cadena &amp; " : " &amp; Format(Format(valor, mask), "@@@@@@@@@@@@@@")     Return End Sub</pre>	<div>Immediato</div> <table><tr><td>Num. periodos</td><td>:</td><td>12,00</td></tr><tr><td>Pago (1)</td><td>:</td><td>1.822,32</td></tr><tr><td>Pago (2)</td><td>:</td><td>1.519,36</td></tr><tr><td>Pago (3)</td><td>:</td><td>1.544,68</td></tr><tr><td>Pago (4)</td><td>:</td><td>1.570,42</td></tr><tr><td>Pago (5)</td><td>:</td><td>1.596,60</td></tr><tr><td>Pago (6)</td><td>:</td><td>1.623,21</td></tr><tr><td>Pago (7)</td><td>:</td><td>1.650,26</td></tr><tr><td>Pago (8)</td><td>:</td><td>1.677,77</td></tr><tr><td>Pago (9)</td><td>:</td><td>1.705,73</td></tr><tr><td>Pago (10)</td><td>:</td><td>1.734,16</td></tr><tr><td>Pago (11)</td><td>:</td><td>1.763,06</td></tr><tr><td>Pago (12)</td><td>:</td><td>1.792,44</td></tr></table>	Num. periodos	:	12,00	Pago (1)	:	1.822,32	Pago (2)	:	1.519,36	Pago (3)	:	1.544,68	Pago (4)	:	1.570,42	Pago (5)	:	1.596,60	Pago (6)	:	1.623,21	Pago (7)	:	1.650,26	Pago (8)	:	1.677,77	Pago (9)	:	1.705,73	Pago (10)	:	1.734,16	Pago (11)	:	1.763,06	Pago (12)	:	1.792,44
Num. periodos	:	12,00																																						
Pago (1)	:	1.822,32																																						
Pago (2)	:	1.519,36																																						
Pago (3)	:	1.544,68																																						
Pago (4)	:	1.570,42																																						
Pago (5)	:	1.596,60																																						
Pago (6)	:	1.623,21																																						
Pago (7)	:	1.650,26																																						
Pago (8)	:	1.677,77																																						
Pago (9)	:	1.705,73																																						
Pago (10)	:	1.734,16																																						
Pago (11)	:	1.763,06																																						
Pago (12)	:	1.792,44																																						

## 1.6.10 PV

Calcula el valor actual de una anualidad basada en pagos periódicos constantes que se van a realizar en el futuro, con un tipo de interés también constante.

Sintaxis:

PV(intPer, numPer, pagoPer [,valorFuturo[, cuandoVencenPagos]])	
Argumento	Comentario
<b>intPer</b>	Interés por período. Por ejemplo un 20% anual correspondería a $0,2/12 = 0,016$ .
<b>numPer</b>	Número de periodos de pago.
<b>pagoPer</b>	Pago a realizar en cada periodo.
<b>valorFuturo</b>	Valor futuro que se desea tener una vez realizado el pago final.
<b>cuandoVencenPagos</b>	Flag que indica si los pagos vencen al final del período (0) o al inicio (1).

Ejemplo:

Código	Resultado
<pre> Sub muestra_PV()     Dim intPer, numPer, pagoPer As Double     Dim valorActual, valorFuturo, cuandoVencenPagos As Double     Dim valor As Double     Dim mask, s, cadena As String      mask = "#,##0.00"      intPer = 0.2     pagoPer = 12000     numPer = 24     valorFuturo = 200000     cuandoVencenPagos = 1      valorActual = PV(intPer, numPer, - pagoPer, _     valorFuturo, cuandoVencenPagos)      s = "Valor futuro": valor = valorFuturo: GoSub Fmt     s = "Pago por período": valor = pagoPer: GoSub Fmt     s = "Interés periodo": valor = intPer: GoSub Fmt </pre>	<div> <div>Inmediato</div> <div> <div>Valor futuro : 200.000,00</div> <div>Pago por periodo : 12.000,00</div> <div>Interés periodo : 0,20</div> <div>VALOR ACTUAL : 68.578,48</div> </div> </div>

```

s = "VALOR ACTUAL ": valor = valorActual:
GoSub Fmt

Exit Sub

Fmt:
cadena = Format(s, "!@@@@@@@@@@@@@@@@")
Debug.Print cadena & " : "
& Format(Format(valor, mask),
"@@@@@@@@@@@@")
Return
End Sub

```

### 1.6.11 Rate

Calcula el tipo de interés por periodo de una anualidad.

Sintaxis:

```

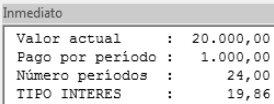
Rate(numPer, pagoPer, valorActual[, valorFuturo[,
cuandoVencenPagos[, factor]])

```

Argumento	Comentario
<b>numPer</b>	Número de periodos de pago.
<b>pagoPer</b>	Pago a realizar en cada periodo.
<b>valorActual</b>	Saldo actual o importe inicial.
<b>valorFuturo</b>	Valor futuro que se desea tener una vez realizado el pago final.
<b>cuandoVencenPagos</b>	Flag que indica si los pagos vencen al final del período (0) o al inicio (1).
<b>factor</b>	Estimación que devuelve Rate. Por defecto es 0.1.

Ra-Ma®

Ejemplo:

Código	Resultado
<pre> Sub muestra_Rate()     Dim intPer, numPer, pagoPer As Double     Dim valorActual, valorFuturo,     cuandoVencenPagos As Double     Dim factor, valor As Double     Dim mask, s, cadena As String      mask = "#,##0.00"      pagoPer = 1000     numPer = 24     valorActual = 20000     valorFuturo = 0     cuandoVencenPagos = 1     factor = 0.1      intPer = (Rate(numPer, -pagoPer,     valorActual, _     valorFuturo, cuandoVencenPagos, factor) *     12) * 100      s = "Valor actual": valor = valorActual: GoSub Fmt     s = "Pago por período": valor = pagoPer: GoSub Fmt     s = "Número períodos": valor = numPer: GoSub Fmt     s = "TIPO INTERES ": valor = intPer: GoSub Fmt      Exit Sub Fmt:     cadena = Format(s, "!@@@@@@@@@@@@@@@@")     Debug.Print cadena &amp; " : " _     &amp; Format(Format(valor, mask), "@@@@@@@@")     Return End Sub </pre>	

### 1.6.12 SLN

Calcula la depreciación lineal de un bien durante un único periodo.

Sintaxis:

SLN(coste, valResidual, duración)
-----------------------------------

Argumento	Comentario
<b>coste</b>	Coste inicial del bien.
<b>valResidual</b>	Valor del bien al final de su vida útil.
<b>duración</b>	Duración de la vida útil del bien.

Ejemplo:

Código	Resultado								
<pre> Sub muestra_SLN()     Dim depr, coste, valResidual, duracion, valor     Dim mask, s, cadena As String      mask = "#,##0.00"      coste = 10000     valResidual = 10     duracion = 5      depr = SLN(coste, valResidual, duracion)      s = "Coste inicial": valor = coste: GoSub Fmt     s = "Valor residual": valor = valResidual: GoSub Fmt     s = "Duración": valor = duracion: GoSub Fmt     s = "DEPRECIACION ": valor = depr: GoSub Fmt      Exit Sub Fmt:     cadena = Format(s, "!@@@@@@@@@@@@")     Debug.Print cadena &amp; " : "     &amp; Format(valor, mask), "@@@@@@@@")     Return End Sub </pre>	<div> <div>Inmediato</div> <table> <tr> <td>Coste inicial</td><td>: 10.000,00</td></tr> <tr> <td>Valor residual</td><td>: 10,00</td></tr> <tr> <td>Duración</td><td>: 5,00</td></tr> <tr> <td>DEPRECIACION</td><td>: 1.998,00</td></tr> </table> </div>	Coste inicial	: 10.000,00	Valor residual	: 10,00	Duración	: 5,00	DEPRECIACION	: 1.998,00
Coste inicial	: 10.000,00								
Valor residual	: 10,00								
Duración	: 5,00								
DEPRECIACION	: 1.998,00								

### 1.6.13 SYD

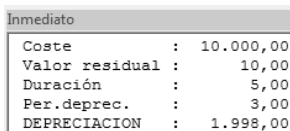
Calcula la depreciación de un bien en dígitos de suma de años durante un periodo especificado.

Sintaxis:

```
SYD(coste, valResidual, duración, perCalDeprec))
```

Argumento	Comentario
<b>coste</b>	Coste inicial del bien.
<b>valResidual</b>	Valor del bien al final de su vida útil.
<b>duración</b>	Duración de la vida útil del bien.
<b>perCalDeprec</b>	Periodo a considerar para la depreciación del bien.

Ejemplo:

Código	Resultado
<pre>Sub muestra_SYD()     Dim depr, coste, valResidual, duracion,     perCalDeprec, valor     Dim mask, s, cadena As String      mask = "#,##0.00"      coste = 10000     valResidual = 10     duracion = 5     perCalDeprec = 3      depr = SYD(coste, valResidual, duracion,     perCalDeprec)      s = "Coste": valor = coste: GoSub Fmt     s = "Valor residual": valor = valResidual: GoSub Fmt     s = "Duración": valor = duracion: GoSub Fmt     s = "Per.deprec. ": valor = perCalDeprec: GoSub Fmt</pre>	 <pre>Inmediato Coste      : 10.000,00 Valor residual : 10,00 Duración   : 5,00 Per.deprec. : 3,00 DEPRECIACION : 1.998,00</pre>

<pre>s = "DEPRECIACION ": valor = depr: GoSub Fmt Exit Sub Fmt: cadena = Format(s, "!@@@@@@@@@@@@@@") Debug.Print cadena &amp; " : " _ &amp; Format(Format(valor, mask), "@@@@@@@@@@@@") Return End Sub</pre>	
---	--

## 1.7 FUNCIONES DE COMPROBACIÓN

Función	Comentario
<b>IsArray(Expresión).</b>	Comprueba si expresión (una variable) es un array o no.
<b>IsDate(Expresión).</b>	Comprueba si expresión tiene un valor que se puede interpretar como tipo fecha.
<b>IsEmpty(Expresión).</b>	Comprueba que expresión tenga algún valor, que se haya inicializado.
<b>IsError(Expresión)</b>	Comprueba si expresión devuelve algún valor de error.
<b>IsNull(Expresión).</b>	Comprueba si expresión contiene un valor nulo debido a datos no válidos.
<b>IsNumeric(Expresión)</b>	Comprueba si expresión tiene un valor que se puede interpretar como numérico.
<b>IsObject(Expresión)</b>	Comprueba si expresión (una variable) representa una variable tipo objeto.
<b>Nothing.</b>	<p>No es propiamente una función, sirve para comprobar si una variable objeto está asociada a un objeto antes de hacer cualquier operación con ella. Recuerde que para trabajar con</p> <p>Una variable objeto antes debe asignarse a uno (mediante la instrucción Set), en caso contrario se producirá un error en el programa cuando utilice el objeto y se detendrá su ejecución.</p>

### 1.7.1 IsArray

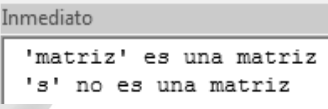
Comprueba si expresión (o variable) es un array o no.

Sintaxis:

```
IsArray(expresión)
```

Argumento	Comentario
<b>expresión</b>	Expresión a evaluar.

Ejemplo:

Código	Resultado
<pre>Sub muestra_IsArray()     Dim matriz(3) As String     Dim s As String     If IsArray(matriz) Then         Debug.Print "'matriz' es una matriz"     Else         Debug.Print "'matriz' no es una matriz"     End If     If IsArray(s) Then         Debug.Print "'s' es una matriz"     Else         Debug.Print "'s' no es una matriz"     End If End Sub</pre>	

Ra-Ma®



## 1.7.2 IsDate

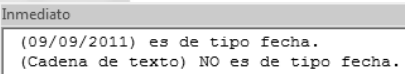
Comprueba si expresión tiene un valor que se puede interpretar como tipo fecha.

Sintaxis:

```
IsDate(expresión)
```

Argumento	Comentario
<b>expresión</b>	Expresión a evaluar.

Ejemplo:

Código	Resultado
<pre>Sub muestra_IsDate()     Dim s As String     s = "09/09/2011"     If IsDate(s) Then         Debug.Print "(" &amp; s &amp; ") es de tipo fecha."     Else         Debug.Print "(" &amp; s &amp; ") NO es de tipo fecha."     End If     s = "Cadena de texto"     If IsDate(s) Then         Debug.Print "(" &amp; s &amp; ") es de tipo fecha."     Else         Debug.Print "(" &amp; s &amp; ") NO es de tipo fecha."     End If End Sub</pre>	 <pre>Inmediato (09/09/2011) es de tipo fecha. (Cadena de texto) NO es de tipo fecha.</pre>

Ra-Ma®

### 1.7.3 IsEmpty

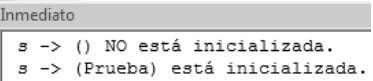
Comprueba que expresión tenga algún valor y por tanto, que se haya inicializado.

Sintaxis:

IsEmpty (expresión)
---------------------

Argumento	Comentario
<b>expresión</b>	Expresión a evaluar.

Ejemplo:

Código	Resultado
<pre>Sub muestra_IsEmpty()     Dim s     If IsEmpty(s) Then         Debug.Print "s -&gt; (" &amp; s &amp; ") NO está inicializada."     Else         Debug.Print "s -&gt; (" &amp; s &amp; ") está inicializada."     End If     s = "Prueba"     If IsEmpty(s) Then         Debug.Print "s -&gt; (" &amp; s &amp; ") NO está inicializada."     Else         Debug.Print "s -&gt; (" &amp; s &amp; ") está inicializada."     End If End Sub</pre>	

## 1.7.4 IsError

Comprueba si expresión devuelve algún valor de error especificado por el usuario.

Sintaxis:

IsError(expresión)

Argumento	Comentario
<b>expresión</b>	Expresión a evaluar.

Ejemplo:

Código	Resultado
<pre> Sub muestra_IsError()     Dim s As Variant     s = CVErr(123)     If IsError(s) Then         Debug.Print "s -&gt; (" &amp; CStr(s) &amp; ") es un error."     Else         Debug.Print "s -&gt; (" &amp; s &amp; ") NO es un error."     End If     s = "a"     If IsError(s) Then         Debug.Print "s -&gt; (" &amp; CStr(s) &amp; ") es un error."     Else         Debug.Print "s -&gt; (" &amp; s &amp; ") NO es un error."     End If End Sub </pre>	<p>Immediate</p> <pre> s -&gt; (Error 123) es un error. s -&gt; (a) NO es un error. </pre>

Ra-Ma®

## 1.7.5 IsNull

Comprueba si expresión contiene un valor nulo debido a datos no válidos.

Sintaxis:

IsNull(expresión)
-------------------

Argumento	Comentario
<b>expresión</b>	Expresión a evaluar.

Ejemplo:

Código	Resultado
<pre>Sub muestra_IsNull()     Dim s As Variant     s = Null     If IsNull(s) Then         Debug.Print "s -&gt; (" &amp; s &amp; ") es Null."     Else         Debug.Print "s -&gt; (" &amp; s &amp; ") NO es Null."     End If     s = "Prueba"     If IsNull(s) Then         Debug.Print "s -&gt; (" &amp; s &amp; ") es Null."     Else         Debug.Print "s -&gt; (" &amp; s &amp; ") NO es Null."     End If End Sub</pre>	<div>Inmediato</div> <pre>s -&gt; () es Null. s -&gt; (Prueba) NO es Null.</pre>

Ra-Ma®

## 1.7.6 IsNumeric

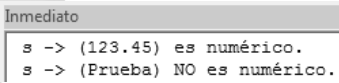
Comprueba si expresión tiene un valor que se puede interpretar como numérico.

Sintaxis:

IsNumeric (expresión)
-----------------------

Argumento	Comentario
<b>expresión</b>	Expresión a evaluar.

Ejemplo:

Código	Resultado
<pre> Sub muestra_IsNumeric()     Dim s As Variant     s = "123.45"     If IsNumeric(s) Then         Debug.Print "s -&gt; (" &amp; s &amp; ") es numérico."     Else         Debug.Print "s -&gt; (" &amp; s &amp; ") NO es numérico."     End If     s = "Prueba"     If IsNumeric(s) Then         Debug.Print "s -&gt; (" &amp; s &amp; ") es numérico."     Else         Debug.Print "s -&gt; (" &amp; s &amp; ") NO es numérico."     End If End Sub </pre>	

Ra-Ma®

### 1.7.7 IsObject

Comprueba si expresión (una variable) representa una variable tipo objeto.

Sintaxis:

IsObject (expresión)
----------------------

Argumento	Comentario
<b>expresión</b>	Expresión a evaluar.

Ejemplo:

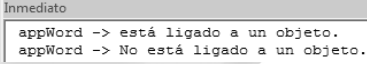
Código	Resultado
<pre>Sub muestra_IsObject()   Dim s As String   Dim o As Object   If IsObject(o) Then     Debug.Print "o -&gt; es un objeto."   Else     Debug.Print "o -&gt; NO es un objeto."   End If   s = "Prueba"   If IsObject(s) Then     Debug.Print "s -&gt; es un objeto."   Else     Debug.Print "s -&gt; NO es un objeto."   End If End Sub</pre>	<div>Inmediato</div> <div>o -&gt; es un objeto. s -&gt; NO es un objeto.</div>

Ra-Ma®

### 1.7.8 Nothing

Se trata de una palabra clave que sirve para una variable de tipo objeto de un objeto real. De la misma manera que se puede desligar dicha variable utilizando **Set** también es posible comprobar si la variable está ligada a un objeto o no (porque contiene **Nothing**).

Ejemplo:

Código	Resultado
<pre> Sub muestra_Nothing()     Dim appWord As Word.Application     Set appWord = New Word.Application      If appWord Is Nothing Then         Debug.Print "appWord -&gt; No está ligado a un objeto."     Else         Debug.Print "appWord -&gt; está ligado a un objeto."     End If      Set appWord = Nothing      If appWord Is Nothing Then         Debug.Print "appWord -&gt; No está ligado a un objeto."     Else         Debug.Print "appWord -&gt; está ligado a un objeto."     End If End Sub </pre>	 <pre> Immediate appWord -&gt; está ligado a un objeto. appWord -&gt; No está ligado a un objeto. </pre>

## 1.8 FUNCIONES DE CONVERSIÓN DE TIPO

Función	Comentario
Cbool	Convierte una expresión a tipo Boolean.
Cbyte	Convierte una expresión a tipo Byte.
Ccur	Convierte una expresión a tipo Currency.
Cdate	Convierte una expresión a tipo Date.
CDbl	Convierte una expresión a tipo Double.
Cdec	Convierte una expresión a tipo Decimal.
CInt	Convierte una expresión a tipo Integer.
CLng	Convierte una expresión a tipo Long.
CLngLng	Convierte una expresión a tipo LongLong.
CSng	Convierte una expresión a tipo Single.
CStr	Convierte una expresión a tipo String.
CVar	Convierte una expresión a tipo Variant.
CVDate	Convierte una expresión a tipo Variant de tipo Date. Se utiliza por compatibilidad con versiones anteriores.
CVErr	Permite crear errores definidos por el usuario.
Fix	Devuelve la parte entera de un número despreciando los decimales.
Hex	Devuelve el valor hexadecimal de un número.
Int	Devuelve la parte entera de un número. En los números negativos, se devuelve el primer negativo que es menor o igual que la expresión.
Oct	Devuelve el valor octal de un número.
Str	Convierte un número en un String.
Val	Convierte un String en número.



### 1.8.1 Cbool

Convierte una expresión a tipo Boolean. Si la expresión da un resultado diferente de 0, se evalúa como True. En caso contrario, se evalúa como False.

Sintaxis:

Cbool (expresión)

Argumento	Comentario
<b>expresión</b>	Expresión a convertir.

Ejemplo:

Código	Resultado
<pre>Sub muestra_Cbool()     Dim i, j As Integer     i = 0     Debug.Print "CBool(" &amp; i &amp; ") (" &amp; CBool(i) &amp; ")"     j = 1     Debug.Print "CBool(" &amp; j &amp; ") (" &amp; CBool(j) &amp; ")"      Debug.Print "CBool(" &amp; i &amp; " = " &amp; j &amp; _     ") (" &amp; CBool(i = j) &amp; ")" End Sub</pre>	<div>Inmediato</div> <pre>CBool(0) (Falso) CBool(1) (Verdadero) CBool(0 = 1) (Falso)</pre>

Ra-Ma®

## 1.8.2 Cbyte

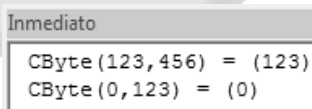
Convierte una expresión a tipo Byte.

Sintaxis:

```
Cbyte(expresión)
```

Argumento	Comentario
<b>expresión</b>	Expresión a convertir.

Ejemplo:

Código	Resultado
<pre>Sub muestra_Cbyte()   Dim i   i = 123.456   Debug.Print "CByte(" &amp; i &amp; ") = (" &amp; CByte(i) &amp; ")"   i = 0.123   Debug.Print "CByte(" &amp; i &amp; ") = (" &amp; CByte(i) &amp; ")" End Sub</pre>	

## 1.8.3 Ccur

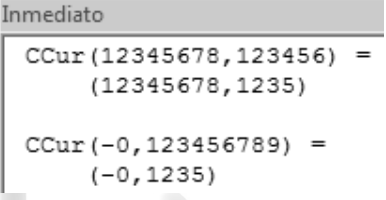
Convierte una expresión a tipo Currency.

Sintaxis:

```
Ccur(expresión)
```

Argumento	Comentario
<b>expresión</b>	Expresión a convertir.

Ejemplo:

Código	Resultado
<pre>Sub muestra_Ccur()   Dim i   i = 12345678.123456   Debug.Print "CCur(" &amp; i &amp; ") = "   Debug.Print "      (" &amp; CCur(i) &amp; ") "   Debug.Print   i = -0.123456789   Debug.Print "CCur(" &amp; i &amp; ") = "   Debug.Print "      (" &amp; CCur(i) &amp; ") " End Sub</pre>	

## 1.8.4 Cdate

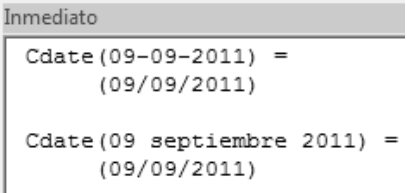
Convierte una expresión a tipo Date.

Sintaxis:

Cdate( <u>expresión</u> )
---------------------------

Argumento	Comentario
<b>expresión</b>	Expresión a convertir.

Ejemplo:

Código	Resultado
<pre>Sub muestra_Cdate()   Dim i As String   i = "09-09-2011"   Debug.Print "Cdate(" &amp; i &amp; ") = "   Debug.Print "      (" &amp; CDate(i) &amp; " &amp;   ") "   i = "09 septiembre 2011"   Debug.Print   Debug.Print "Cdate(" &amp; i &amp; ") = "   Debug.Print "      (" &amp; CDate(i) &amp; " &amp;   ") " End Sub</pre>	

### 1.8.5 CDb1

Convierte una expresión a tipo Double.

Sintaxis:

CDbl (expresión)

Argumento	Comentario
expresión	Expresión a convertir.

Ejemplo:

Código	Resultado
<pre>Sub muestra_CDb1()   Dim i As String   i = "0000123,123456"   Debug.Print "CDbl(" &amp; i &amp; ") = "   Debug.Print "      (" &amp; CDb1(i) &amp; ") "   i = "-0,56789e3"   Debug.Print   Debug.Print "CDbl(" &amp; i &amp; ") = "   Debug.Print "      (" &amp; CDb1(i) &amp; ") " End Sub</pre>	<div>Inmediato</div> <div>CDbl (0000123,123456) =                           (123,123456)</div> <div>CDbl (-0,56789e3) =                       (-567,89)</div>

### 1.8.6 Cdec

Convierte una expresión a tipo Decimal.

Sintaxis:

CDec (expresión)

Argumento	Comentario
expresión	Expresión a convertir.

Ejemplo:

Código	Resultado
<pre>Sub muestra_CDec()   Dim i As String   i = "0000123,123456"   Debug.Print "CDec(" &amp; i &amp; ") = "   Debug.Print "      (" &amp; CDec(i) &amp; ") "   i = "-0,56789e3"   Debug.Print   Debug.Print "CDec(" &amp; i &amp; ") = "   Debug.Print "      (" &amp; CDec(i) &amp; ") " End Sub</pre>	<p>Inmediato</p> <pre>CDec(0000123,123456) =       (123,123456)  CDec(-0,56789e3) =       (-567,89)</pre>

### 1.8.7 CInt

Convierte una expresión a tipo Integer.

Sintaxis:

CInt(expresión)
-----------------

Argumento	Comentario
<b>expresión</b>	Expresión a convertir.

Ejemplo:

Código	Resultado
<pre>Sub muestra_CInt()   Dim i As String   i = "0000123,123456"   Debug.Print "CInt(" &amp; i &amp; ") = "   Debug.Print "      (" &amp; CInt(i) &amp; ") "   i = "-0,56789e3"   Debug.Print   Debug.Print "CInt(" &amp; i &amp; ") = "   Debug.Print "      (" &amp; CInt(i) &amp; ") " End Sub</pre>	<p>Inmediato</p> <pre>CInt(0000123,123456) =       (123)  CInt(-0,56789e3) =       (-568)</pre>

### 1.8.8 CLng

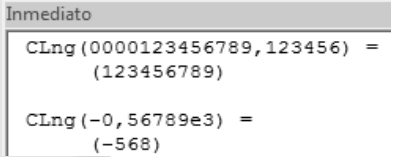
Convierte una expresión a tipo Long.

Sintaxis:

CLng (expresión)
------------------

Argumento	Comentario
<b>expresión</b>	Expresión a convertir.

Ejemplo:

Código	Resultado
<pre>Sub muestra_CLng()     Dim i As String     i = "0000123456789,123456"     Debug.Print "CLng(" &amp; i &amp; ") = "     Debug.Print "      (" &amp; CLng(i) &amp; " )"     i = "-0,56789e3"     Debug.Print "CLng(" &amp; i &amp; ") = "     Debug.Print "      (" &amp; CLng(i) &amp; " )" End Sub</pre>	 <pre>Inmediato CLng(0000123456789,123456) =     (123456789)  CLng(-0,56789e3) =     (-568)</pre>

### 1.8.9 CLngLng

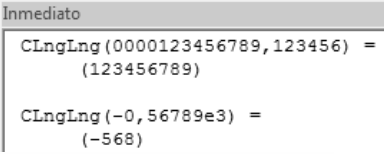
Convierte una expresión a tipo LongLong.

Sintaxis:

CLngLng (expresión)
---------------------

Argumento	Comentario
<b>expresión</b>	Expresión a convertir.

Ejemplo:

Código	Resultado
<pre>Sub muestra_CLngLng()     Dim i As String     i = "0000123456789,123456"     Debug.Print "CLngLng(" &amp; i &amp; ") = "     Debug.Print "      (" &amp; CLngLng(i) &amp; " ")"     i = "-0,56789e3"     Debug.Print     Debug.Print "CLngLng(" &amp; i &amp; ") = "     Debug.Print "      (" &amp; CLngLng(i) &amp; " ")" End Sub</pre>	

### 1.8.10 CSng

Convierte una expresión a tipo Single.

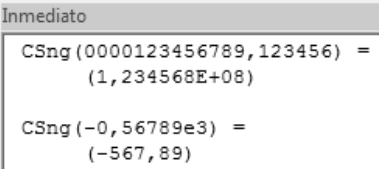
Sintaxis:

CSng(expresión)
-----------------

Argumento	Comentario
<b>expresión</b>	Expresión a convertir.

Ra-Ma®

Ejemplo:

Código	Resultado
<pre>Sub muestra_CSng()     Dim i As String     i = "0000123456789,123456"     Debug.Print "CSng(" &amp; i &amp; ") = "     Debug.Print "      (" &amp; CSng(i) &amp; ")"     i = "-0,56789e3"     Debug.Print     Debug.Print "CSng(" &amp; i &amp; ") = "     Debug.Print "      (" &amp; CSng(i) &amp; ")" End Sub</pre>	 <pre>Inmediato CSng(0000123456789,123456) =     (1,234568E+08)  CSng(-0,56789e3) =     (-567,89)</pre>

## 1.8.11 CStr

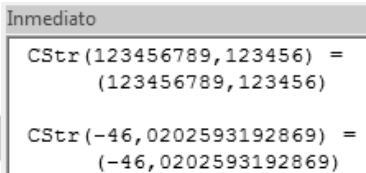
Convierte una expresión a tipo String.

Sintaxis:

CStr(expresión)
-----------------

Argumento	Comentario
<b>expresión</b>	Expresión a convertir.

Ejemplo:

Código	Resultado
<pre>Sub muestra_CStr()     Dim i     i = 123456789.123456     Debug.Print "CStr(" &amp; i &amp; ") = "     Debug.Print "      (" &amp; CStr(i) &amp; ") "     i = -567.89 / 12.34     Debug.Print     Debug.Print "CStr(" &amp; i &amp; ") = "     Debug.Print "      (" &amp; CStr(i) &amp; ") " End Sub</pre>	 <pre>Inmediato CStr(123456789,123456) =     (123456789,123456)  CStr(-46,0202593192869) =     (-46,0202593192869)</pre>



## 1.8.12 CVar

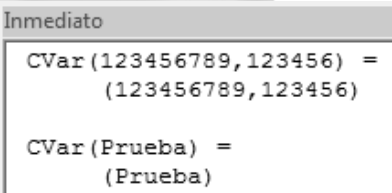
Convierte una expresión a tipo Variant.

Sintaxis:

CVar(expresión)
-----------------

Argumento	Comentario
<b>expresión</b>	Expresión a convertir.

Ejemplo:

Código	Resultado
<pre> Sub muestra_CVar()     Dim i As Double     Dim j As String     i = 123456789.123456     Debug.Print "CVar(" &amp; i &amp; ") = "     Debug.Print "      (" &amp; CVar(i) &amp;     ") "     j = "Prueba"     Debug.Print     Debug.Print "CVar(" &amp; j &amp; ") = "     Debug.Print "      (" &amp; CVar(j) &amp;     ") " End Sub </pre>	

## 1.8.13 CVDate


Convierte una expresión a tipo Variant de tipo Date. Se utiliza por compatibilidad con versiones anteriores.

Sintaxis:

CVDate(expresión)
-------------------

Argumento	Comentario
<b>expresión</b>	Expresión a convertir.

Ejemplo:

Código	Resultado
<pre>Sub muestra_CVDate()     Dim i As String     i = "09-09-2011"     Debug.Print "CVDate(" &amp; i &amp; ") = "     Debug.Print "      (" &amp; CVDate(i) &amp; " ")"     i = "09 septiembre 2011"     Debug.Print     Debug.Print "CVDate(" &amp; i &amp; ") = "     Debug.Print "      (" &amp; CVDate(i) &amp; " ")" End Sub</pre>	

### 1.8.14 CVerErr

Permite crear errores definidos por el usuario.

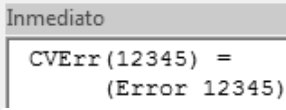
Sintaxis:

CVerErr(expresión)
--------------------

Argumento	Comentario
<b>expresión</b>	Valor que se desea utilizar para identificar el error.

Ra-Ma®

Ejemplo:

Código	Resultado
<pre>Sub muestra_CVErr()   Dim i As Integer   i = 12345   Debug.Print "CVErr(" &amp; i &amp; ") = "   Debug.Print "      (" &amp; CStr(CVErr(i)) &amp; " ")" End Sub</pre>	

### 1.8.15 Fix

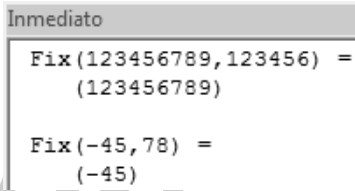
Devuelve la parte entera de un número despreciando los decimales con la particularidad de que en el caso de números negativos, **Fix** devuelve el primer entero negativo mayor o igual al número tratado como argumento de la función. Es decir, **Fix(5,7)=5** y **Fix(-5,7)=-5**.

Sintaxis:

<b>Fix</b> (expresión)
------------------------

Argumento	Comentario
<b>expresión</b>	Expresión a tratar.

Ejemplo:

Código	Resultado
<pre>Sub muestra_Fix()   Dim i As Double   i = 123456789.123456   Debug.Print "Fix(" &amp; i &amp; ") = "   Debug.Print "      (" &amp; Fix(i) &amp; ")"   i = -45.78   Debug.Print "Fix(" &amp; i &amp; ") = "   Debug.Print "      (" &amp; Fix(i) &amp; ")" End Sub</pre>	

## 1.8.16 Hex

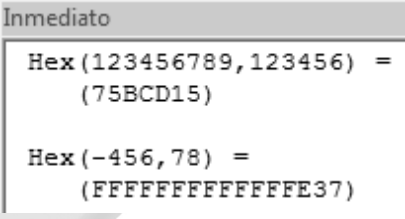
Devuelve el valor hexadecimal de un número.

Sintaxis:

Hex( <i>expresión</i> )
-------------------------

Argumento	Comentario
<b>expresión</b>	Expresión a convertir.

Ejemplo:

Código	Resultado
<pre>Sub muestra_Hex()   Dim i As Double   i = 123456789.123456   Debug.Print "Hex(" &amp; i &amp; ") = "   Debug.Print "      (" &amp; Hex(i) &amp; ") "   i = -456.78   Debug.Print   Debug.Print "Hex(" &amp; i &amp; ") = "   Debug.Print "      (" &amp; Hex(i) &amp; ") " End Sub</pre>	

## 1.8.17 Int

Devuelve la parte entera de un número. En los números negativos, se devuelve el primer negativo que es menor o igual que la expresión. Es decir, **Int(5,7)=5** e **Int(-5,7)=-6**.

Sintaxis:

Int( <i>expresión</i> )
-------------------------

Argumento	Comentario
<b>expresión</b>	Expresión a tratar.

Ejemplo:

Código	Resultado
<pre>Sub muestra_Int()   Dim i As Double   i = 123456789.123456   Debug.Print "Int(" &amp; i &amp; ") = "   Debug.Print "      (" &amp; Int(i) &amp; ")"   i = -45.78   Debug.Print   Debug.Print "Int(" &amp; i &amp; ") = "   Debug.Print "      (" &amp; Int(i) &amp; ")" End Sub</pre>	<div>Inmediato</div> <pre>Int (123456789,123456) =     (123456789)  Int (-45,78) =     (-46)</pre>

### 1.8.18 Oct

Devuelve el valor octal de un número.

Sintaxis:

Oct (expresión)
-----------------

Argumento	Comentario
<b>expresión</b>	Expresión a convertir.

Ra-Ma®

Ejemplo:

Código	Resultado
<pre>Sub muestra_Oct()   Dim i As Integer   i = 1234   Debug.Print "Oct(" &amp; i &amp; ") = "   Debug.Print "      (" &amp; Oct(i) &amp; ") "   i = 8   Debug.Print   Debug.Print "Oct(" &amp; i &amp; ") = "   Debug.Print "      (" &amp; Oct(i) &amp; ") " End Sub</pre>	<div>Inmediato</div> <div>Oct(1234) =           (2322)</div> <div>Oct(8) =           (10)</div>

1.8.19 Str

Convierte un número en un String.

Sintaxis:

Str(expresión)
----------------

Argumento	Comentario
expresión	Expresión a convertir.

Ejemplo:

Código	Resultado
<pre>Sub muestra_Str()   Dim i   i = 123456789.123456   Debug.Print "Str(" &amp; i &amp; ") = "   Debug.Print "      (" &amp; Str(i) &amp; ") "   i = -567.89 / 12.34   Debug.Print   Debug.Print "Str(" &amp; i &amp; ") = "   Debug.Print "      (" &amp; Str(i) &amp; ") " End Sub</pre>	<div>Inmediato</div> <div>Str(123456789,123456) =           ( 123456789.123456)</div> <div>Str(-46,0202593192869) =           (-46.0202593192869)</div>

## 1.8.20 Val

Convierte un String en número teniendo en cuenta que va considerando los números que se encuentra en la expresión de izquierda a derecha hasta que se encuentra un carácter que no es numérico. Los únicos caracteres que elimina y por tanto no considera son los espacios en blanco, los tabuladores y los saltos de línea. Igualmente no tiene en cuenta los prefijos &H y &O para los valores expresados en hexadecimal y octal respectivamente.

Sintaxis:

Val (expresión)
-----------------

Argumento	Comentario
<b>expresión</b>	Expresión a convertir.

Ejemplo:

Código	Resultado
<pre> Sub muestra_Val()     Dim i As String     i = "12 3456 789,123456"     Debug.Print "Val(" &amp; i &amp; ") = "     Debug.Print "      (" &amp; Val(i) &amp; ")"     i = "-567.89ABC 12"     Debug.Print     Debug.Print "Val(" &amp; i &amp; ") = "     Debug.Print "      (" &amp; Val(i) &amp; ")" End Sub </pre>	<div>Inmediato</div> <pre> Val(12 3456 789,123456) =     (123456789)  Val(-567.89ABC 12) =     (-567,89) </pre>

Ra-Ma®

## 1.9 OTRAS FUNCIONES

Función	Descripción
AppActivate	Activa una aplicación que se esté ejecutando en el sistema.
Beep	Emite sonido.
CallByName	Permite ejecutar un método en un objeto o bien asigna o recupera una propiedad del mismo.
Choose	Devuelve un valor de una lista a partir de un índice.
CreateObject	Permite crear un objeto ActiveX y devolver una referencia al mismo.
CurDir	Devuelve la ruta actual.
DeleteSetting	Permite borrar una entrada en el registro de Windows para la aplicación indicada.
Dir	Devuelve una cadena con el primer fichero que coincide con un determinado tipo o atributo de archivo.
DoEvents	Cede el control al sistema operativo permitiendo que éste finalice los eventos que tiene en cola.
Environ	Permite recuperar el contenido de una variable de entorno del sistema.
EOF	Indica si se ha llegado al final de un archivo Random o secuencial devolviendo True o False.
Err	Objeto que contiene información sobre un error que se produce en tiempo de ejecución.
GetAllSettings	Devuelve una lista de claves y valores creados previamente con SaveSetting.
GetObject	Devuelve una referencia a un objeto.



GetSetting	Devuelve el valor de una clave previamente almacenada en el registro de Windows.
IMEStatus	Especifica el modo actual del Editor de método de entrada (Input Method Editor o IME) de Microsoft Windows.
Partition	Devuelve cadena que indica cual es el intervalo que contiene un determinado valor.
QBColor	Devuelve el código de color RGB correspondiente al número de color indicado.
RGB	Devuelve un número que representa un valor de color RGB.
SaveSetting	Permite crear una entrada en el registro de Windows para la aplicación VBA en:  HKEY_CURRENT_USER\Software\VB and VBA Program Settings
SendKeys	Permite enviar secuencias de teclas a una determinada aplicación.
Shell	Ejecuta un programa y devuelve su identificador de proceso.
TypeName	Devuelve información sobre una variable.
VarType	Indica el subtipo de una variable.

### 1.9.1 AppActivate

Activa una aplicación que se esté ejecutando en el sistema.

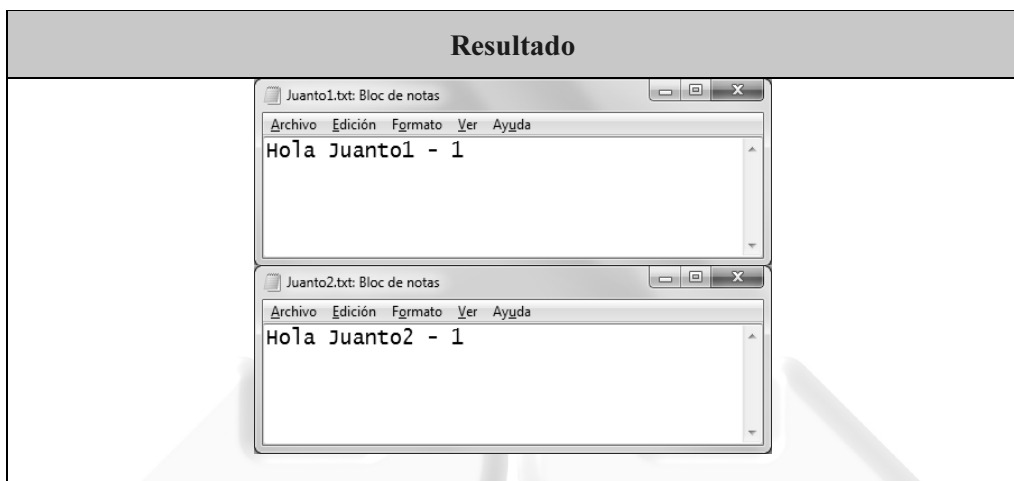
Sintaxis:

AppActivate título [,wait] ó AppActivate(título [,wait])
--

Argumento	Comentario
<b>título</b>	Título que se halla como título de la ventana en la aplicación que se desea activar. Si la aplicación se abre mediante Shell, también es posible utilizar el identificador que devuelve dicha función para referirse a la aplicación a activar.
<b>[,wait]</b>	Opcional. Especifica si la aplicación que hace la llamada tiene el foco antes de activar otra. Por defecto, posee el valor <b>False</b> . Si tiene <b>True</b> , la llamada no se realiza hasta que la aplicación que efectúa la llamada tenga el foco.

Ejemplo:

Código
<pre> Sub muestra_AppActivate()     Dim notepadID1, notepadID2, tiempo As Integer     Dim path, ruta, file1, file2 As String     tiempo = 100     ruta = "C:\Temp"     file1 = "Juanto1.txt"     path = ruta &amp; "\" &amp; file1     If Dir(path) &lt;&gt; "" Then         notepadID1 = Shell("notepad.exe " &amp; path, vbNormalFocus)     Else         MsgBox ("El fichero (" &amp; path &amp; ") no existe")     End If     Sleep tiempo     file2 = "Juanto2.txt"     path = ruta &amp; "\" &amp; file2     If Dir(path) &lt;&gt; "" Then         notepadID2 = Shell("notepad.exe " &amp; path, vbNormalFocus)     Else         MsgBox ("El fichero (" &amp; path &amp; ") no existe")     End If     Sleep tiempo ' Si los ficheros ya están abiertos ...     AppActivate notepadID1     Sleep tiempo     Application.SendKeys "Hola Juanto1 - 1^M"     Sleep tiempo     AppActivate (notepadID2)     Sleep tiempo     Application.SendKeys "Hola Juanto2 - 1^M" End Sub </pre>



### 1.9.2 Beep

Emite sonido.

Sintaxis:

Beep
------

Argumento	Comentario
<b>ninguno</b>	No tiene.

Ejemplo:

Código	Resultado
<pre>Sub muestra_Beep()   Dim i As Integer   For i = 1 To 10     Beep   Next i End Sub</pre>	Emite un sonido.

### 1.9.3 CallByName

Permite ejecutar un método en un objeto o bien asigna o recupera una propiedad del mismo.

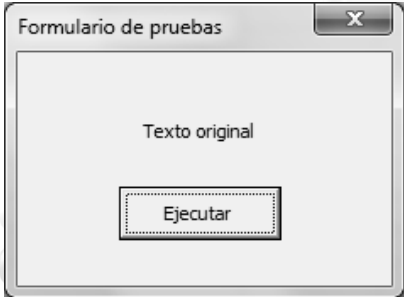
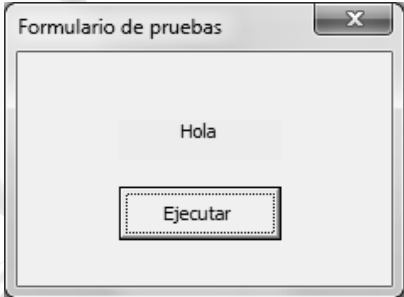
Sintaxis:

```
CallByName(objeto, método, tipo_llamada, [argumentos()])  
ó  
CallByName objeto, método, tipo llamada, [argumentos()]
```

Argumento	Comentario										
<b>objeto</b>	Objeto sobre el que se desea trabajar.										
<b>método</b>	Método a ejecutar o propiedad a consultar o establecer.										
<b>tipo_llamada</b>	Indica el tipo de procedimiento que se llama. Puede ser alguno de los siguientes: <table border="1"> <thead> <tr> <th>Valor</th><th>Comentario</th></tr> </thead> <tbody> <tr> <td><b>VbGet</b></td><td>Indica Property Get.</td></tr> <tr> <td><b>VbLet</b></td><td>Indica Property Let.</td></tr> <tr> <td><b>VbMethod</b></td><td>Indica llamada de un método.</td></tr> <tr> <td><b>VbSet</b></td><td>Indica Property Set.</td></tr> </tbody> </table>	Valor	Comentario	<b>VbGet</b>	Indica Property Get.	<b>VbLet</b>	Indica Property Let.	<b>VbMethod</b>	Indica llamada de un método.	<b>VbSet</b>	Indica Property Set.
Valor	Comentario										
<b>VbGet</b>	Indica Property Get.										
<b>VbLet</b>	Indica Property Let.										
<b>VbMethod</b>	Indica llamada de un método.										
<b>VbSet</b>	Indica Property Set.										
<b>argumentos</b>	Matriz de datos a utilizar en función del tipo de llamada.										

Ejemplo:

En el siguiente ejemplo se dispone de un formulario con una etiqueta y un botón. El ejercicio consiste en que al hacer clic sobre el botón, cambiamos el texto y el color de la etiqueta mediante la función **CallByName**.

Código	
<pre>Private Sub btnEjecutar_Click()     CallByName lbPrueba, "Caption", VbLet, "Hola"     CallByName lbPrueba, "BackColor", VbLet, VBA.ColorConstants.vbYellow End Sub</pre>	
Antes de hacer clic	Después de hacer clic
	

### 1.9.4 Choose

Devuelve un valor de una lista a partir de un índice.

Sintaxis:

```
Choose(índice, valor1[, valor2, ... [, valorN]])
```

Argumento	Comentario
<b>índice</b>	Índice que indica el valor de la lista a devolver. Si el índice es menor que 1 o mayor que el número de valores previstos, la función devolverá Null. Índice se redondea al entero más próximo antes de su utilización.
<b>valor1, valor2, ... valorN</b>	Cada uno de los valores posibles que puede retornar la función según el índice.

Ejemplo:

Código	Resultado
<pre>Sub muestra_Choose()   Dim dia As Integer   Dim nomDia As String   dia = 3    nomDia = Choose(dia, "LUN", "MAR", _     "MIE", "JUE", "VIE", "SAB", "DOM")    Debug.Print "Nombre dia (" &amp; dia &amp; _     ") -&gt; (" &amp; nomDia &amp; ")" End Sub</pre>	<div>Inmediato</div> <div>Nombre dia (3) -&gt; (MIE)</div>

### 1.9.5 CreateObject

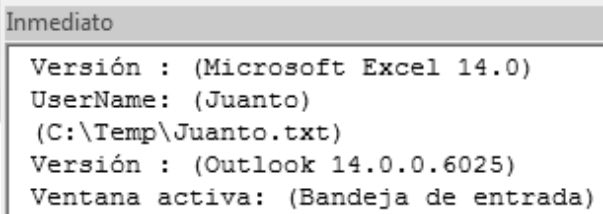
Permite crear un objeto ActiveX y devolver una referencia al mismo.

Sintaxis:

CreateObject(apliClase, [nomServer])
--------------------------------------

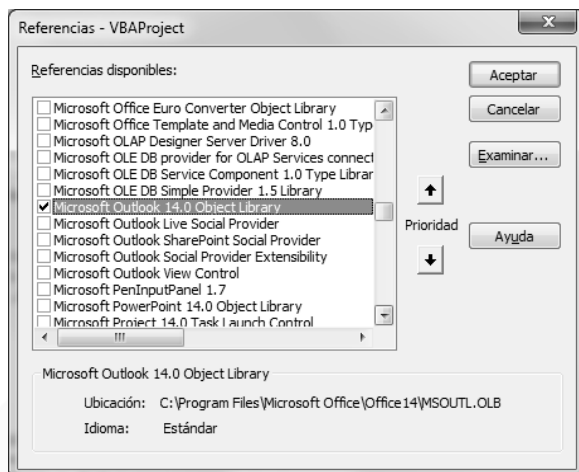
Argumento	Comentario
<b>apliClase</b>	Nombre de la aplicación y clase a crear.
<b>[nomServer]</b>	Nombre del servidor donde se desea crear el objeto. Por defecto, el objeto se crea en el servidor local.

Ejemplo:

Código
<pre>Sub muestra_CreateObject()     Dim file     Dim apli As Object     Set apli = CreateObject("Excel.Application")     Debug.Print "Versión : (" &amp; apli.Value &amp; " " &amp; apli.Version &amp;     ") "     Debug.Print "UserName: (" &amp; apli.UserName &amp; ") "     file = apli.GetOpenFilename     Debug.Print "(" &amp; file &amp; ") "     Set apli = CreateObject("Outlook.Application")     Debug.Print "Versión : (" &amp; apli.Name &amp; " " &amp; apli.Version &amp; ") "     Debug.Print "Ventana activa: (" &amp; apli.ActiveWindow &amp; ") "     Set apli = Nothing End Sub</pre>
Resultado


En este ejemplo, el método **GetOpenFilename** abre un cuadro de diálogo en el que el usuario selecciona el fichero que desea abrir.

El objeto utilizado se ha creado como **Object** y por tanto, el enlace del objeto a la variable se realiza en tiempo de ejecución. También es posible crear el objeto asociando una clase en concreto con lo que, en este caso tenemos la ventaja de poder acceder a sus propiedades en tiempo de diseño. Para poder realizar esta acción, hay que asegurarse de asociar la referencia a la librería que contiene la clase que queremos instanciar. En nuestro ejemplo, asociaríamos las librerías mediante la opción **Herramientas->Referencias...** dando paso al siguiente cuadro de diálogo en el que seleccionaríamos la librería deseada y posteriormente haríamos clic en **Aceptar**:



El ejemplo en este caso, quedaría de la siguiente forma:

### Código

```
Sub muestra_CreateObject()
    Dim apliX As Excel.Application
    Set apliX = CreateObject("Excel.Application")
    Debug.Print "Versión : (" & apliX.Value & " " & apliX.Version &
    ") "
    Debug.Print "UserName: (" & apliX.UserName & ") "
    Set apliX = Nothing

    Dim apliO As Outlook.Application
    Set apliO = CreateObject("Outlook.Application")
    Debug.Print "Versión : (" & apliO.Name & " " & apliO.Version &
    ") "
    Debug.Print "Ventana activa: (" & apliO.ActiveWindow & ") "
    Set apliO = Nothing
End Sub
```

### Resultado

Inmediato

```
Versión : (Microsoft Excel 14.0)
UserName: (Juanto)
Versión : (Outlook 14.0.0.6025)
Ventana activa: (Bandeja de entrada)
```





### 1.9.6 CurDir

Devuelve la ruta actual.

Sintaxis:

CurDir [(unidad)]
-------------------

Argumento	Comentario
<b>unidad</b>	Opcional. Si no se especifica nada, se devuelve la ruta correspondiente a la unidad actual (que normalmente es la correspondiente al disco C:).

Ejemplo:

Código	Resultado
<pre>Sub muestra_CurDir()     Debug.Print "Ruta actual: (" &amp; CurDir &amp; ")"     Debug.Print "Ruta actual: (" &amp; CurDir("D:")     &amp; ")" End Sub</pre>	<div>Inmediato</div> <div>Ruta actual: (C:\Temp)</div> <div>Ruta actual: (D:\)</div>

### 1.9.7 DeleteSetting

Permite borrar una entrada en el registro de Windows para la aplicación indicada (ver función **SaveSetting** comentada en este mismo capítulo más adelante).

Ra-Ma®

Sintaxis:

DeleteSetting aplicación, [sección[, clave]]

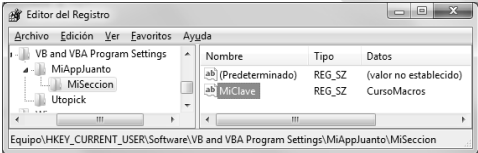
Argumento	Comentario
aplicación	Nombre de la aplicación a tratar.
[sección]	Opcional. Sección dentro de la aplicación. Si no se especifica sección, se elimina la referencia a la aplicación.
[,clave]	Clave a eliminar dentro de la sección. Si no se especifica clave, se elimina la sección entera. Si no se especifica clave se elimina la referencia a la sección.

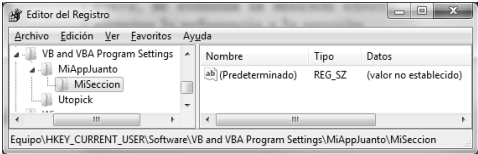
Ejemplo:

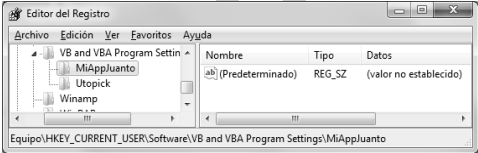
Código

```
Sub muestra_DeleteSetting()  
    ' Elimina clave  
    DeleteSetting "MiAppJuanto", "MiSeccion", "MiClave"  
    ' Elimina sección  
    DeleteSetting "MiAppJuanto", "MiSeccion"  
    ' Elimina aplicación  
    DeleteSetting "MiAppJuanto"  
End Sub
```

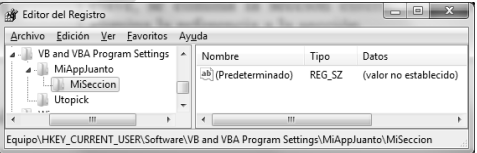
Antes de ejecutar cada eliminación

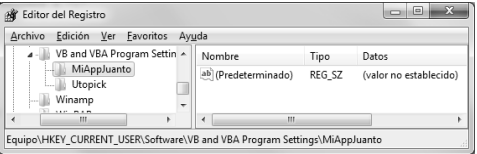


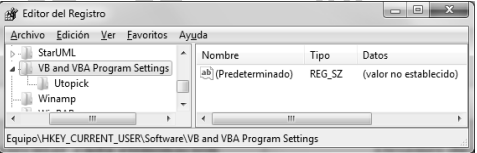




Después de ejecutar







## 1.9.8 Dir

Devuelve una cadena con el primer fichero que coincide con un determinado tipo o atributo de archivo. A partir de la primera devolución, cada vez que se ejecuta Dir sin argumentos se devuelve el siguiente fichero que coincide con la ruta y patrón especificados hasta que devuelve una cadena vacía indicando que ya no hay más archivos por tratar.

Sintaxis:

```
Dir([ruta[, atributos]])
```

Argumento	Comentario																
<b>ruta</b>	<p>Nombre de la ruta o archivo a tratar. Se pueden utilizar los caracteres comodín "*" y "?" para representar varios o un solo carácter respectivamente en el patrón definido como ruta. Así pues, son rutas válidas las siguientes:</p> <pre>Dir("c:\Tmp\*.*.")</pre> <pre>Dir("c:\Tmp\Libro?*.")</pre>																
<b>[,atributos]</b>	<p>Opcional que puede tener alguno de los siguientes valores:</p> <table> <tr> <th>Valores</th><th>Comentario</th></tr> <tr> <td><b>vbNormal</b></td><td>Opción por defecto que hace referencia a archivos sin atributos.</td></tr> <tr> <td><b>vbReadOnly</b></td><td>Archivos sólo de lectura.</td></tr> <tr> <td><b>vbHidden</b></td><td>Archivos ocultos.</td></tr> <tr> <td><b>VbSystem</b></td><td>Archivos del sistema.</td></tr> <tr> <td><b>vbVolume</b></td><td>Indica la etiqueta del volumen.</td></tr> <tr> <td><b>vbDirectory</b></td><td>Directorios o carpetas.</td></tr> <tr> <td><b>vbAlias</b></td><td>Indica que el nombre del archivo es un alias (sólo en Macintosh).</td></tr> </table>	Valores	Comentario	<b>vbNormal</b>	Opción por defecto que hace referencia a archivos sin atributos.	<b>vbReadOnly</b>	Archivos sólo de lectura.	<b>vbHidden</b>	Archivos ocultos.	<b>VbSystem</b>	Archivos del sistema.	<b>vbVolume</b>	Indica la etiqueta del volumen.	<b>vbDirectory</b>	Directorios o carpetas.	<b>vbAlias</b>	Indica que el nombre del archivo es un alias (sólo en Macintosh).
Valores	Comentario																
<b>vbNormal</b>	Opción por defecto que hace referencia a archivos sin atributos.																
<b>vbReadOnly</b>	Archivos sólo de lectura.																
<b>vbHidden</b>	Archivos ocultos.																
<b>VbSystem</b>	Archivos del sistema.																
<b>vbVolume</b>	Indica la etiqueta del volumen.																
<b>vbDirectory</b>	Directorios o carpetas.																
<b>vbAlias</b>	Indica que el nombre del archivo es un alias (sólo en Macintosh).																

Ejemplo:

Código	Resultado
<pre>Sub muestra_Dir()   Dim s As String   s = Dir("c:\Tmp\*.*)"   Do While s &lt;&gt; ""     Debug.Print "(" &amp; s &amp; ")"     s = Dir   Loop End Sub</pre>	<div>Inmediato</div> <pre>(Doc1.docx) (Doc2.docx) (Libro1.xlsx) (Libro2.xlsx)</pre>

## 1.9.9 DoEvents

Cede el control al sistema operativo permitiendo que éste finalice los eventos que tiene en cola. Es útil en bucles de retraso, en llamadas al sistema operativo, etc...

Sintaxis:

DoEvents
----------

Argumento	Comentario
<b>Ninguno</b>	No tiene.

Ejemplo:

Código
<pre>Sub muestra_DoEvents()   Dim i As Integer   For i = 1 To 1000     DoEvents   Next i End Sub</pre>

### 1.9.10 Environ

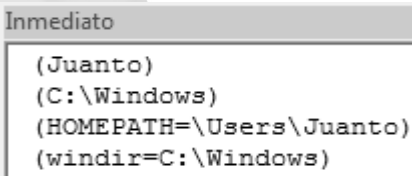
Permite recuperar el contenido de una variable de entorno del sistema.

Sintaxis:

Environ(variable) ó Environ(número)
---

Argumento	Comentario
<b>variable</b>	Variable a tratar.
<b>número</b>	Orden numérico de la variable de entorno en la tabla de cadenas de entorno

Ejemplo:

Código	Resultado
<pre>Sub muestra_Environ()     Dim s As String     s = Environ("username")     Debug.Print "(" &amp; s &amp; ")"     s = Environ("windir")     Debug.Print "(" &amp; s &amp; ")"     s = Environ(11)     Debug.Print "(" &amp; s &amp; ")"     s = Environ(40)     Debug.Print "(" &amp; s &amp; ")" End Sub</pre>	

Ra-Ma®

### 1.9.11 EOF

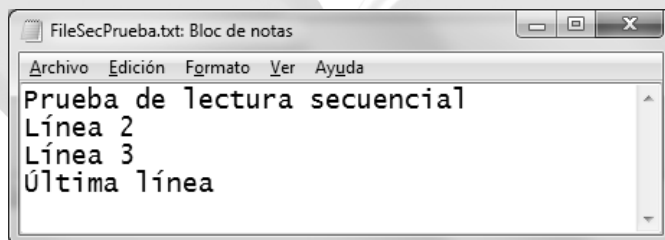
Indica si se ha llegado al final de un archivo Random o secuencial devolviendo True o False.

Sintaxis:

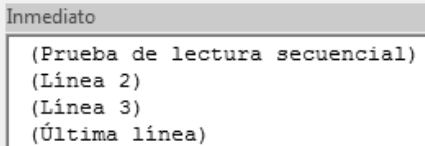
EOF (númeroArchivo)	
Argumento	Comentario
<b>númeroArchivo</b>	Número utilizado en la apertura del fichero.

Ejemplo:

Supongamos que disponemos de un fichero de texto con el siguiente contenido:



Vemos cómo podemos leerlo y detectar el final del mismo:

Código	Resultado
<pre>Sub muestra_EOF()     Dim linea, file As String     file =     "c:\Temp\FileSecPrueba.txt"     Open file For Input As #1     While Not EOF(1)         Line Input #1, linea         Debug.Print "(" &amp; linea &amp;     " "     Wend     Close #1 End Sub</pre>	

### 1.9.12 Err

Se trata de un objeto que contiene información sobre un error que se produce en tiempo de ejecución.

Este objeto posee las siguientes propiedades:

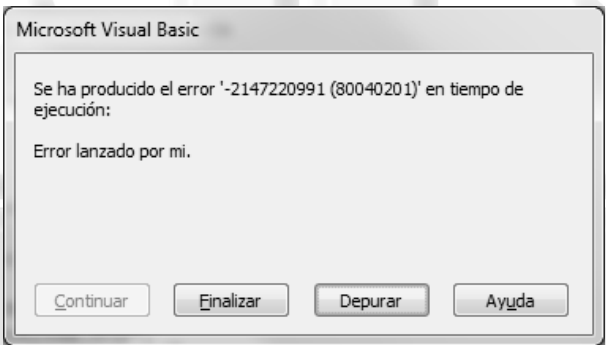
Propiedad	Comentario
<b>Number</b>	Número que indica que error se ha producido.
<b>Source</b>	Objeto o aplicación que ha generado el error.
<b>Description</b>	Devuelve o establece el texto que describe el error.
<b>HelpFile</b>	Ruta en la que se haya el archivo de Ayuda de Microsoft Windows.
<b>HelpContext</b>	Contiene el identificador de contexto de un tema de archivo de Ayuda de Microsoft Windows.
<b>LastDllError</b>	Error de sistema producido como consecuencia de la llamada a una DLL (biblioteca de vínculos dinámicos).

También posee los siguientes métodos:

Método	Comentario												
<b>Clear</b>	Borra las propiedades del objeto Err. Se ejecuta de la siguiente manera:  <code>Err.Clear</code>												
<b>Raise</b>	<p>Método que permite lanzar un error al sistema. Posee la siguiente sintaxis:</p> <p><code>Err.Raise número, origen, descripción, fileAyuda, contextoAyuda</code></p> <table> <tr> <th>Argumento</th><th>Comentario</th></tr> <tr> <td><b>número</b></td><td>Número que identifica el error. Los errores del 0 al 512 están reservados para el sistema. A partir del 513 y, hasta el 65535 (ambos inclusive) se pueden utilizar para definir errores de usuario. Existe una constante de error denominada <b>vbObjectError</b> y es la encargada de contener los errores a utilizar en nuestro módulo (tanto de sistema como los que hayamos definido).</td></tr> <tr> <td><b>origen</b></td><td>Indica la aplicación u objeto que ha generado el error.</td></tr> <tr> <td><b>descripción</b></td><td>Descripción del error. Si no se especifica nada, se atenderá el texto asociado al número de error indicado.</td></tr> <tr> <td><b>fileAyuda</b></td><td>Fichero de ayuda a invocar cuando el usuario pulsa sobre el botón ayuda.</td></tr> <tr> <td><b>contextoAyuda</b></td><td>Identificador del tema dentro de la ayuda invocada al pulsar sobre el botón ayuda.</td></tr> </table>	Argumento	Comentario	<b>número</b>	Número que identifica el error. Los errores del 0 al 512 están reservados para el sistema. A partir del 513 y, hasta el 65535 (ambos inclusive) se pueden utilizar para definir errores de usuario. Existe una constante de error denominada <b>vbObjectError</b> y es la encargada de contener los errores a utilizar en nuestro módulo (tanto de sistema como los que hayamos definido).	<b>origen</b>	Indica la aplicación u objeto que ha generado el error.	<b>descripción</b>	Descripción del error. Si no se especifica nada, se atenderá el texto asociado al número de error indicado.	<b>fileAyuda</b>	Fichero de ayuda a invocar cuando el usuario pulsa sobre el botón ayuda.	<b>contextoAyuda</b>	Identificador del tema dentro de la ayuda invocada al pulsar sobre el botón ayuda.
Argumento	Comentario												
<b>número</b>	Número que identifica el error. Los errores del 0 al 512 están reservados para el sistema. A partir del 513 y, hasta el 65535 (ambos inclusive) se pueden utilizar para definir errores de usuario. Existe una constante de error denominada <b>vbObjectError</b> y es la encargada de contener los errores a utilizar en nuestro módulo (tanto de sistema como los que hayamos definido).												
<b>origen</b>	Indica la aplicación u objeto que ha generado el error.												
<b>descripción</b>	Descripción del error. Si no se especifica nada, se atenderá el texto asociado al número de error indicado.												
<b>fileAyuda</b>	Fichero de ayuda a invocar cuando el usuario pulsa sobre el botón ayuda.												
<b>contextoAyuda</b>	Identificador del tema dentro de la ayuda invocada al pulsar sobre el botón ayuda.												



Ejemplo:

Código
<pre>Sub muestra_Err_Raise()     Err.Raise vbObjectError + 513, "MiMetodo", "Error lanzado por mi." End Sub</pre>
Resultado


### 1.9.13 GetAllSettings

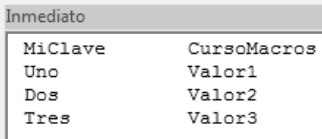
Devuelve una lista de claves y valores creados previamente con **SaveSetting** y almacenados en el registro de **Windows**.

Sintaxis:

GetAllSettings(aplicación, sección)
-------------------------------------

Argumento	Comentario
<b>aplicación</b>	Aplicación a consultar.
<b>sección</b>	Sección de la aplicación a consultar.

Ejemplo:

Código	Resultado
<pre>Sub muestra_GetAllSettings()     Dim v As Variant     Dim i, minimo, maximo As Integer     v = GetAllSettings("MiAppJuanto", "MiSeccion")     minimo = LBound(v, 1)     maximo = UBound(v, 1)     For i = minimo To maximo         Debug.Print v(i, 0), v(i, 1)     Next i End Sub</pre>	 <pre>Immediate MiClave      CursoMacros Uno          Valor1 Dos          Valor2 Tres         Valor3</pre>

### 1.9.14 GetObject

Proporciona una referencia asociada a un objeto de tipo componente Active X.


Sintaxis:

GetObject([ruta] [, clase])
-----------------------------

Argumento	Comentario
<b>ruta</b>	Ruta y nombre del archivo que contiene el objeto.
<b>clase</b>	Opcional. Representa la clase del objeto.

Ra-Ma®

Ejemplo:

Código	Resultado
<pre> Sub muestra_GetObject()     Dim e As Excel.Application     Set e = CreateObject("Excel.Application")     e.Workbooks.Open ("c:\temp\Ventas.xlsm")     e.Visible = True     e.Worksheets(1).Cells(1, 1) = "8000"     e.Worksheets(1).Cells(2, 1) = "1500"     e.Worksheets(1).Cells(3, 1).FormulaLocal = "=SUMA(A1:A2)"     e.Workbooks(1).Save     e.Run "Saluda", "Juanito"     e.Quit     Set e = Nothing End Sub </pre>	

Vemos que el ejemplo, abre el libro Ventas.xlsm, inserta los valores 8000 y 1500 en A1 y A2 respectivamente, los suma en A3 y por último ejecuta una macro llamada Saluda que admite un parámetro.

### 1.9.15 GetSetting

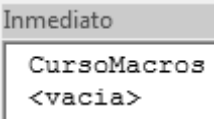
Devuelve el valor de una clave previamente almacenada en el registro de Windows.

Sintaxis:

GetSetting(aplicación, sección, clave[, defecto])
---

Argumento	Comentario
<b>aplicación</b>	Nombre que se desea dar a la aplicación.
<b>sección</b>	Sección dentro de la aplicación.
<b>clave</b>	Clave a crear dentro de la sección.
<b>[, defecto]</b>	Valor a devolver si la clave no tiene valor. En ese caso se devuelve una cadena vacía ("").

Ejemplo:

Código
<pre>Sub muestra_GetSetting()     Debug.Print GetSetting("MiAppJuanto", "MiSeccion", "MiClave")     Debug.Print GetSetting("MiAppJuanto", "MiSeccion", "PruebaClaveVacía", "&lt;vacía&gt;") End Sub</pre>
Resultado


1.9.16 IMEStatus

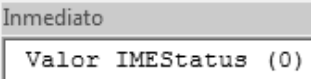
Especifica el modo actual del Editor de método de entrada (Input Method Editor o IME) de Microsoft Windows. Sólo está disponible en versiones del este asiático.

Sintaxis:

IMEStatus
-----------

Argumento	Comentario
ninguno	No tiene.

Ejemplo:

Código	Resultado
<pre>Sub muestra_IMEStatus()     Debug.Print "Valor IMEStatus (" &amp; IMEStatus &amp; ")" End Sub</pre>	

Posibles valores devueltos:

Configuración regional	Valor	Comentario
Japonés	0	Sin control IME (predeterminado).
	1	IME activado.
	2	IME desactivado.
	3	IME deshabilitado.
	4	Modo Hiragana de ancho completo.
	5	Modo Katakana de ancho completo.
	6	Modo Katakana de medio ancho.
	7	Modo alfanumérico de ancho completo.
	8	Modo alfanumérico de medio ancho.
Coreano	0	Sin control IME (predeterminado).
	7	Modo alfanumérico de ancho completo.
	8	Modo alfanumérico de mitad de ancho.
	9	Modo Hangul de ancho completo.
	10	Modo Hangul de mitad de ancho.
Chino	0	Sin control IME (predeterminado).
	1	IME activado.
	2	IME desactivado.

(fuente: Ayuda de Excel)

### 1.9.17 Partition

Devuelve cadena que indica cual es el intervalo que contiene un determinado valor.

Sintaxis:

Partition(número, inicio, fin, intervalo)
---

Argumento	Comentario
<b>número</b>	Número a evaluar.
<b>inicio</b>	Número indicativo del comienzo del intervalo de números. Ha de ser 0 o mayor que 0.
<b>fin</b>	Número que indica el final del intervalo a considerar.
<b>intervalo</b>	División a considerar

Ejemplo:

Código	Resultado												
<pre>Sub muestra_Partition()   Debug.Print Partition(15, 0, 1000, 20)   Debug.Print Partition(37, 0, 1000, 20)   Debug.Print Partition(560, 0, 1000, 20)   Debug.Print Partition(-3, 0, 1000, 20)   Debug.Print Partition(2540, 0, 1000, 20) End Sub</pre>	<table border="1"> <thead> <tr> <th colspan="2">Inmediato</th> </tr> </thead> <tbody> <tr> <td>0:</td> <td>19</td> </tr> <tr> <td>20:</td> <td>39</td> </tr> <tr> <td>560:</td> <td>579</td> </tr> <tr> <td>:</td> <td>-1</td> </tr> <tr> <td>1001:</td> <td></td> </tr> </tbody> </table>	Inmediato		0:	19	20:	39	560:	579	:	-1	1001:	
Inmediato													
0:	19												
20:	39												
560:	579												
:	-1												
1001:													

Ra-Ma®

### 1.9.18 QBColor

Devuelve el código de color RGB correspondiente al número de color indicado.

Sintaxis:

```
QBColor(color)
```

Argumento	Comentario
<b>color</b>	Número de color a representar.

Ejemplo:

Código
<pre>Sub muestra_QBColor()     Dim color As Long     Dim colores     colores = Array("Negro", "Azul", "Verde", _         "Aguamarina", "Rojo", "Fucsia", "Amarillo", _         "Blanco", "Gris", "Azul claro", "Verde claro", _         "Aguamarina claro", "Rojo claro", "Fucsia claro", _         "Amarillo claro", "Blanco brillante")      For color = 0 To 15         Debug.Print "Color (" &amp; Format(color, "@@") &amp; ") (" &amp; _             colores(color) &amp; ") -&gt; (" &amp; QBColor(color) &amp; ")"     Next color End Sub</pre>

Ra-Ma®

Resultado	
<div> <div>Inmediato</div> <div> Color ( 0) (Negro) -&gt; (0)  Color ( 1) (Azul) -&gt; (8388608)  Color ( 2) (Verde) -&gt; (32768)  Color ( 3) (Aguamarina) -&gt; (8421376)  Color ( 4) (Rojo) -&gt; (128)  Color ( 5) (Fucsia) -&gt; (8388736)  Color ( 6) (Amarillo) -&gt; (32896)  Color ( 7) (Blanco) -&gt; (12632256)  Color ( 8) (Gris) -&gt; (8421504)  Color ( 9) (Azul claro) -&gt; (16711680)  Color (10) (Verde claro) -&gt; (65280)  Color (11) (Aguamarina claro) -&gt; (16776960)  Color (12) (Rojo claro) -&gt; (255)  Color (13) (Fucsia claro) -&gt; (16711935)  Color (14) (Amarillo claro) -&gt; (65535)  Color (15) (Blanco brillante) -&gt; (16777215) </div> </div>	

### 1.9.19 RGB

Devuelve un número que representa un valor de color RGB.

Sintaxis:

```
RGB(rojo, verde, azul)
```

Argumento	Comentario
<b>rojo</b>	Componente de rojo (valor entre 0 y 255).
<b>verde</b>	Componente de verde (valor entre 0 y 255).
<b>azul</b>	Componente de azul (valor entre 0 y 255).

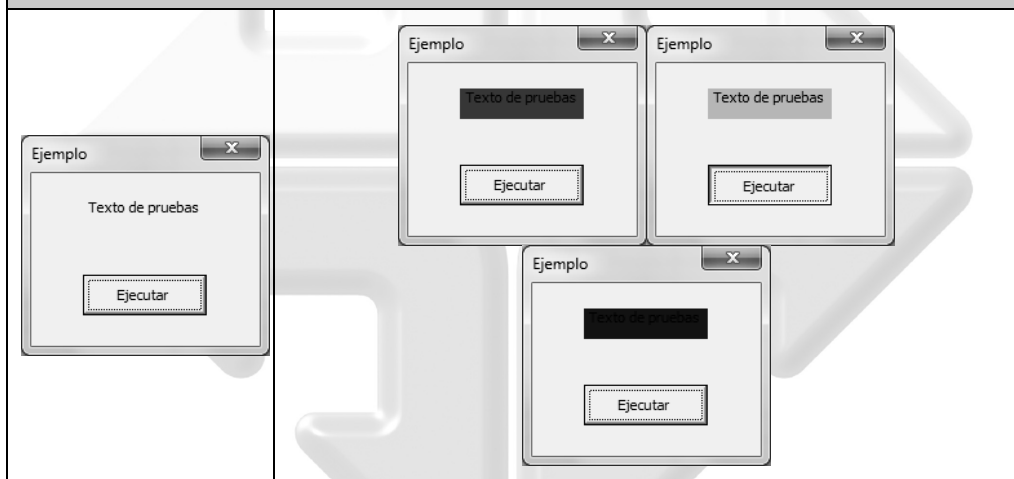
Ejemplo:

En este caso, asociamos el siguiente código al evento clic del botón Ejecutar. El objetivo es que, al hacer clic sobre el botón, la etiqueta cambia de color el fondo 3 veces, esperando 1 segundo, en cada cambio.



**Código**

```
Private Sub btnEjecutar_Click()  
    lbMuestra.BackColor = RGB(255, 0, 0)  
    DoEvents  
    Sleep 1000  
    lbMuestra.BackColor = RGB(0, 255, 0)  
    DoEvents  
    Sleep 1000  
    lbMuestra.BackColor = RGB(0, 0, 255)  
    DoEvents  
End Sub
```

**Resultado**

### 1.9.20 SaveSetting

Permite crear una entrada en el registro de Windows para la aplicación VBA que estamos desarrollando.

Sintaxis:

SaveSetting aplicación, sección, clave, valor
---

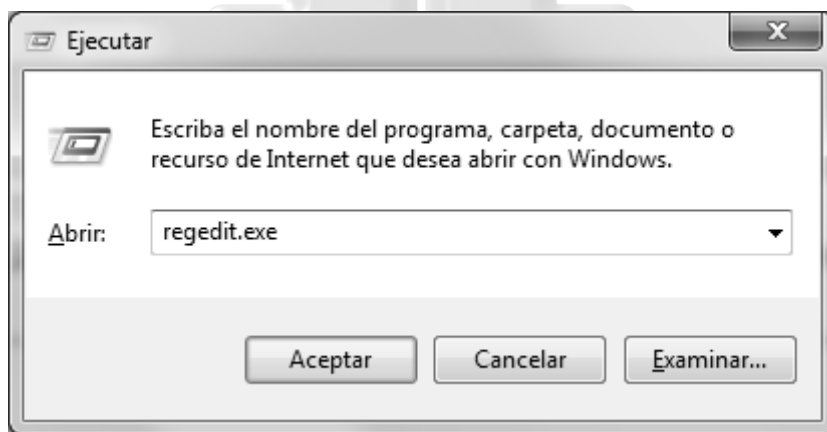
Ra-Ma<sup>®</sup>

Argumento	Comentario
<b>aplicación</b>	Nombre que se desea dar a la aplicación.
<b>sección</b>	Sección dentro de la aplicación.
<b>clave</b>	Clave a crear dentro de la sección.
<b>valor</b>	Valor a asociar a la clave.

Ejemplo:

Código
<pre>Sub muestra_SaveSetting()     SaveSetting "MiAppJuanto", "MiSeccion", "MiClave", "CursoMacros" End Sub</pre>

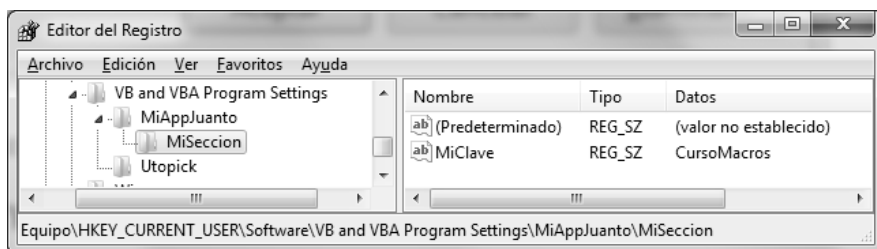
Para comprobar el resultado, ejecutaremos Regedit.exe. Para ello, invocaremos al cuadro de diálogo ejecutar (**Inicio->Ejecutar** o bien tecla **Windows + R**) y tecleamos **regedit.exe** y a continuación **Aceptar**:



Una vez abierto el editor de registro, navegaremos hasta:

**HKEY\_CURRENT\_USER \Software\VB and VBA Program Settings**

En este punto observaremos:



### 1.9.21 SendKeys

Permite enviar secuencias de teclas a una determinada aplicación .

Sintaxis:

```
SendKeys texto[, espera]
```

Argumento	Comentario
<b>texto</b>	Secuencia de teclas o texto a enviar.
<b>[,espera]</b>	Admite los valores False (valor por defecto) o True. Indica si el control se devuelve inmediatamente al procedimiento que envía las teclas (False) o se ha de esperar a que las teclas enviadas sean procesadas (True).

Ejemplo:

El siguiente ejemplo se ejecuta teniendo abiertos 2 ficheros denominados juanto1.txt y juanto2.txt y el objetivo es copiar el contenido de juanto1.txt sobre juanto2.txt (seleccionando todo el texto y copiándolo en el primero y pegándolo en el segundo)

Ra-Ma®

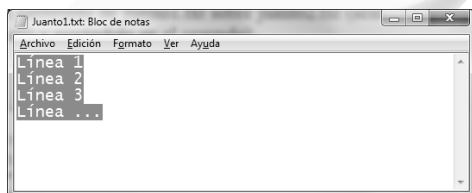
### Código

```
Sub muestra_SendKeys()
    ' Seleccionamos todo el texto y lo copiamos.
    AppActivate ("juanto1.txt: Bloc de notas")
    Application.SendKeys "^e"
    Application.SendKeys "^c"

    ' Seleccionamos todo el texto y pegamos el texto copiado
    anteriormente.
    AppActivate ("juanto2.txt: Bloc de notas")
    Application.SendKeys "^e"
    Application.SendKeys "^v"
End Sub
```

### Resultado

#### Selecciona y copia



#### Pega



## 1.9.22 Shell

Ejecuta un programa y devuelve su identificador de proceso.

Sintaxis:

```
Shell (nombreRuta[, tipoVentana])
```

Argumento	Comentario
<b>nombreRuta</b>	Programa que se desea ejecutar. Si el programa precisa de algún argumento, se puede indicar junto con el mismo.
<b>tipoVentana</b>	Determina el estilo de la ventana en la que se va a ejecutar el programa.

valor	Comentario
<b>vbHide</b>	Se oculta la ventana y se le pasa el foco.
<b>vbNormalFocus</b>	Se pasa el foco a la ventana restaurando su tamaño normal.
<b>vbMinimizedFocus</b>	Se pasa el foco a la ventana pero ésta se muestra como un icono.
<b>vbMaximizedFocus</b>	Se pasa el foco a la ventana y se maximiza.
<b>vbNormalNoFocus</b>	Se restaura la ventana a su tamaño normal pero no se le pasa el foco quedando activa la ventana actual.
<b>vbMinimizedNoFocus</b>	Se minimiza la ventana pero no se le pasa el foco quedando activa la ventana actual.

Ejemplo:

Código	Resultado
<pre>Sub muestra_Shell()   Dim ap As String   ap = Shell("WINWORD.EXE")   Debug.Print ap End Sub</pre>	<p>Abre Microsoft Word</p> <p>y muestra</p> <p>Inmediato</p> <p>5912</p>

### 1.9.23 TypeName

Devuelve información sobre una variable.

Sintaxis:

TypeName(variable)
--------------------

Argumento	Comentario
<b>variable</b>	Variable a analizar.

Ejemplo:

Código	Resultado
<pre>Sub muestra_TypeName()   Dim s As String, i As Integer   s = "Hola"   i = 10   Debug.Print "Tipo de s (" &amp; TypeName(s) &amp;   ") "   Debug.Print "Tipo de i (" &amp; TypeName(i) &amp;   ") " End Sub</pre>	<div>Inmediato</div> <div>Tipo de s (String) Tipo de i (Integer)</div>

Valores que puede devolver la función:

Valor	Resultado
<b>tipo de objeto</b>	Objeto con tipo de objeto tipoobjeto.
<b>Byte</b>	Byte.
<b>Integer</b>	Entero.
<b>Long</b>	Entero largo.
<b>Single</b>	Coma flotante de precisión simple.
<b>Double</b>	Coma flotante de precisión doble.
<b>Currency</b>	Moneda.
<b>Decimal</b>	Decimal.
<b>Date</b>	Fecha.

<b>String</b>	Cadena de texto.
<b>Boolean</b>	Booleano.
<b>Error</b>	Error
<b>Empty</b>	No inicializado.
<b>Null</b>	Datos no válidos.
<b>Object</b>	Objeto.
<b>Nothing</b>	No referencia a objeto.

### 1.9.24 VarType

Indica el subtipo de una variable.

Sintaxis:

<code>VarType(variable)</code>
--------------------------------

Argumento	Comentario
<b>variable</b>	Variable a analizar.

Ejemplo:

Código	Resultado
<pre>Sub muestra_VarType()     Dim s As String, i As Integer     s = "Hola"     i = 10     Debug.Print "Tipo de s (" &amp; VarType(s) &amp; ")"     Debug.Print "Tipo de i (" &amp; VarType(i) &amp; ")" End Sub</pre>	<div>Inmediato</div> <div>Tipo de s (8)</div> <div>Tipo de i (2)</div>

Valores que puede devolver la función:

Constante	Comentario
vbEmpty	Empty
vbNull	Null
vbInteger	Entero
vbLong	Entero largo
vbSingle	Número de precisión simple y de coma flotante
vbDouble	Número de precisión doble y de coma flotante
vbCurrency	Valor tipo moneda
vbDate	Valor tipo fecha
vbString	Cadena
vbObject	Objeto
vbError	Valor de error.
vbBoolean	Booleano.
vbVariant	Variant
vbDataObject	Un objeto de acceso a datos
vbDecimal	Valor decimal
vbByte	Valor de byte
vbLongLong	Entero LongLong (sólo es válido en plataformas de 64 bits).
vbUserDefinedType	Variantes que contienen tipos definidos por el usuario
vbArray	Matriz



## **EL MODELO DE OBJETOS**

---

### **2.1 EL OBJETO APLICACIÓN (APPLICATION)**

#### **2.1.1 Propiedades**

##### **2.1.1.1 ACTIVECELL**

Devuelve un objeto Range que representa la celda activa de la ventana activa o de la que se haya especificado. Se produce un error si no se está visualizando ninguna ventana.

##### **2.1.1.2 ACTIVECHART**

Devuelve un objeto Chart que representa el gráfico seleccionado o activo.

##### **2.1.1.3 ACTIVEPRINTER**

Devuelve o establece el nombre de la impresora que se halla activa.

##### **2.1.1.4 ACTIVEPROTECTEDVIEWWINDOW**

Retorna un objeto ProtectedViewWindow que representa la ventana denominada ‘Vista protegida’ activa. Las ventanas ‘Vista protegida’ se usa para mostrar libros desde ubicaciones potencialmente no seguras (internet, adjuntos en correos, abiertos expresamente como Vista protegida, etc).

### **2.1.1.5 ACTIVESHEET**

Devuelve un objeto que representa la hoja que se halle activa en este momento del libro o de la ventana especificada. Devuelve Nothing si no hay ninguna hoja activa.

### **2.1.1.6 ACTIVEWINDOW**

Devuelve un objeto Window que representa la ventana que se halle activa en este momento.

### **2.1.1.7 ACTIVESTRANS**

Devuelve un objeto Trans que representa el libro de la ventana que se halle activa en este momento.

### **2.1.1.8 ADDINS**

Retorna una colección AddIns con todos los complementos existentes en el cuadro de diálogo Complementos.

### **2.1.1.9 ADDINS2**

Retorna una colección AddIns con todos los complementos disponibles independientemente de si están instalados o no.

### **2.1.1.10 ALERTBEFOREOVERWRITING**

Devuelve un valor que indica si Microsoft Excel muestra un mensaje antes de sobrescribir el contenido de las celdas que no están vacías durante una operación de arrastre de celdas.

### **2.1.1.11 ALTSTARTUPPATH**

Permite definir o consultar el nombre de la carpeta de inicio.

### **2.1.1.12 ALWAYSUSECLEARTYPE**

Permite definir o consultar si se debe usar Clear Type para mostrar las fuentes en el menú, cintas de opciones y cuadros de diálogo.

### **2.1.1.13 ANSWERWIZARD**

Devuelve un objeto de tipo Microsoft.Office.Core.AnswerWizard el cual contiene los archivos usados por el motor de búsqueda de la Ayuda en línea.

### **2.1.1.14 APPLICATION**

Devuelve un objeto Application que representa la aplicación Microsoft Excel. Cuando se utiliza calificador de objeto, devuelve un objeto de tipo Application que representa al creador del objeto especificado.

### **2.1.1.15 ARBITRARYXMLSUPPORTAVAILABLE**

Indica mediante un booleano si las características XML están disponibles.

### **2.1.1.16 ASKTOUPDATELINKS**

Devuelve True si al abrir archivos con vínculos se solicita al usuario que los actualice. Devuelve False si los vínculos se actualizan automáticamente.

### **2.1.1.17 ASSISTANCE**

Devuelve un objeto IAssistance que representa a Microsoft Office Help Viewer.

### **2.1.1.18 ASSISTANT**

Devuelve un objeto Microsoft.Office.Core.Assistant de Microsoft Excel.

### **2.1.1.19 AUTOCORRECT**

Devuelve un objeto AutoCorrect que contiene los atributos de Autocorrección de Microsoft.

Algunos atributos de corrección son: AutoExpandListRange, AutoFillFormulasInLists, CapitalizeNamesOfDays, CorrectCapsLock, etc.

Dicho objeto permite también añadir o eliminar parejas de strings para reemplazar un texto por otro utilizando los métodos AddReplacement y DeleteReplacement respectivamente.

### **2.1.1.20 AUTOFORMATASYOUTYPEREPLACEHYPERLINKS**

Permite indicar si se aplica o se ha de aplicar el formato automáticamente a los hipervínculos mientras se escribe.

### **2.1.1.21 AUTOMATIONSECURITY**

Devuelve o establece una constante de tipo MsoAutomationSecurity la cual especifica el modo de seguridad que utiliza una aplicación cuando mediante programación abre archivos.

Puede tener alguno de los siguientes valores:

**msoAutomationSecurityLow** (valor 1): Permite que todas las macros. Este es el valor predeterminado cuando se inicia la aplicación.

**msoAutomationSecurityByUI** (valor 2): Usa la configuración de seguridad especificadas en el cuadro de diálogo Seguridad.

**msoAutomationSecurityForceDisable** (valor 3): Desactiva todas las macros en todos los archivos abiertos mediante programación, sin mostrar ninguna alerta de seguridad.

#### 2.1.1.22 AUTOPERCENTENTRY

Devuelve o establece True si las entradas que tienen formato de porcentaje en las celdas no se han de multiplicar automáticamente por 100.

#### 2.1.1.23 AUTORECOVER

Devuelve un objeto AutoRecover. Dicho objeto representa las características de recuperación automática de un libro (Path: ruta de Autorrecuperación, Time: el intervalo de tiempo para realizar copias de seguridad todos los archivos, etc).

#### 2.1.1.24 BUILD

Devuelve el número de versión de Microsoft Excel.

#### 2.1.1.25 CALCULATEBEFORESAVE

Devuelve VERDADERO si los libros se calculan antes de que se guarden en disco.

#### 2.1.1.26 CALCULATION

Devuelve el módo de cálculo establecido.

Puede tener alguno de los siguientes valores:

**xlCalculationAutomatic** (-4105) Los cálculos se realizan automáticamente.

**xlCalculationManual** (-4135) Los cálculos se realizan manualmente.

**xlCalculationSemiautomatic** (2) Los cálculos se realizan automáticamente excepto en las tablas.

#### 2.1.1.27 CALCULATIONINTERRUPTKEY

Devuelve o define una constante que indica la tecla que puede interrumpir a Microsoft Excel cuando está realizando los cálculos. Dicha constante es **xlCalculationInterruptKey** y puede tomar los siguientes valores:

**xlAnyKey** (2) La actualización de los cálculos se interrumpe al presionar cualquier tecla.

**xlEscKey** (1) La actualización de los cálculos se interrumpe al presionar la tecla ESC.

**xlNoKey** (0) La actualización de los cálculos no se interrumpe al presionar ninguna tecla.

#### 2.1.1.28 CALCULATIONSTATE

Devuelve un valor que indica el si hay cálculos pendientes o no y corresponden a la constante **xlCalculationState** que puede tomar los siguientes valores:

**xlCalculating** (1) Los cálculos están en curso.

**xlDone** (0) Cálculos completados.

**xlPending** (2) Se han realizado cambios que desencadenan la actualización de los cálculos, pero todavía no se ha efectuado la actualización.

#### 2.1.1.29 CALCULATIONVERSION

Devuelve un número de la versión del motor de cálculo.

#### 2.1.1.30 CALLER

Devuelve información sobre la forma en que se llamó a Visual Basic.

Dicha llamada pudo ser realizada por una función o una macro.<sup>®</sup>

#### 2.1.1.31 CAPTION

Devuelve el título de la ventana principal de Excel.

### **2.1.1.32 CELLDragAndDrop**

Devuelve True si está permitido arrastrar y soltar celdas.

### **2.1.1.33 Cells**

Devuelve objeto Range con todas las celdas de la hoja activa.

### **2.1.1.34 Charts**

Devuelve un objeto con la colección de todas las hojas de tipo gráfico del libro activo.

### **2.1.1.35 ClipboardFormats**

Devuelve los diferentes formatos existentes en el portapapeles. Dichos formatos están enumerados en la constante **XlClipboardFormat**.

### **2.1.1.36 ClusterConnector**

Define o consulta el nombre del conector de clúster de sistemas HPC.

### **2.1.1.37 Columns**

Devuelve un objeto de tipo Range representando todas las columnas de la hoja activa.

### **2.1.1.38 COMAddIns**

Devuelve la colección de complementos COM (COMAddIns) instalados.

### **2.1.1.39 CommandBars**

Devuelve un objeto con las diferentes barras de comandos existentes.

### **2.1.1.40 CommandUnderlines**

Devuelve o establece el estado del subrayado de comandos para Macintosh.

### **2.1.1.41 ConstrainNumeric**

Devuelve un valor boolean indicando si el reconocimiento de escritura a mano está limitado a números y signos de puntuación.

#### 2.1.1.42 CONTROLCHARACTERS

Devuelve un valor boolean indicando si se muestran los caracteres de control para idiomas que se escriben de derecha a izquierda.

#### 2.1.1.43 COPYOBJECTSWITHCELLS

Devuelve un valor boolean indicando si los objetos se pueden cortar, copiar, extraer y ordenar con celdas.

#### 2.1.1.44 CREATOR

Devuelve el tipo de cursor mediante la constante `XlMousePointer` y puede tener alguno s los siguientes valores:

**`xlDefault`** (-4143) Puntero predeterminado.

**`xlIBeam`** (3) Puntero en forma de I.

**`xlNorthwestArrow`** (1) Puntero de flecha orientada al noroeste.

**`xlWait`** (2) Puntero de reloj de arena.

#### 2.1.1.45 CURSORMOVEMENT

Valor que determina si el cursor es

**Visual** (`xlVisualCursor`) o **lógico** (`xlLogicalCursor`).

#### 2.1.1.46 CUSTOMLISTCOUNT

Indica el número de listas personalizadas definidas.

#### 2.1.1.47 CUTCOPYMODE

Indica o define el estado del modo Cortar o Copiar que puede tener alguno de los valores: **true**, **false**, **`xlCopy`** (1) o **`xlCut`** (2).

#### 2.1.1.48 DATAENTRYMODE

Indica o define el modo de entrada de Datos que puede ser alguno de los siguientes:

**`xlOn`** Modo de entrada de Datos activado.

**`xlOff`** Modo de entrada de Datos desactivado.

**xlStrict** Modo de entrada de Datos activado y no se desactiva al pulsar ESCAPE.

En modo de entrada de Datos sólo se puede escribir en celdas desbloqueadas del rango seleccionado.

#### **2.1.1.49 DDEAPPRETURNCODE**

Código de devolución de DDE específico de la aplicación que contenía el mensaje de confirmación de DDE.

#### **2.1.1.50 DECIMALSEPARATOR**

Define carácter que se utiliza como separador decimal.

#### **2.1.1.51 DEFAULTFILEPATH**

Ruta de acceso predeterminada utilizada para abrir archivos.

#### **2.1.1.52 DEFAULTSAVEFORMAT**

Formato por defecto utilizado para guardar los archivos.

#### **2.1.1.53 DEFAULTSHEETDIRECTION**

Dirección por defecto en la que se muestran las ventanas y hojas de cálculo nuevas. Puede tener alguno de los siguiente svalores:

**xlRTL** (de derecha a izquierda)

**xlLTR** (de izquierda a derecha).

#### **2.1.1.54 DEFAULTWEBOPTIONS**

Devuelve objeto con atributos globales utilizados en una página web.

#### **2.1.1.55 DEFERASYNCQUERIES**

Indica o define si se ejecutan las consultas asincrónicas a orígenes de Datos OLAP cuando la actualización se realiza mediante código de VBA.

#### **2.1.1.56 DIALOGS**

Devuelve colección de Dialogs que representa todos los cuadros de diálogo integrados.



### **2.1.1.57 DISPLAYALERTS**

Permite indicar o averiguar si los mensajes que aparecen durante la ejecución de una macro se han de visualizar o no. En caso de que un mensaje solicite una respuesta, en caso de definir esta variable como False, se elegirá la respuesta predeterminada.

### **2.1.1.58 DISPLAYCLIPBOARDWINDOW**

Indica o define si es posible mostrar el portatapeles.

### **2.1.1.59 DISPLAYCOMMENTINDICATOR**

Indica o define como se muestran los indicadores y comentarios. Sus valores pueden ser:

**xlCommentAndIndicator** (1) Mostrar comentario e indicador.

**xlCommentIndicatorOnly** (-1) Mostrar sólo el indicador de comentario. Muestra comentario el puntero del mouse pasa sobre la celda.

**xlNoIndicator** (0) No mostrar ni comentario ni indicador de comentario.

### **2.1.1.60 DISPLAYDOCUMENTACTIONTASKPANE**

Indica o define si se muestra el panel de tareas Acciones de Documentos.

### **2.1.1.61 DISPLAYDOCUMENTINFORMATIONPANEL**

Indica o define si se muestra el panel de propiedades del Documentos.

### **2.1.1.62 DISPLAYEXCEL4MENUS**

Indica o define si se muestran las barras de menú de la versión 4.0.

### **2.1.1.63 DISPLAYFORMULAAUTOCOMPLETE**

Indica o define si se muestra una lista de funciones y nombres definidos al generar fórmulas.

### **2.1.1.64 DISPLAYFORMULABAR**

Indica o define si se muestra la barra de fórmulas.

### **2.1.1.65 DISPLAYFULLSCREEN**

Indica o define si se muestra la pantalla completa.

### **2.1.1.66 DISPLAYFUNCTIONTOOLTIPS**

Indica o define la posibilidad de mostrar información sobre herramientas.

### **2.1.1.67 DISPLAYINSERTOPTIONS**

Indica o define si se muestra el botón de Opciones de inserción.

### **2.1.1.68 DISPLAYNOTEINDICATOR**

Indica o define si se han de mostrar los indicadores en las celdas que contienen notas.

### **2.1.1.69 DISPLAYPASTEOPTIONS**

Indica o define si es posible mostrar el botón de Opciones de pegado.

### **2.1.1.70 DISPLAYRECENTFILES**

Indica o define si se ha de mostrar la lista de ficheros usados recientemente.

### **2.1.1.71 DISPLAYSCROLLBARS**

Indica o define si se han de mostrar las barras de desplazamiento.

### **2.1.1.72 DISPLAYSTATUSBAR**

Indica o define si se ha la barra de estado.

### **2.1.1.73 EDITDIRECTLYINCELL**

Indica o define si se pueden editar las celdas directamente.

### **2.1.1.74 ENABLEANIMATIONS**

Indica o define si la animación de inserción o eliminación de filas está activa.

### **2.1.1.75 ENABLEAUTOCOMPLETE**

Indica o define si está habilitada la característica de Autocompletar.

### **2.1.1.76 ENABLECANCELKEY**

Define cómo controlar la interrupción de un procedimiento. La interrupción puede provocarse con CTRL+INTER o con ESC.

Los valores que pueden utilizarse son:

**xlDisabled** (0) Interrupción deshabilitada.

**xlErrorHandler** (2) La interrupción se trata como un error, y se puede gestionar con On Error GoTo. El código de error es 18.

**xlInterrupt** (1) El procedimiento se puede interrumpir.

### 2.1.1.77 ENABLEEVENTS

Indica o determina si los eventos están habilitados.

### 2.1.1.78 ENABLELARGEOPERATIONALERT

Indica si se ha de mostrar un aviso cuando el número de celdas afectadas por una determinada operación supera el especificado en la interfaz de usuario central de Office.

### 2.1.1.79 ENABLELIVEPREVIEW

Indica o determina si el sonido está habilitado.

### 2.1.1.80 ERRORCHECKINGOPTIONS

Objeto que representa las opciones de comprobación de errores de una aplicación. Dicho objeto posee las siguientes propiedades:

- BackgroundChecking
- EmptyCellReferences
- EvaluateToError
- InconsistentFormula
- IndicatorColorIndex
- NumberAsText
- OmittedCells
- TextDate
- UnlockedFormulaCells

### 2.1.1.81 EXCEL4INTLMACROSHEETS

Devuelve una colección de hojas de macros internacionales de MS Excel 4.0 de libro indicado.

### 2.1.1.82 EXCEL4MACROSHEETS

Devuelve una colección de hojas de macros de MS Excel 4.0 de libro indicado.

### 2.1.1.83 EXTENDLIST

Indica o determina si se han de aplicar las formulas y formatos a los datos nuevos que se incluyan en una lista.

### 2.1.1.84 FEATUREINSTALL

Indica o determina como se han de controlar las llamadas a métodos y propiedades que usan características que aun no se han instalado. Los valores posibles corresponden a la constante `MsoFeatureInstall` y son:

**`msoFeatureInstallNone`**. Constante predeterminada. Genera un error genérico en tiempo de ejecución.

**`msoFeatureInstallOnDemand`**. Solicita instalación de las nuevas características.

**`msoFeatureInstallOnDemandWithUI`**. Muestra un indicador de progreso durante la instalación.

### 2.1.1.85 FILECONVERTERS

Informa sobre los convertidores de archivos instalados.

### 2.1.1.86 FILEDIALOG

Devuelve un objeto en función de la constante `MsoFileDialogType` la cual permite invocar alguno de los siguientes diálogos:

**`msoFileDialogFilePicker`**. Cuadro de diálogo seleccionar archivo.

**`msoFileDialogFolderPicker`**. Cuadro de diálogo seleccionar carpeta.

**`msoFileDialogOpen`**. Cuadro de diálogo abrir archivo.

**`msoFileDialogSaveAs`**. Cuadro de diálogo guardar archivo.

### **2.1.1.87 FILEEXPORTCONVERTERS**

Devuelve colección de convertidores de archivo disponibles para MS Excel.

### **2.1.1.88 FILEVALIDATION**

Indica o determina cómo se validan los archivos antes de su apertura.

### **2.1.1.89 FILEVALIDATIONPIVOT**

Indica o determina cómo se valida el contenido de la caché de datos para informes de tabla dinámica.

### **2.1.1.90 FINDFORMAT**

Indica o determina los criterios de búsqueda de tipo de formatos de celda a buscar.

### **2.1.1.91 FIXEDDECIMAL**

Si se define como True, provoca que todos los valores numéricos pasen a tener el número de decimales indicado en la propiedad FixedDecimalPlaces.

### **2.1.1.92 FIXEDDECIMALPLACES**

Indica o determina el número de decimales que tendrá el valor que se introduzca después de haber activado a True la propiedad FixedDecimal.

### **2.1.1.93 FORMULABARHEIGHT**

Indica o determina la altura de la barra de fórmulas en número de líneas.

### **2.1.1.94 GENERATEGETPIVOTDATA**

Indica o determina si se pueden obtener datos de un informe de tabla dinámica.

### **2.1.1.95 GENERATETABLEREFS**

Indica o determina qué método se utiliza para hacer referencia a tablas dentro de fórmulas. Se trata del valor **Usar nombres de tablas en las fórmulas** (apartado **Trabajando con fórmulas** en **Opciones de Excel->Fórmulas**).

Puede utilizarse alguna de las siguientes constantes:

**xlGenerateTableRefStruct**

**xlGenerateTableRefA1**

#### **2.1.1.96 HEIGHT**

Indica o determina el alto de la ventana principal en puntos.

#### **2.1.1.97 HIGHQUALITYMODEFORGRAPHICS**

Indica o determina si se utiliza el modo de alta calidad para la impresión de gráficos.

#### **2.1.1.98 HINSTANCE**

Devuelve un controlador de tipo Long a la instancia de MS 20210 representada por application.

#### **2.1.1.99 HINSTANCEPTR**

Devuelve un controlador de tipo Variant a la instancia de MS 20210 representada por application.

#### **2.1.1.100 HWND**

Devuelve identificador de la ventana superior de MS Excel.

#### **2.1.1.101 IGNOREREMOTEREQUESTS**

Indica o determina si se han de ignorar las peticiones DDE remotas.

#### **2.1.1.102 INTERACTIVE**

Indica o determina si se ha de bloquear cualquier entrada desde el teclado o mouse excepto la que se realice a través de los cuadros de diálogo mostrados en el método.

Ra-Ma®

### 2.1.1.103 INTERNATIONAL

Devuelve información sobre la configuración regional e internacional. Por ejemplo:

**xlDateSeparator** (separador de fecha “/”).

**xlCurrencyCode** (símbolo de la moneda).

**xlDecimalSeparator** (separador decimal).

### 2.1.1.104 ISSANDBOXED

Indica si el libro está abierto en una ventana Vista protegida.

### 2.1.1.105 ITERATION

Indica o determina si se ha de usar una iteración para resolver referencias circulares.

### 2.1.1.106 LANGUAGESETTINGS

Devuelve objeto LanguageSettings que contiene información sobre la configuración del idioma.

### 2.1.1.107 LARGEOPERATIONCELLTHOUSANDCOUNT

Indica o determina el número máximo de celdas para que se desencadene un aviso al realizar una operación.

### 2.1.1.108 LEFT

Indica o determina la distancia en puntos existente entre el borde izquierdo de la pantalla hasta el borde izquierdo de la ventana principal de MS Excel.

### 2.1.1.109 LIBRARYPATH

Devuelve el path de la carpeta Library. Por ejemplo:

**C:\Program Files\Microsoft Office\Office14\LIBRARY**

### 2.1.1.110 MAILSESSION

Devuelve valor en hexadecimal que representa el número de la sesión de correo MAPI.

### 2.1.1.111 MAILSYSTEM

Indica tipo de sistema de correo instalado en el equipo. La constante utilizada (XlMailSystem) tiene los siguientes valores:

**xlMAPI** (Microsoft Mail)

**xlNoMailSystem** (No hay sistema mail)

**xlPowerTalk** (PowerTalk)

### 2.1.1.112 MAPPAPERSIZE

Indica o determina si los documentos con formato de otro país o región se ajustan automáticamente al tamaño del papel estándar del país del usuario (p.e. A4).

### 2.1.1.113 MATHCOPROCESSORAVAILABLE

Indica si se dispone de un coprocesador matemático.

### 2.1.1.114 MAXCHANGE

Indica o determina cambio máximo entre iteraciones en la resolución de referencias circulares.

### 2.1.1.115 MAXITERATIONS

Indica o determina el número máximo de iteraciones a realizar en referencias circulares.

### 2.1.1.116 MEASUREMENTUNIT

Indica o determina la unidad de medida que se utiliza en la aplicación. La constante utilizada es XlMeasurementUnits y tiene los siguientes valores:

**xlCentimeters** (1) Centímetros

**xlInches** (0) Pulgadas

**xlMillimeters** (2) Milímetros

### 2.1.1.117 MOUSEAVAILABLE

Indica si el mouse está disponible.



### **2.1.1.118 MOVEAFTERRETURN**

Indica o determina si la celda activa se desplazará al presionar la tecla **ENTER**.

### **2.1.1.119 MOVEAFTERRETURNIRECTION**

Indica o determina el sentido del desplazamiento del cursor al presionar la tecla **ENTER**. Puede utilizarse alguno de los siguientes valores:

**xlDown** (-4121) Hacia abajo.

**xlToLeft** (-4159) Hacia la izquierda.

**xlToRight** (-4161) Hacia la derecha.

**xlUp** (-4162) Hacia arriba.

### **2.1.1.120 MULTITHREADED CALCULATION**

Devuelve objeto `MultiThreadedCalculation` que controla la configuración de los cálculos multiproceso.

### **2.1.1.121 NAME**

Devuelve nombre del objeto.

### **2.1.1.122 NAMES**

Devuelve colección con todos los nombres del libro activo.

### **2.1.1.123 NETWORKTEMPLATESPATH**

Devuelve la ruta de red en la que se almacenan las plantillas.

### **2.1.1.124 NEWWORKBOOK**

Devuelve objeto `NewFile`.

### **2.1.1.125 ODBCERRORS**

Devuelve colección de errores ODBC generados por la consulta más reciente.

### **2.1.1.126 ODBCETIMEOUT**

Indica o determina el tiempo límite en segundos que una consulta ODBC puede esperar. Por defecto son 45 segundos.

### **2.1.1.127 OLEDBERRORS**

Devuelve colección de errores OLEDB generados por la consulta más reciente.

### **2.1.1.128 ONWINDOW**

Indica o determina el nombre del procedimiento que se ejecuta al activar una ventana.

### **2.1.1.129 OPERATINGSYSTEM**

Devuelve el nombre y versión del sistema operativo actual. Por ejemplo:

**Windows (64-bit) NT 6.01**

### **2.1.1.130 ORGANIZATIONNAME**

Devuelve el nombre de la organización.

### **2.1.1.131 PARENT**

Devuelve el objeto primario.

### **2.1.1.132 PATH**

Devuelve la ruta de acceso de la aplicación.

### **2.1.1.133 PATHSEPARATOR**

Devuelve el carácter utilizado como separador de la ruta de acceso ("").

### **2.1.1.134 PIVOTTABLESELECTION**

Indica o determina si los informes de una tabla dinámica utilizan una selección estructurada.

### **2.1.1.135 PREVIOUSSELECTIONS**

Devuelve los últimos 4 rangos o nombres seleccionados mediante el comando Ir o usando el cuadro Nombre.

**2.1.1.136 PRINTCOMMUNICATION**

Indica o determina si hay comunicación con la impresora.

**2.1.1.137 PRODUCTCODE**

Devuelve identificador único de MS Excel.

**2.1.1.138 PROMPTFORSUMMARYINFO**

Indica o determina si la primera vez que se guardan los archivos se solicita información de resumen.

**2.1.1.139 PROTECTEDVIEWWINDOWS**

Devuelve una colección de objetos que representan las Vistas protegidas actualmente abiertas en la aplicación.

**2.1.1.140 RANGE**

Devuelve un rango.

**2.1.1.141 READY**

Indica si la aplicación está preparada o no.

**2.1.1.142 RECENTFILES**

Devuelve la lista de archivos usados recientemente.

**2.1.1.143 RECORDRELATIVE**

Indica si las macros se graban con referencias relativas o absolutas.

**2.1.1.144 REFERENCESTYLE**

Indica o determina el estilo de referencia A1 o F1C1 a utilizar. La constante utilizada es `XlReferenceStyle` y puede tener los siguientes valores:

`xlA1`

`xlR1C1`

Ra-Ma®

### **2.1.1.145 REGISTEREDFUNCTIONS**

Devuelve información sobre las DLL o recursos registrados mediante REGISTER o REGISTER.ID.

### **2.1.1.146 REPLACEFORMAT**

Define los criterios a utilizar para remplazar formatos de celda mediante el método Replace de los objetos de tipo Range.

### **2.1.1.147 ROLLZOOM**

Indica o determina como funciona Intellimouse del ratón (rueda central) para que aleje o acerque o bien, se desplace.

### **2.1.1.148 ROWS**

Devuelve objeto de tipo Range con las filas de la hoja de cálculo activa.

### **2.1.1.149 RTD**

Devuelve objeto RTD (RealTimeData) el cual permite obtener información en tiempo real.

### **2.1.1.150 SAVEISO8601DATES**

Indica o determina si Excel utiliza el formato ISO 8601 para guardar fechas y horas.

### **2.1.1.151 SCREENUPDATING**

Indica o determina si está activada la actualización de pantalla.

### **2.1.1.152 SELECTION**

Devuelve el objeto que se halle seleccionado en la ventana activa.

### **2.1.1.153 SHEETS**

Devuelve una colección de hojas pertenecientes al libro activo.

### **2.1.1.154 SHEETSINNEWWORKBOOK**

Indica o determina el número de hojas a insertar cada vez que se crea un nuevo libro.

### **2.1.1.155 SHOWCHARTTIPNAMES**

Indica o determina si se han de mostrar los nombres de las sugerencias de los gráficos.

### **2.1.1.156 SHOWCHARTTIPVALUES**

Indica o determina si se han de mostrar los valores de las sugerencias de los gráficos.

### **2.1.1.157 SHOWDEVTOOLS**

Indica o determina si se ha de mostrar la ficha de Programador en la cinta de opciones.

### **2.1.1.158 SHOWMENUFLOATIES**

Indica o determina si se han de mostrar las barras de herramientas flotantes al pulsar el botón derecho del mouse.

### **2.1.1.159 SHOWSELECTIONFLOATIES**

Indica o determina si se han de mostrar las barras de herramientas flotantes al pulsar el botón derecho del mouse cuando el usuario haya seleccionado texto.

### **2.1.1.160 SHOWSTARTUPDIALOG**

Indica o determina si se ha de mostrar el panel de tareas Libro nuevo cada vez que se cree un nuevo libro.

### **2.1.1.161 SHOWTOOLTIPS**

Indica o determina si se han de mostrar los Tool Tips (información sobre herramientas).

### **2.1.1.162 SHOWWINDOWSINTASKBAR**

Indica o determina si cada libro se visualiza en la barra de tareas como un botón.

### **2.1.1.163 SMARTARTCOLORS**

Devuelve conjunto de estilos de colores cargado en la aplicación.

### **2.1.1.164 SMARTARTLAYOUTS**

Devuelve el conjunto de diseños SmartArt cargados en la aplicación.

### **2.1.1.165 SMARTARTQUICKSTYLES**

Devuelve el conjunto de estilos rápidos de SmartArt cargados en la aplicación.

### **2.1.1.166 SPEECH**

Devuelve un valor de tipo Speech.

### **2.1.1.167 SPELLINGOPTIONS**

Devuelve objeto que representa las opciones de ortografía. Algunas de dichas opciones son: IgnoreCaps, IgnoredMixedDigits, etc.

### **2.1.1.168 STANDARDFONT**

Indica o determina el nombre de la fuente estándar.

### **2.1.1.169 STANDARDFONTSIZE**

Indica o determina el tamaño de la fuente estándar.

### **2.1.1.170 STARTUPPATH**

Devuelve la ruta de la carpeta de inicio.

### **2.1.1.171 STATUSBAR**

Indica o determina el texto de la barra de estado.

### **2.1.1.172 TEMPLATESPATH**

Devuelve la ruta en la que se almacenan las plantillas.

### **2.1.1.173 THISCELL**

Devuelve un objeto Range que contiene la celda desde la que se llama a la función definida por el usuario.

### **2.1.1.174 THISWORKBOOK**

Devuelve un objeto Workbook con el libro en el que se está ejecutando el código de la actual macro.

### **2.1.1.175 THOUSANDSSEPARATOR**

Indica o determina el carácter separador de miles.

### **2.1.1.176 TOP**

Indica o determina la distancia en puntos entre el borde superior de la pantalla y el borde superior de la ventana principal de MS Excel.

### **2.1.1.177 TRANSITIONMENUKEY**

Indica o determina la tecla de menú o de ayuda de MS Excel, que suele ser "/".

### **2.1.1.178 TRANSITIONMENUKEYACTION**

Indica o determina la acción que se realiza al presionar la tecla de menú de MS Excel.

### **2.1.1.179 TRANSITIONNAVIGKEYS**

Indica o determina si las teclas de desplazamiento de transición están activadas.

### **2.1.1.180 USABLEHEIGHT**

Devuelve el alto máximo en puntos que puede ocupar una ventana en el área de ventana de la aplicación.

### **2.1.1.181 USABLEWIDTH**

Devuelve el ancho máximo en puntos que puede ocupar una ventana en el área de ventana de la aplicación.

### **2.1.1.182 USECLUSTERCONNECTOR**

Indica o determina si MS Excel permite que las funciones de usuario definidas en los complementos XLL se ejecuten en un clúster de cálculo.

### **2.1.1.183 USEDOBJECTS**

Devuelve objeto con los objetos asignados de un libro. Dicho objeto es de tipo UsedObjects.

### **2.1.1.184 USERCONTROL**

Indica o determina si el usuario la ha creado o iniciado o la aplicación está visible.

**2.1.1.185 USERLIBRARYPATH**

Devuelve la ruta de acceso en la que están instalados los complementos COM.

**2.1.1.186 USERNAME**

Indica o determina el nombre de usuario actual.

**2.1.1.187 USESYSTEMSEPARATORS**

Indica o determina si los separadores del sistema de MS Excel están habilitados.

**2.1.1.188 VALUE**

Devuelve el nombre de la aplicación.

**2.1.1.189 VBE**

Devuelve un objeto de tipo VBE que representa el Editor de Visual Basic.

**2.1.1.190 VERSION**

Devuelve el número de versión de MS Excel.

**2.1.1.191 VISIBLE**

Indica o determina si el objeto está visible.

**2.1.1.192 WARNONFUNCTIONNAMECONFLICT**

Indica o determina si se ha de dar aviso cuando el usuario intente crear una función que ya exista.

**2.1.1.193 WATCHES**

Devuelve objeto de tipo Watches el cual refleja el rango que se inspecciona en la Ventana de inspección.

**2.1.1.194 WIDTH**

Indica o determina la distancia en puntos entre el borde izquierdo y el derecho de ventana de la aplicación.



### **2.1.1.195 WINDOWS**

Devuelve una colección que contiene todas las ventanas de todos los libros abiertos.

### **2.1.1.196 WINDOWSFORPENS**

Indica si el equipo ejecuta MS Windows para Pen Computing.

### **2.1.1.197 WINDOWSTATE**

Indica o determina el estado de la ventana. Se define mediante la constante `XlWindowState` y puede tener alguno de los valores siguientes:

**`xlMaximized`** (-4137) Maximizado

**`xlMinimized`** (-4140) Minimizado

**`xlNormal`** (-4143) Normal

### **2.1.1.198 WORKBOOKS**

Devuelve una colección con todos los libros abiertos.

### **2.1.1.199 WORKSHEETFUNCTION**

Devuelve un objeto que permite invocar a cualquier función de MS Excel. Por ejemplo:

**`miApp.WorksheetFunction.Average(10, 15)`**

Devuelve como resultado 20.

### **2.1.1.200 WORKSHEETS**

Devuelve una colección de hojas correspondientes al libro activo.

Ra-Ma®

## 2.1.2 Métodos

### 2.1.2.1 ACTIVATEMICROSOFTAPP

#### 2.1.2.1.1 Descripción

Permite iniciar o activar una aplicación de Microsoft.

#### 2.1.2.1.2 Sintaxis:

ActivateMicrosoftApp(Indice)

#### 2.1.2.1.3 Argumentos

Argumento	Comentario
Indice	Aplicación de Microsoft que se desea activar.

#### 2.1.2.1.4 Ejemplo

Application.ActivateMicrosoftApp (xlMicrosoftPowerPoint)

Este ejemplo abre la aplicación de MS PowerPoint.

### 2.1.2.2 ADDCUSTOMLIST

#### 2.1.2.2.1 Descripción

Añade lista personalizada para Autorrelleno y para ordenación personalizados.

#### 2.1.2.2.2 Sintaxis:

AddCustomList(ListArray, ByRow)

Ra-Ma®

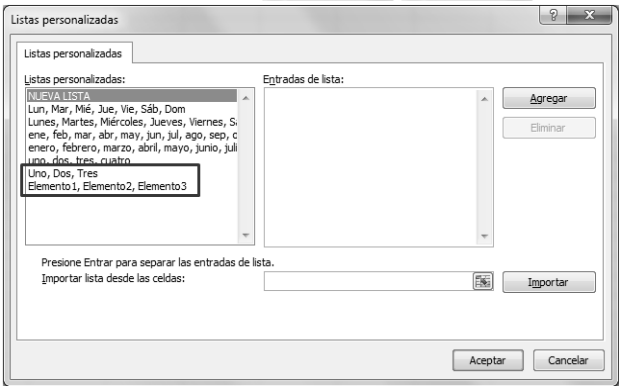
2.1.2.2.3 Argumentos

Argumento	Comentario
ListArray	Datos a incluir en la lista. Puede utilizarse un array o indicar un rango. Es un dato obligatorio.
ByRow	<p>Solo utilizado si la lista se fabrica mediante un rango.</p> <p>Si indicamos True la lista se crea a partir de filas. False, la crea a partir de columnas.</p> <p>Por defecto, si no se indica nada dependerá del número de filas y de columnas. Si hay mas filas que columnas, se crea a partir de las columnas. Si hay mas columnas que filas, se crea a partir de las filas.</p>

2.1.2.2.4 Ejemplo

```
Application.AddCustomList Array("Uno", "Dos", "Tres")
Application.AddCustomList Range("A1:A3")
```

Puede observarse que quedan incluidas como listas personalizadas.



2.1.2.3 CALCULATE

2.1.2.3.1 Descripción

Permite calcular un rango determinado, una hoja concreta o todos los libros abiertos.

### 2.1.2.3.2 Sintaxis:

Calculate

### 2.1.2.3.3 Argumentos

No tiene

### 2.1.2.3.4 Ejemplo

```
Worksheets("MiHoja").Calculate  
Worksheets("MiHoja").UsedRange.Range("A5:A7").Calculate
```

En el primer ejemplo, calcula el rango indicado realizando cálculos pendientes si los hubiera. En el segundo recalcula la hoja llamada “MiHoja”.

## 2.1.2.4 CALCULATEFULL

### 2.1.2.4.1 Descripción

Recalcula todos los libros abiertos.

### 2.1.2.4.2 Sintaxis:

CalculateFull

### 2.1.2.4.3 Argumentos

No tiene

### 2.1.2.4.4 Ejemplo

```
Application.CalculateFull
```

Se realiza un recálculo de todos los libros que se hallen abiertos en el momento de la ejecución.

## 2.1.2.5 CALCULATEFULLREBUILD

### 2.1.2.5.1 Descripción

Para todos los libros abiertos, realiza un recálculo de todos los datos y restablece las dependencias.

### 2.1.2.5.2 Sintaxis:

CalculateFullRebuild

### 2.1.2.5.3 Argumentos

No tiene

### 2.1.2.5.4 Ejemplo

```
Application.CalculateFullRebuild
```

Es idéntico a **CalculateFull** pero estableciendo las dependencias de nuevo.

## 2.1.2.6 CALCULATEUNTILASYNCQUERIESDONE

### 2.1.2.6.1 Descripción

Devuelve las consultas pendientes a OLEDB y OLAP.

### 2.1.2.6.2 Sintaxis:

CalculateUntilAsyncQueriesDone

### 2.1.2.6.3 Argumentos

No tiene

### 2.1.2.6.4 Ejemplo

```
miAplicacion.CalculateUntilAsyncQueriesDone
```

Devuelve todas las consultas pendientes.

## 2.1.2.7 CENTIMETERSTOPOINTS

### 2.1.2.7.1 Descripción

Convierte centímetros a puntos (un punto = 0,035 centímetros)..

### 2.1.2.7.2 Sintaxis:

CentimetersToPoints(Centimeters)

### 2.1.2.7.3 Argumentos

Argumento	Comentario
Centimeters	Especifica el valor en centímetros que se convertirá a puntos.

### 2.1.2.7.4 Ejemplo

Código
Debug.Print Application.CentimetersToPoints(1)
Resultado
<div>Inmediato</div> <div>28,3464566929134</div>

### 2.1.2.8 CHECKABORT

#### 2.1.2.8.1 Descripción

Detiene el proceso de actualización permitiendo indicar como excepción un rango el cual si se actualizará.

#### 2.1.2.8.2 Sintaxis:

CheckAbort(KeepAbort)

#### 1.1.1.3.1 Argumentos

Argumento	Comentario
KeepAbort	Permite actualizar un objeto Range.

### 2.1.2.8.3 Ejemplo

Código
<pre>Dim miRango As Range Set miRango = Application.Range("A7") Application.CheckAbort (miRango)</pre>

Ra-Ma®

## 2.1.2.9 CHECKSPELLING

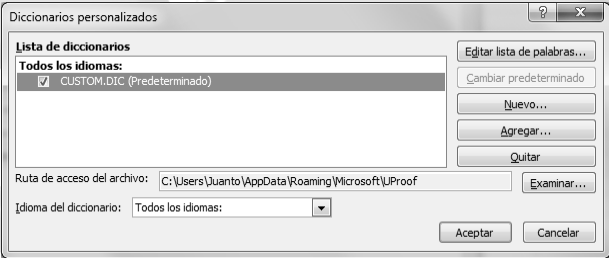
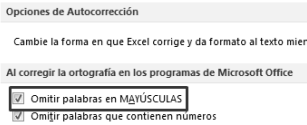
### 2.1.2.9.1 Descripción

Comprueba la ortografía de una palabra.

### 2.1.2.9.2 Sintaxis:

CheckSpelling(Word, CustomDictionary, IgnoreUppercase)

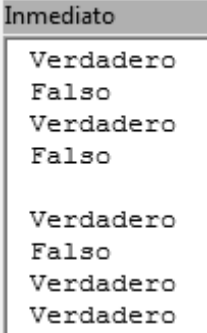
### 2.1.2.9.3 Argumentos

Argumento	Comentario
Word	Palabra a comprobar.
CustomDictionary	<p>Archivo que contiene el diccionario personalizado. Si no se indica nada se usa el diccionario especificado en ese momento. Puede revisar los diccionarios personalizados que posee en <b>Opciones -&gt; Revisión:</b></p> 
IgnoreUppercase	<p>Puede indicar True o False. Si se indica True no se revisan las palabras que contengan todas las letras en mayúsculas. Por defecto se usará la configuración actual indicada en <b>Opciones -&gt; Revisión:</b></p> 

Ra-Ma<sup>®</sup>

### 2.1.2.9.4 Ejemplo

Application.metodo ejemplo

Código
<pre> Debug.Print Application.CheckSpelling("Juan", "", False) Debug.Print Application.CheckSpelling("Juano", "", False) Debug.Print Application.CheckSpelling("JUAN", "", False) Debug.Print Application.CheckSpelling("JUANO", "", False) Debug.Print Debug.Print Application.CheckSpelling("Juan", "", True) Debug.Print Application.CheckSpelling("Juano", "", True) Debug.Print Application.CheckSpelling("JUAN", "", True) Debug.Print Application.CheckSpelling("JUANO", "", True) </pre>
Resultado


### 2.1.2.10 CONVERTFORMULA

#### 2.1.2.10.1 Descripción

Convierte las referencias a celdas dentro de una fórmula.

#### 2.1.2.10.2 Sintaxis:

ConvertFormula(Formula, FromReferenceStyle, ToReferenceStyle, ToAbsolute, RelativeTo)

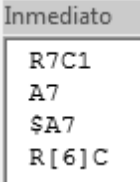
Ra-Ma®



### 2.1.2.10.3 Argumentos

Argumento	Comentario
Formula	Fórmula a convertir (empieza con el simbolo igual '=')
FromReferenceStyle	<p>Estilo de referencia que tiene la fórmula que se desea convertir. Los posibles estilos son: <b>xlA1</b> (p.e. "A7") y <b>xlR1C1</b> (p.e. "R7C1")</p> <p><b>XlReferenceStyle</b></p> <p>xlA1 = 1</p> <p>xlR1C1 = -4150</p>
ToReferenceStyle	Estilo de referencia que ha de tener la fórmula una vez convertida.
ToAbsolute	<p>Cconstante que especifica el tipo de la referencia convertida.</p> <p><b>XlReferenceType</b></p> <p>xlAbsolute = 1</p> <p>xlAbsRowRelColumn = 2</p> <p>xlRelative = 4</p> <p>xlRelRowAbsColumn = 3</p>
RelativeTo	Rango con una sola celda al que hacen referencia las referencias relativas.

2.1.2.10.4 Ejemplo

Código
<pre>Dim RefCell As Range Set RefCell = Application.Range("A1")  Debug.Print Application.ConvertFormula("A7", xlA1, xlR1C1, xlAbsolute) Debug.Print Application.ConvertFormula("R7C1", xlR1C1, xlA1, xlRelative) Debug.Print Application.ConvertFormula("R7C1", xlR1C1, xlA1, xlRelRowAbsColumn) Debug.Print Application.ConvertFormula("A7", xlA1, xlR1C1, , RefCell)</pre>
Resultado


2.1.2.11 DDEEXECUTE

2.1.2.11.1 Descripción

Ejecuta un comando o acción en otra aplicación utilizando DDE (Dynamic Data Exchange).

2.1.2.11.2 Sintaxis:


DDEExecute(Canal, String)

2.1.2.11.3 Argumentos

Argumento	Comentario
Canal	El número devuelto por DDEInitiate.
String	El mensaje a enviar indicando la acción a realizar.

#### 2.1.2.11.4 Ejemplo

En el siguiente ejemplo se muestra como lanzar un aviso mediante el comando ALERT a través de DDE.

Código
<pre>Dim canal As Integer canal = Application.DDEInitiate("Excel", "System")  DDEExecute canal, "[ALERT(" &amp; Chr(34) &amp; "Hola Juanto" &amp; Chr(34) &amp; ")]" DDEExecute canal, "[EXIT]" DDETerminate canal</pre>
Resultado


#### 2.1.2.12 DDEINITIATE

##### 2.1.2.12.1 Descripción

Prepara un canal de DDE para intercambiar datos con una aplicación.

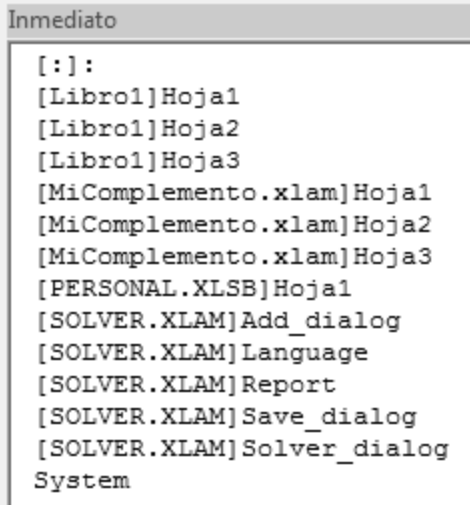
##### 2.1.2.12.2 Sintaxis:

DDEInitiate(Aplicacion, Tema)

##### 2.1.2.12.3 Argumentos

Argumento	Comentario
Aplicacion	Aplicación.
Tema	Elemento de la aplicación con el que se va a trabajar.

#### 2.1.2.12.4 Ejemplo

Código
<pre>Sub prueba_DDE()     Dim canal As Integer     canal = Application.DDEInitiate(app:="Excel", topic:="System")     Dim lElementos As Variant     lElementos = DDERequest(canal, "Topics")      Dim i As Integer     For i = LBound(lElementos) To UBound(lElementos)         Debug.Print lElementos(i)     Next i     DDEExecute canal, "[EXIT]"     DDETerminate canal End Sub</pre>
Resultado


#### 2.1.2.13 DDEPOKE

##### 2.1.2.13.1 Descripción

Envía datos a la aplicación previamente iniciada con DDEInitiate.

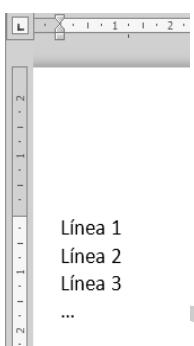
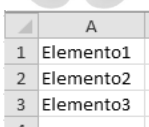
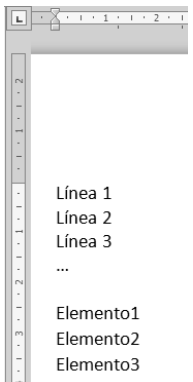
##### 2.1.2.13.2 Sintaxis:

DDEPoke(Canal, Elemento, Datos)

### 2.1.2.13.3 Argumentos

Argumento	Comentario
Canal	Canal obtenido mediante DDEInitiate.
Elemento	Elemento que recibirá los datos.
Datos	Datos a enviar.

### 2.1.2.13.4 Ejemplo

Código		
<pre> Sub prueba_DDEPoke()     Dim canal As Integer     canal = Application.DDEInitiate(app:="WinWord",     topic:="C:\TEMP\JUANTO.DOC")      Dim miRango As Range     Set miRango = Range("A1:A3")      Application.DDEPoke canal, "\EndOfDoc", miRango     Application.DDETerminate canal End Sub </pre>		
Resultado		
		

En este ejemplo, se añade al contenido del rango A1:A3 al final del documento JUANTO.DOC.

### 2.1.2.14 DDEREQUEST

#### 2.1.2.14.1 Descripción

Solicita información de la aplicación especificada. Este método siempre devuelve una matriz.

#### 2.1.2.14.2 Sintaxis:

DDERequest(Canal, Elemento)

#### 2.1.2.14.3 Argumentos

Argumento	Comentario
Canal	Canal obtenido por DDEInitiate.
Elemento	Elemento solicitado.

#### 2.1.2.14.4 Ejemplo

Código
<pre>Sub prueba_DDERequest()     Dim canal As Integer     canal = Application.DDEInitiate(app:="Excel", topic:="System")     Dim lElementos As Variant     lElementos = DDERequest(canal, "Topics")      Dim i As Integer     For i = LBound(lElementos) To UBound(lElementos)         Debug.Print lElementos(i)     Next i     DDEExecute canal, "[EXIT]"     DDETerminate canal End Sub</pre>

Ra-Ma®

Resultado	
	<div>Inmediato</div> <pre>[:]: [Libro1]Hoja1 [Libro1]Hoja2 [Libro1]Hoja3 [MiComplemento.xlam]Hoja1 [MiComplemento.xlam]Hoja2 [MiComplemento.xlam]Hoja3 [SOLVER.XLAM]Add_dialog [SOLVER.XLAM]Language [SOLVER.XLAM]Report [SOLVER.XLAM]Save_dialog [SOLVER.XLAM]Solver_dialog System</pre>

### 2.1.2.15 DDETERMINE

#### 2.1.2.15.1 Descripción

Finaliza sesión DDE cerrando el canal abierto por DDEInitiate.

#### 2.1.2.15.2 Sintaxis:

DDETerminate(Canal)

#### 2.1.2.15.3 Argumentos

Argumento	Comentario
Canal	El número de canal que devuelve el método DDEInitiate y que es el que vamos a cerrar.

#### 2.1.2.15.4 Ejemplo

Código
<pre>Sub prueba_DDEExecute()     Dim canal As Integer     canal = Application.DDEInitiate("Excel", "System")     DDEExecute canal, "[ALERT(" &amp; Chr(34) &amp; "Hola Juanto" &amp; Chr(34)     &amp; ")]"     DDEExecute canal, "[EXIT]"     DDETerminate canal End Sub</pre>

## 2.1.2.16 DELETECUSTOMLIST

### 2.1.2.16.1 Descripción

Permite eliminar una lista personalizada.

### 2.1.2.16.2 Sintaxis:

DeleteCustomList(ListNum)

### 2.1.2.16.3 Argumentos

Argumento	Comentario
ListNum	Número de lista personalizada a borrar.

### 2.1.2.16.4 Ejemplo

Código
<pre>Sub muestra_DeleteCustomList()     Dim lElementos As Variant     Dim numListas As Integer     numListas = Application.CustomListCount     Dim i, n As Integer     Dim texto As String     For n = 1 To numListas         lElementos = Application.GetCustomListContents(n)         texto = ""         For i = LBound(lElementos) To UBound(lElementos)             If texto &lt;&gt; "" Then texto = texto &amp; ", "             texto = texto &amp; lElementos(i)         Next i         Debug.Print "Lista num. " &amp; n &amp; ": " &amp; texto     Next n     Application.DeleteCustomList 6 End Sub</pre>
Resultado
<pre>Inmediato Lista num. 1: Lun,Mar,Mié,Jue,Vie,Sáb,Dom Lista num. 2: Lunes,Martes,Miércoles,Jueves,Viernes,Sábado,Domingo Lista num. 3: ene,feb,mar,abr,may,jun,jul,ago,sep,oct,nov,dic Lista num. 4: enero,febrero,marzo,abril,mayo,junio,julio,agosto,septiembre,octubre,noviembre,diciembre Lista num. 5: uno,dos,tres,cuatro Lista num. 6: Uno,Dos,Tres Lista num. 7: Elemento1,Elemento2,Elemento3</pre>



En este ejemplo, se borra la lista número 6: {Uno, Dos, Tres}

## 2.1.2.17 DISPLAYXMLSOURCEPANE

### 2.1.2.17.1 Descripción

Permite mostrar la asignación XML especificada por el argumento XmlMap.

### 2.1.2.17.2 Sintaxis:

DisplayXMLSourcePane(XmlMap)

### 2.1.2.17.3 Argumentos

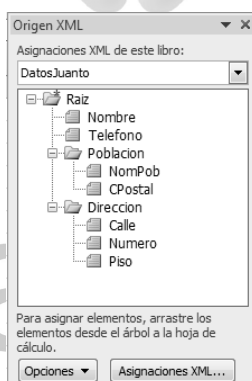
Argumento	Comentario
XmlMap	Asignación XML a mostrar.

### 2.1.2.17.4 Ejemplo

#### Código

```
Sub prueba_DisplayXMLSourcePane()
    Dim objXsd As XmlMap
    Set objXsd = ActiveWorkbook.XmlMaps.Add("c:\temp\Juanto.xsd",
"Raiz")
    objXsd.Name = "DatosJuanto"
    Application.DisplayXMLSourcePane objXsd
End Sub
```

#### Resultado



## 2.1.2.18 DOUBLECLICK

### 2.1.2.18.1 Descripción

Ejecuta un doble clic en la celda que se halle activa.

### 2.1.2.18.2 Sintaxis:

DoubleClick

### 2.1.2.18.3 Argumentos

No tiene.

### 2.1.2.18.4 Ejemplo

Código
<pre>Sub prueba DoubleClick() Application.Worksheets("MiHoja").UsedRange.Range("A5:A5").Select Application.DoubleClick End Sub</pre>

## 2.1.2.19 EVALUATE

### 2.1.2.19.1 Descripción

Permite evaluar un nombre convirtiéndolo en un valor u objeto.

### 2.1.2.19.2 Sintaxis:

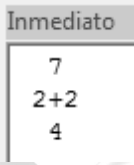
Evaluate(Name)

### 2.1.2.19.3 Argumentos

Argumento	Comentario
Name	Expresión a evaluar.

Ra-Ma®

### 2.1.2.19.4 Ejemplo

Código
<pre> Sub prueba_Evaluate()     'Evalua la expresión     Debug.Print Evaluate("2" &amp; "+" &amp; "5")     'Si la celda B4 contiene el string "2+2" ...     Dim formula As String     formula = Application.Worksheets("MiHoja").Cells(4, 2).Value     Debug.Print formula     Debug.Print Evaluate(formula) End Sub </pre>
Resultado


## 2.1.2.20 EXECUTEEXCEL4MACRO

### 2.1.2.20.1 Descripción

Permite ejecutar una macro de Microsoft Excel 4.0 y obtener su resultado si se produce alguno. El tipo devuelto dependerá de la función.

### 2.1.2.20.2 Sintaxis:

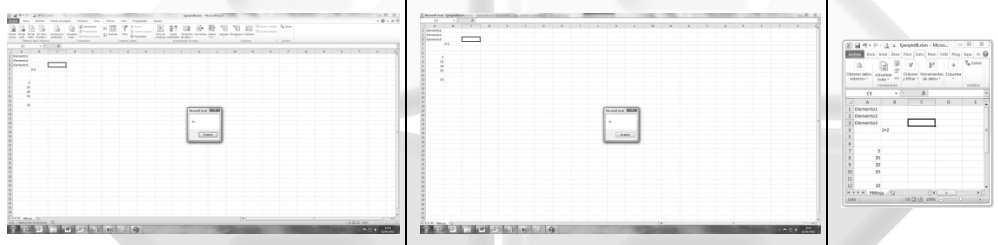
ExecuteExcel4Macro(String)

### 1.1.1.3.2 Argumentos

Argumento	Comentario
String	Acción a ejecutar (obligatorio).

Ra-Ma®

### 2.1.2.20.3 Ejemplo

Código
<pre>Sub prueba_ExecuteExcel4Macro() Application.Worksheets("MiHoja").Activate ExecuteExcel4Macro ("APP.MAXIMIZE()") MsgBox ("Ya") ExecuteExcel4Macro ("FULL.SCREEN()") MsgBox ("Ya") ExecuteExcel4Macro ("APP.RESTORE()") MsgBox ("Ya") End Sub</pre>
Resultado


En este ejemplo maximizamos la ventana, la mostramos a pantalla completo y por último recuperamos su tamaño original.

### 2.1.2.21 FINDFILE

#### 2.1.2.21.1 Descripción

Invoca el cuadro de diálogo Abrir.

#### 2.1.2.21.2 Sintaxis:


FindFile

#### 2.1.2.21.3 Argumentos

No tiene.

Ra-Ma<sup>®</sup>

1.1.1.3.3 Ejemplo

<b>Código</b>
<pre>Sub prueba_FindFile()     Application.FindFile End Sub</pre>
<b>Resultado</b>


2.1.2.22 GETCUSTOMLISTCONTENTS

2.1.2.22.1 Descripción

Devuelve el contenido de la lista personalizada especificada.

2.1.2.22.2 Sintaxis:

GetCustomListContents(ListNum)

2.1.2.22.3 Argumentos

Argumento	Comentario
ListNum	El número de la lista de la que se desean obtener los elementos.

#### 2.1.2.22.4 Ejemplo

Código
<pre>Sub prueba_GetCustomListContents()     Dim lElementos As Variant     Dim numListas As Integer      numListas = Application.CustomListCount      Dim i, n As Integer     Dim texto As String     For n = 1 To numListas         lElementos = Application.GetCustomListContents(n)         texto = ""         For i = LBound(lElementos) To UBound(lElementos)             If texto &lt;&gt; "" Then texto = texto &amp; ","             texto = texto &amp; lElementos(i)         Next i         Debug.Print "Lista num. " &amp; n &amp; ": " &amp; texto     Next n End Sub</pre>
Resultado
<pre>Inmediato Lista num. 1: Lun,Mar,Mié,Jue,Vie,Sáb,Dom Lista num. 2: Lunes,Martes,Miércoles,Jueves,Viernes,Sábado,Domingo Lista num. 3: ene,feb,mar,abr,may,jun,jul,ago,sep,oct,nov,dic Lista num. 4: enero,febrero,marzo,abril,mayo,junio,julio,agosto,septiembre,octubre,noviembre,diciembre Lista num. 5: uno,dos,tres,cuatro Lista num. 6: Elemento1,Elemento2,Elemento3</pre>

En este ejemplo, se obtiene el número de listas personalizadas existentes y se muestran los elementos de cada una de las mismas.

### 2.1.2.23 GETCUSTOMLISTNUM

#### 2.1.2.23.1 Descripción

Devuelve el número de la lista personalizada correspondiente a una matriz de cadenas. Puede usar este método para buscar listas integradas y listas definidas por el usuario.

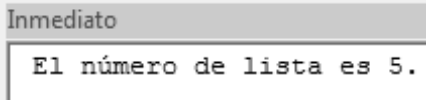
#### 2.1.2.23.2 Sintaxis:

GetCustomListNum(ListArray)

### 2.1.2.23.3 Argumentos

Argumento	Comentario
ListArray	Matriz de cadenas de caracteres.

### 2.1.2.23.4 Ejemplo

Código
<pre>Sub prueba_GetCustomListNum()     Dim numLista As Integer     numLista = Application.GetCustomListNum(Array("uno", "dos", "tres", "cuatro"))     Debug.Print "El número de lista es " &amp; numLista &amp; "." End Sub</pre>
Resultado


## 2.1.2.24 GETOPENFILENAME

### 2.1.2.24.1 Descripción

Permite obtener un nombre de archivo a partir del cuadro de diálogo Abrir.

### 2.1.2.24.2 Sintaxis:

GetOpenFilename(FileFilter, FilterIndex, Title, ButtonText, MultiSelect)

### 2.1.2.24.3 Argumentos

Argumento	Comentario
-----------	------------

Ra-Ma®

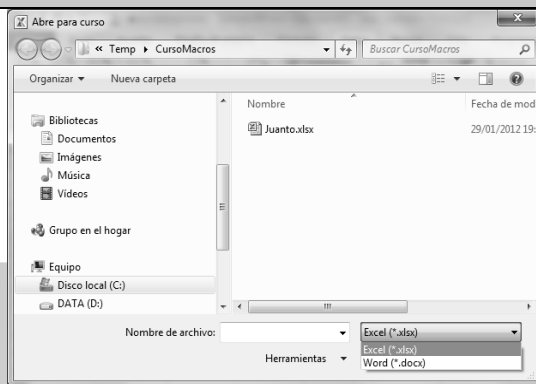
FileFilter	Permite especificar un filtro para que solo se muestren ciertos archivos.  Dicho parámetro se indica mediante pares de cadenas de texto en las que se especifica el nombre o tipo permitido y un patrón que deben cumplir los archivos para ser mostrados.
FilterIndex	Indica el número de filtro a utilizar en primer lugar. Si se omite se utiliza el 1.
Title	Permite definir el título del cuadro.
ButtonText	Sólo Macintosh.
MultiSelect	Si indicamos True, se pueden seleccionar varios archivos a la vez. Por defecto es False.

### 2.1.2.24.4 Ejemplo

#### Código

```
Sub prueba_GetOpenFilename()
    Dim fichero As String
    fichero = Application.GetOpenFilename( _
        "Excel (*.xlsx), *.xlsx, Word (*.docx), *.docx", 1, "Abre para
curso")
End Sub
```

#### Resultado





## 2.1.2.25 GETPHONETIC

### 2.1.2.25.1 Descripción

Devuelve el texto fonético en japonés de la cadena indicada.

### 2.1.2.25.2 Sintaxis:

GetPhonetic(Text)

### 2.1.2.25.3 Argumentos

Argumento	Comentario
Text	Texto a tratar

## 2.1.2.26 GETSAVEASFILENAME

### 2.1.2.26.1 Descripción

Permite obtener el nombre de fichero a utilizar para guardar el documento en curso mostrando al cuadro de diálogo Guardar.

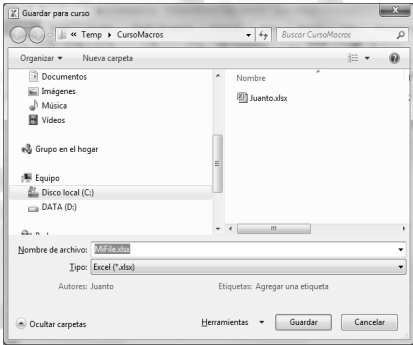
### 2.1.2.26.2 Sintaxis:

GetSaveAsFilename(InitialFilename, FileFilter, FilterIndex, Title, ButtonText)

### 2.1.2.26.3 Argumentos

Argumento	Comentario
InitialFilename	Nombre de fichero sugerido para ser guardado.
FileFilter	Permite especificar un filtro para que solo se muestren ciertos archivos.  Dicho parámetro se indica mediante pares de cadenas de texto en las que se especifica el nombre o tipo permitido y un patrón que deben cumplir los archivos para ser mostrados.
FilterIndex	Indica el número de filtro a utilizar en primer lugar. Si se omite se utiliza el 1.
Title	Permite definir el título del cuadro.
ButtonText	Sólo Macintosh.

### 2.1.2.26.4 Ejemplo

Código
<pre> Sub prueba_GetSaveAsFilename()     Dim fichero As String     fichero = Application.GetSaveAsFilename( _         "MiFile.xlsx", "Excel (*.xlsx), *.xlsx, Word (*.docx), *.docx", 1,         "Guardar para curso") End Sub </pre>
Resultado


### 2.1.2.27 GOTO

#### 2.1.2.27.1 Descripción

Permite seleccionar un rango o procedimiento de cualquier libro activándolo al mismo tiempo.

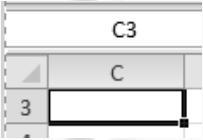
#### 2.1.2.27.2 Sintaxis:

Goto(Referencia, Scroll)

#### 2.1.2.27.3 Argumentos

Argumento	Comentario
Referencia	Celda o procedimiento a activar y sobre el que se desplazará el foco.
Scroll	True coloca la celda indicada en la esquina superior izquierda de la ventana de mostrada por la hoja.

#### 2.1.2.27.4 Ejemplo

Código
<pre>Sub muestra_Goto()     Application.Goto     Reference:=Application.Worksheets("MiHoja").Range("C3"),     Scroll:=True End Sub</pre>
Resultado


#### 2.1.2.28 HELP

##### 2.1.2.28.1 Descripción

Permite invocar una determinada ayuda.

##### 2.1.2.28.2 Sintaxis:

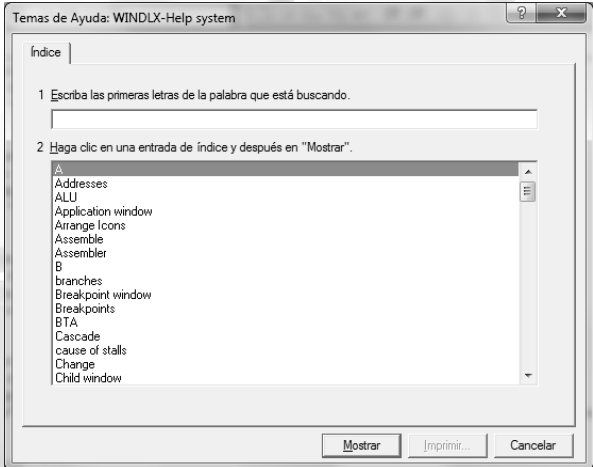
Help(FicheroAyuda, HelpContextID)

##### 2.1.2.28.3 Argumentos

Argumento	Comentario
FicheroAyuda	Fichero que contiene la ayuda.
HelpContextID	Identificador que contiene la ayuda a mostrar. Si no se especifica se muestra el cuadro de temas de ayuda.

Ra-Ma®

#### 2.1.2.28.4 Ejemplo

Código
<pre>Sub prueba_Help()     Application.Help HelpFile:="C:\Temp\CursoMacros\WINDLX.HLP" End Sub</pre>
Resultado


#### 2.1.2.29 INCHESTOPOINTS

##### 2.1.2.29.1 Descripción

Permite convertir pulgadas a puntos.

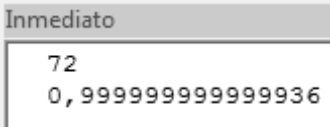
##### 2.1.2.29.2 Sintaxis:

InchesToPoints(Inches)

##### 2.1.2.29.3 Argumentos

Argumento	Comentario
Inches	Valor en pulgadas a convertir.

### 2.1.2.29.4 Ejemplo

Código
<pre>Sub prueba_InchesToPoints()     Debug.Print Application.InchesToPoints(1)     Debug.Print Application.InchesToPoints(0.0138888888888888) End Sub</pre>
Resultado


### 2.1.2.30 INPUTBOX

#### 2.1.2.30.1 Descripción

Permite solicitar información al usuario mediante un cuadro de diálogo.

#### 2.1.2.30.2 Sintaxis:

InputBox(Prompt, Title, Default, Left, Top, HelpFile, HelpContextID, Type)

#### 2.1.2.30.3 Argumentos

Argumento	Comentario
Prompt	Mensaje a mostrar para solicitar la información.
Title	Título del cuadro de diálogo. Por defecto será <b>“Introducir”</b> .
Default	Valor propuesto por defecto como entrada.
Left	Posición X del cuadro en puntos respecto a la esquina superior izquierda de la pantalla.
Top	Posición Y del cuadro en puntos respecto a la esquina superior izquierda de la pantalla.

HelpFile	Nombre del fichero de ayuda.																
HelpContextID	Nódigo del contexto de la ayuda.																
Type	<p>Tipo de datos que ha de tener la entrada. Si no se especifica nada, se devolverá un String.</p> <table border="1"> <thead> <tr> <th>Valor</th><th>Tipo de entrada</th></tr> </thead> <tbody> <tr> <td>0</td><td>Fórmula.</td></tr> <tr> <td>1</td><td>Número.</td></tr> <tr> <td>2</td><td>Texto.</td></tr> <tr> <td>4</td><td>True or false.</td></tr> <tr> <td>8</td><td>Referencia a Celda.</td></tr> <tr> <td>16</td><td>Valor de error.</td></tr> <tr> <td>64</td><td>Matriz de valores.</td></tr> </tbody> </table>	Valor	Tipo de entrada	0	Fórmula.	1	Número.	2	Texto.	4	True or false.	8	Referencia a Celda.	16	Valor de error.	64	Matriz de valores.
Valor	Tipo de entrada																
0	Fórmula.																
1	Número.																
2	Texto.																
4	True or false.																
8	Referencia a Celda.																
16	Valor de error.																
64	Matriz de valores.																

#### 2.1.2.30.4 Ejemplo

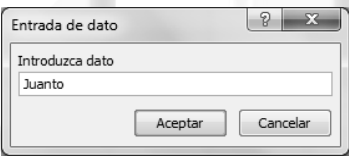
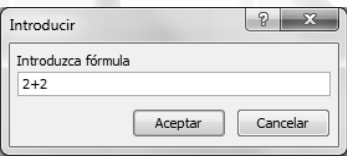
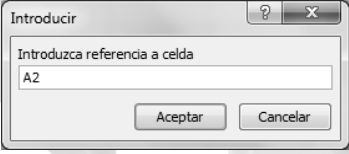
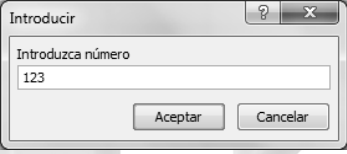
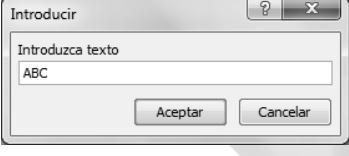
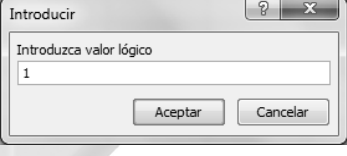
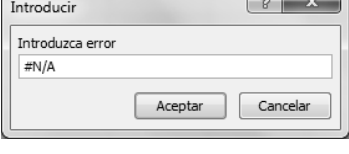
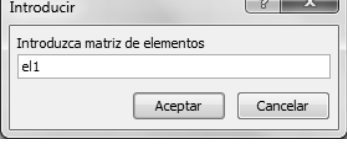
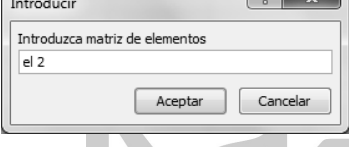
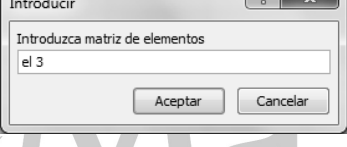
Código
<pre> Sub muestra_InputBox()     Dim formula As String     Dim numero As Double     Dim texto As String     Dim logico As Boolean     Dim ref As String     Dim error As Variant     Dim matriz, res As Variant      texto = Application.InputBox("Introduzca dato", "Entrada de dato", "Juanto")     formula = Application.InputBox("Introduzca fórmula", , "2+2", , , , 0)     numero = Application.InputBox("Introduzca número", , "123", , , , , 1) </pre>

```

    texto = Application.InputBox("Introduzca texto", , "ABC", , ,
, , 2)
    logico = Application.InputBox("Introduzca valor lógico", ,
True, , , , 4)
    ref = Application.InputBox("Introduzca referencia a celda", ,
"A2", , , , 8)
    error = Application.InputBox("Introduzca error", , "#N/A", , ,
, , 16)
    matriz = Array("a", "b", "c")
    res = Application.InputBox("Introduzca matriz de elementos", ,
matriz, , , , 64 + 2)
End Sub

```

### Resultado

### 2.1.2.31 INTERSECT

#### 2.1.2.31.1 Descripción

Permite obtener un rango que contiene la intersección entre 2 ó más rangos.

#### 2.1.2.31.2 Sintaxis:

Intersect(Arg1, Arg2, Arg3, Arg4, Arg5, Arg6, Arg7, Arg8, Arg9, Arg10, Arg11, Arg12, Arg13, Arg14, Arg15, Arg16, Arg17, Arg18, Arg19, Arg20, Arg21, Arg22, Arg23, Arg24, Arg25, Arg26, Arg27, Arg28, Arg29, Arg30)

#### 2.1.2.31.3 Argumentos

Argumento	Comentario
Arg1	Obligatorio. Rango a utilizar para la intersección.
Arg2	Obligatorio. Rango a utilizar para la intersección. Al menos, se han de utilizar 2 rangos para poder realizar la intersección
Arg3 Arg30	Opcional.

#### 2.1.2.31.4 Ejemplo

Código
<pre> Sub prueba_Intersect()     Dim rangoRes As Range     Dim rango1 As Range     Dim rango2 As Range     Set rango1 = Worksheets("MiHoja").Range("C3:D5")     Set rango2 = Worksheets("MiHoja").Range("D4:E5")     Application.Worksheets("MiHoja").Activate     Set rangoRes = Application.Intersect(rango1, rango2)     If Not (rangoRes Is Nothing) Then         rangoRes.Select     End If End Sub </pre>



Resultado			
	C	D	E
	10	40	70
	20	50	80
	30	60	90

### 2.1.2.32 MACROOPTIONS

#### 2.1.2.32.1 Descripción

Permite identificar la función dentro del cuadro de diálogo Opciones de la macro pudiendo indicar en qué categoría debe aparecer y configurando algunos parámetros como son la descripción de la macro, las descripciones de los argumentos, etc...

#### 2.1.2.32.2 Sintaxis:

MacroOptions(Macro, Description, HasMenu, MenuText, HasShortcutKey, ShortcutKey, Category, StatusBar, HelpContextID, HelpFile, ArgumentDescriptions)

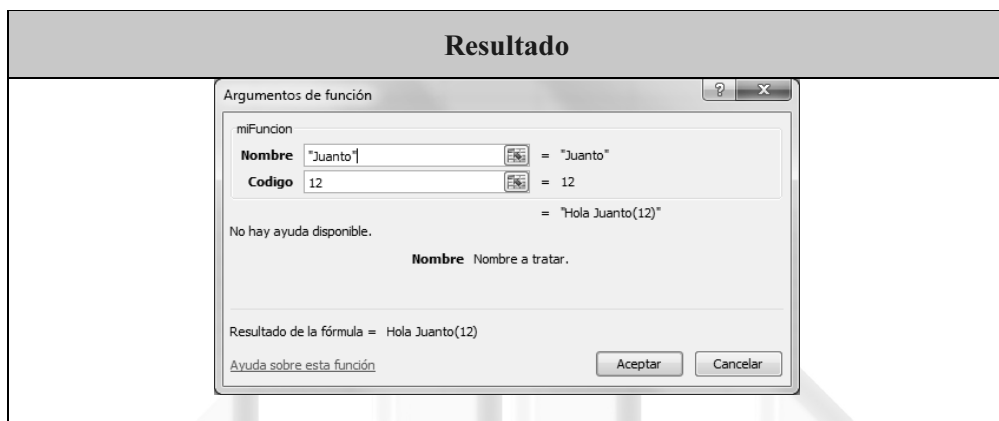
#### 2.1.2.32.3 Argumentos

Argumento	Comentario
Macro	Macro o función de usuario a tratar.
Description	Descripción de la macro o función.
HasMenu	No usado.
MenuText	No usado..
HasShortcutKey	Indica si la función o macro puede invocarse mediante una tecla de método abreviado. Si se define False, no se asigna ninguna tecla.

ShortcutKey	Tecla del método abreviado en caso de que esté definida.
Category	<p>Entero que especifica una categoría de función de macro existente (por ejemplo, Financiera, Fecha y hora o Definida por el usuario). Vea el apartado de comentarios para determinar los enteros asignados a las categorías integradas. Además, puede especificar una cadena para una categoría personalizada. Si proporciona una cadena, se tratará como el nombre de categoría que se muestra en el cuadro de diálogo Insertar función. Si el nombre de categoría no se ha usado nunca, se definirá una nueva categoría con ese nombre. Si usa un nombre de categoría que coincide con el de una categoría integrada (vea la lista en el apartado de comentarios), Microsoft Excel asignará la función definida por el usuario a dicha categoría integrada.</p> <p>Categoría donde insertar la función. Puede utilizarse una existente o bien definir una por el usuario simplemente proporcionando una descripción para tal categoría.</p>
StatusBar	Texto para la barra de estado.
HelpContextID	Id. de contexto de la Ayuda asignado a la macro.
HelpFile	Nombre del archivo de Ayuda.
ArgumentDescriptions	Descripciones a utilizar para cada uno de los argumentos de una función de usuario.

#### 2.1.2.32.4 Ejemplo

Código
<pre>Sub prueba_MacroOptions()     Application.MacroOptions Macro:="miFuncion", _         Category:="Descripción de mi función", _         StatusBar:="Mensaje para barra de estado", _         ArgumentDescriptions:=Array("Nombre a tratar", "Numero a tratar") End Sub</pre>



### 2.1.2.33 MAILLOGOFF

#### 2.1.2.33.1 Descripción

Cierra una sesión de correo MAPI establecida por Microsoft Excel.

Permite cerrar una sesión de correo establecida previamente.

#### 2.1.2.33.2 Sintaxis:

MailLogoff

#### 2.1.2.33.3 Argumentos

No tiene.

#### 2.1.2.33.4 Ejemplo

Código
<pre> Sub prueba_MailLogoff()     If Not IsNull(Application.MailSession) Then         Application.MailLogoff         Debug.Print "Sesión de correo cerrada."     Else         Debug.Print "Sesión de correo ya estaba cerrada anteriormente."     End If End Sub </pre>

Resultado
<div>Inmediato</div> <pre>Sesión de correo cerrada. Sesión de correo ya estaba cerrada anteriormente.</pre>

Esta prueba cierra una sesión de correo previamente establecida y, al intentar volver a cerrarla da un mensaje indicando que ya estaba cerrada.

### 2.1.2.34 MAILLOGON

#### 2.1.2.34.1 Descripción

Permite establecer una sesión de correo de MAPI Mail o Microsoft Exchange.

#### 2.1.2.34.2 Sintaxis:

MailLogon(Name, Password, DownloadNewMail)

#### 2.1.2.34.3 Argumentos

Argumento	Comentario
Name	Nombre de la cuenta de correo. Por defecto se usará el nombre de la cuenta de correo predeterminada.
Password	Contraseña para la cuenta de correo. No es necesario con Microsoft Exchange.
DownloadNewMail	Si True, descarga el correo nuevo inmediatamente.

#### 2.1.2.34.4 Ejemplo

Código
<pre>Sub prueba_MailLogon()     If IsNull(Application.MailSession) Then         Application.MailLogon         Debug.Print "Sesión de correo establecida"     Else         Debug.Print "Sesión de correo ya estaba establecida anteriormente."     End If End Sub</pre>

Resultado	
Inmediato	<pre>Sesión de correo establecida Sesión de correo ya estaba establecida anteriormente.</pre>

Esta prueba abre una sesión de correo y, al intentar volver a abrirla da un mensaje indicando que ya estaba abierta.

### 2.1.2.35 ONKEY

#### 2.1.2.35.1 Descripción

Ejecuta un procedimiento especificado cuando se presiona una tecla o una combinación de teclas determinada.

Permite ejecutar un procedimiento al pulsar una tecla o combinación de teclas.

#### 2.1.2.35.2 Sintaxis:

OnKey(Key, Procedure)

#### 2.1.2.35.3 Argumentos

Argumento	Comentario
Key	Combinación de teclas a definir.
Procedure	Procedimiento a ejecutar cuando se pulse la combinación de teclas definida en Key. Si se deja en blanco, se restaura el funcionamiento por defecto que pudiera tener la combinación por si misma.

Ra-Ma®

Códigos a utilizar para indicar la tecla que se quiere definir:

Código	Tecla
{PGDN}	AV PÁG
{HELP}	AYUDA
{CAPSLOCK}	BLOQ MAYÚS
{SCROLLLOCK}	BLOQUEO DE DESPLAZAMIENTO
{NUMLOCK}	BLOQUEO NUMÉRICO
{CLEAR}	BORRAR
~ (tilde)	ENTRAR
{ESCAPE} o {ESC}	ESC
{F1} a {F15}	F1 a F15
{END}	FIN
{DOWN}	FLECHA ABAJO
{UP}	FLECHA ARRIBA
{RIGHT}	FLECHA DERECHA
{LEFT}	FLECHA IZQUIERDA
{HOME}	INICIO
{INSERT}	INSERTAR
{BREAK}	INTERRUMPIR
{ENTER}	INTRO (teclado numérico)
{PGUP}	RE PÁG
{RETURN}	RETORNO
{BACKSPACE} o {BS}	RETROCESO
{DELETE} o {DEL}	SUPRIMIR o SUPR
{TAB}	TAB

La definición puede basarse en una combinación de alguna de las teclas anteriores más las teclas MAYUS, CTRL y/o ALT.

Para indicar éstas últimas se utilizan los siguientes códigos:

Código	Tecla
+ (signo más)	MAYÚSCULAS
^ (signo exponencial)	CONTROL
% (signo de porcentaje)	ALT

#### 2.1.2.35.4 Ejemplo

Código
<pre> Sub prueba_Activa_OnKey()     Application.OnKey "+^{F1}", "saludar" End Sub  Sub prueba_Desactiva_OnKey()     Application.OnKey "+^{F1}" End Sub  Function saludar()     MsgBox ("Hola Juanto") End Function </pre>

Al ejecutar **prueba\_Activa\_OnKey()**, definimos que al pulsar la combinación de teclas **CTRL + MAYUS + F1** aparezca un mensaje con el texto **“Hola Juanto”** gracias a la invocación de la función **saludar**.

Al ejecutar **prueba\_Desactiva\_OnKey()**, restablecemos la definición original de dicha combinación de teclas y por tanto, al teclearlas no sucederá nada especial.

#### 2.1.2.36 ONREPEAT

##### 2.1.2.36.1 Descripción

Permite indicar el procedimiento a ejecutar y el texto mediante el cual se identificará el mismo si se elige la opción Repetir.

### 2.1.2.36.2 Sintaxis:

OnRepeat(Text, Procedure)

### 2.1.2.36.3 Argumentos

Argumento	Comentario
Text	Texto a mostrar con el comando Repetir del menú Edición.
Procedure	Procedimiento que se ejecutará.

### 2.1.2.36.4 Ejemplo

Código
<pre>Sub prueba_OnRepeat()     Application.OnRepeat "Mensaje Repetir", "msgRepetir" End Sub  Function msgRepetir()     MsgBox ("Repito") End Function</pre>
Resultado


## 2.1.2.37 ONTIME

### 2.1.2.37.1 Descripción

Programa la ejecución de un procedimiento a una hora especificada, ya sea a una hora específica o después del transcurso de un período de tiempo específico.

Permite programar la ejecución de un procedimiento a una hora determinada o bien, desde ahora en un tiempo concreto.



### 2.1.2.37.2 Sintaxis:

OnTime(EarliestTime, Procedure, LatestTime, Schedule)

### 2.1.2.37.3 Argumentos

Argumento	Comentario
EarliestTime	Hora a la que se ejecutará el procedimiento indicado.
Procedure	Nombre del procedimiento a ejecutar.
LatestTime	Hora a partir de la cual el procedimiento ya no se ejecutará.
Schedule	True indica que se desea programar un nuevo procedimiento.  False permite anular una programación previa. Por defecto, el valor es True.

### 2.1.2.37.4 Ejemplo

Código
<pre>Sub prueba_OnTime()     Application.OnTime Now + TimeValue("00:00:05"), "saludar"     Application.OnTime TimeValue("22:27:30"), "msgRepetir" End Sub</pre>

Tras la ejecución del procedimiento **prueba\_OnTime**, recibiremos un mensaje a los 5 segundos gracias a la ejecución del método **saludar**. De la misma manera, a las 22:27:30 recibiremos otro mensaje como consecuencia de la ejecución del procedimiento **msgRepetir**.

## 2.1.2.38 ONUNDO

### 2.1.2.38.1 Descripción

Permite indicar el procedimiento a ejecutar y el texto mediante el cual se identificará el mismo si se elige la opción **Deshacer**.


### 2.1.2.38.2 Sintaxis:

OnUndo(Text, Procedure)

### 2.1.2.38.3 Argumentos

Argumento	Comentario
Text	Texto a mostrar con el comando <b>Deshacer</b> del menú Edición.
Procedure	Procedimiento que se ejecutará al ejecutar <b>Deshacer</b> .

### 2.1.2.38.4 Ejemplo

Código
<pre>Function msgDeshacer()     MsgBox ("Deshago") End Function  Sub prueba_OnUndo()     Application.OnUndo "Mensaje Deshacer", "msgDeshacer" End Sub</pre>
Resultado


## 2.1.2.39 QUIT

### 2.1.2.39.1 Descripción

Provoca la salida de MS Excel.

### 2.1.2.39.2 Sintaxis:

Quit

### 2.1.2.39.3 Argumentos

No tiene.

#### 2.1.2.39.4 Ejemplo

Código
<pre>Sub prueba_Quit()     Application.Quit End Sub</pre>

Este ejemplo, cierra la Excel MS.

#### 2.1.2.40 RECORDMACRO

##### 2.1.2.40.1 Descripción

Permite insertar código durante la grabación de una macro de forma que dicho código se ejecutará como si se hubiese generado durante dicha grabación.

##### 2.1.2.40.2 Sintaxis:

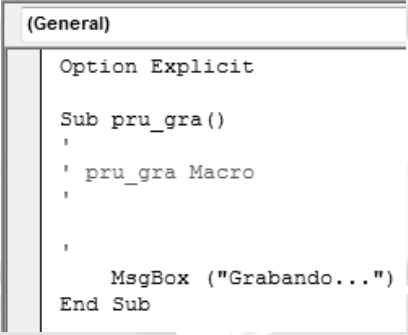
RecordMacro(BasicCode, XlmCode)

##### 2.1.2.40.3 Argumentos

Argumento	Comentario
BasicCode	<p>Cadena que especifica el código de Visual Basic que se grabará si la grabadora de macros está grabando en un módulo de Visual Basic. La cadena se grabará en una sola línea. Si la cadena contiene un retorno de carro (carácter ASCII 10 o Chr\$(10) en código), se grabará en varias líneas.</p> <p>Código de Visual Basic a grabar en la macro. Es posible insertar varias líneas separando cada una de ellas con el carácter obtenido mediante la función Chr\$(10).</p>
XlmCode	No usado.

Ra-Ma®

2.1.2.40.4 Ejemplo

Código
<pre>Sub prueba_RecordMacro()     Application.RecordMacro BasicCode:="MsgBox (""Grabando..."") " End Sub</pre>
Resultado


Este ejemplo, deja la macro ‘**pru\_gra**’ con el código indicado en el parámetro **BasicCode**.

2.1.2.41 REGISTERXLL

2.1.2.41.1 Descripción

Carga un recurso de código XLL y registra automáticamente las funciones y comandos que contiene.

Permite cargar y registrar automáticamente las funciones y comandos contenidas en un recurso de código XLL.

2.1.2.41.2 Sintaxis:

RegisterXLL(Filename)

2.1.2.41.3 Argumentos

Argumento	Comentario
Filename	Nombre del XLL que se desea cargar.

#### 2.1.2.41.4 Ejemplo

Código
<pre>Sub prueba_RegisterXLL()     Dim objXll As Object     Set objXll = CreateObject("Excel.Application")     objXll.Workbooks.Open _      ("C:\Users\Juanto\AppData\Roaming\Microsoft\Complementos\MiComplemen to.xlam")      objXll.Workbooks("MiComplemento.xlam").RunAutoMacros 2     Set objXll = Nothing End Sub</pre>

#### 2.1.2.42 REPEAT

##### 2.1.2.42.1 Descripción

Repite la última acción efectuada en la interfaz del usuario.

Permite repetir la última acción realizada por parte del usuario.

##### 2.1.2.42.2 Sintaxis:

Repeat

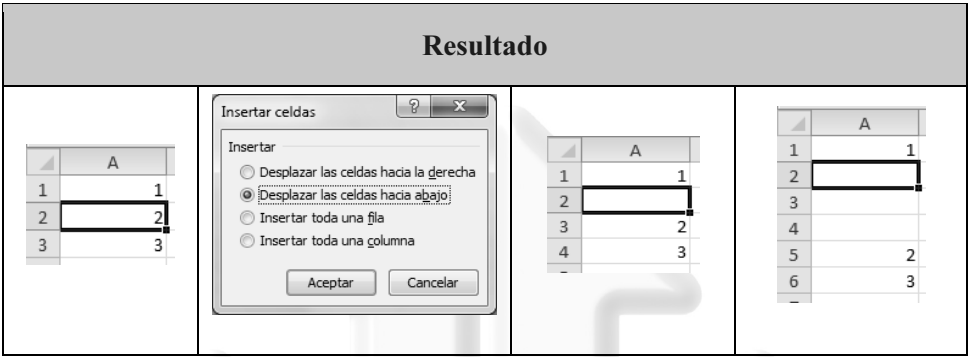
##### 2.1.2.42.3 Argumentos

No tiene.

##### 2.1.2.42.4 Ejemplo

Código
<pre>Sub prueba_Repeat()     Application.Repeat End Sub</pre>

Ra-Ma®



En este ejemplo, puede observarse que, tras realizar un desplazamiento de celdas hacia abajo, el usuario ha ejecutado 2 veces el procedimiento `prueba_Repeat()`

2.1.2.43 RUN

2.1.2.43.1 Descripción

Permite ejecutar un procedimiento o llamar a una función pasando parámetros. Es posible obtener el resultado de la ejecución del procedimiento.



2.1.2.43.2 Sintaxis:

`Run(Macro, Arg1, Arg2, Arg3, Arg4, Arg5, Arg6, Arg7, Arg8, Arg9, Arg10, Arg11, Arg12, Arg13, Arg14, Arg15, Arg16, Arg17, Arg18, Arg19, Arg20, Arg21, Arg22, Arg23, Arg24, Arg25, Arg26, Arg27, Arg28, Arg29, Arg30)`

1.1.1.3.4 Argumentos

Argumento	Comentario
Macro	Procedimiento o función a ejecutar.
Arg1 Arg30	Posibles argumentos a pasar a la función o procedimiento.

### 2.1.2.43.3 Ejemplo

Código	
<pre> Sub prueba_Run()     ' Llamada a función con paso de parámetros     MsgBox (Application.Run("miFuncion", "Juanto", 123))     ' Ejecución de un procedimiento     Application.Run ("saludar") End Sub  Function miFuncion(ByVal nombre As String, ByVal codigo As Integer)     miFuncion = "Hola " &amp; nombre &amp; "(" &amp; codigo &amp; ")" End Function </pre>	
Resultado	
	

Este ejemplo ejecuta la llamada a la función **miFunción** pasándole 2 parámetros y a continuación ejecuta el procedimiento **saludar**.

### 2.1.2.44 SAVEWORKSPACE

#### 2.1.2.44.1 Descripción

Permite guardar el área de trabajo actual con un nombre determinado (al que le añade la extensión xlw ó xlsx según la versión de MS Excel que se esté utilizando) y equivale a pulsar sobre la opción siguiente:



#### 2.1.2.44.2 Sintaxis:

SaveWorkspace(Filename)

### 2.1.2.44.3 Argumentos

Argumento	Comentario
Filename	Archivo a utilizar para guardar el área de trabajo.

### 2.1.2.44.4 Ejemplo

Código
<pre>Sub prueba_SaveWorkspace()     Application.SaveWorkspace "WA_Juanto" End Sub</pre>

Este ejemplo crea un archivo denominado **WA\_Juanto.xlsx** (si la versión usada es Excel 2007 ó Excel 2010).

### 2.1.2.45 SENDKEYS

#### 2.1.2.45.1 Descripción

Envía pulsaciones de tecla a la aplicación activa.

Permite enviar pulsaciones de teclas a una aplicación simulando que el usuario las pulsa sobre la misma.

#### 2.1.2.45.2 Sintaxis:

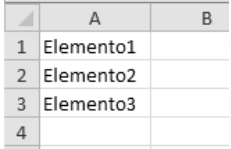
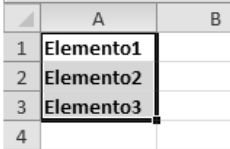
SendKeys(Keys, Wait)

#### 2.1.2.45.3 Argumentos

Argumento	Comentario
Keys	Pulsaciones a enviar.
Wait	True provoca la espera al procesamiento de las teclas enviadas antes de continuar con la ejecución del procedimiento.



### 2.1.2.45.4 Ejemplo

Código		
<pre>Sub prueba_SendKeys()     Application.Worksheets("MiHoja").Activate      Application.Worksheets("MiHoja").UsedRange.Range("A1:A3").Select     Application.SendKeys "^N" End Sub</pre>		
Resultado		
		

Tras la ejecución de este procedimiento, tenemos que se ha aplicado el formato **Negrita** al rango **A1:A3**.

### 2.1.2.46 SHAREPOINTVERSION

#### 2.1.2.46.1 Descripción

Permite obtener el número de versión de Microsoft SharePoint Foundation que se ejecuta en una determinada URL.

#### 2.1.2.46.2 Sintaxis:

SharePointVersion(bstrUrl)

#### 2.1.2.46.3 Argumentos

Argumento	Comentario
bstrUrl	URL a tratar.

### 2.1.2.46.4 Ejemplo

Código
<pre>Sub prueba_SharePointVersion()     Debug.Print     Application.SharePointVersion("http://www.gencat.cat") End Sub</pre>

## 2.1.2.47 UNDO

### 2.1.2.47.1 Descripción

Permite deshacer la última acción realizada por parte del usuario.

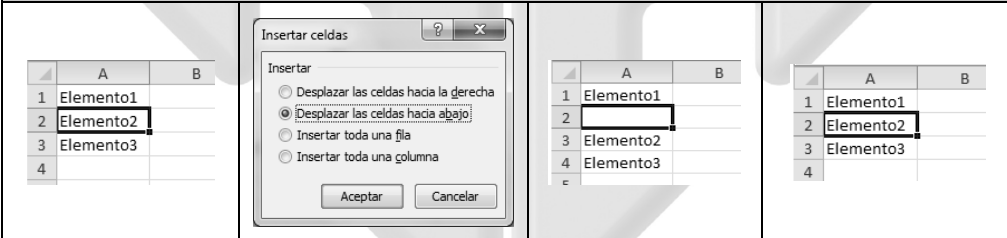
### 2.1.2.47.2 Sintaxis:

Undo

### 2.1.2.47.3 Argumentos

No tiene.

### 2.1.2.47.4 Ejemplo

Código	
<pre>Sub prueba_Undo()     Application.Undo End Sub</pre>	
Resultado	
 <p>The diagram illustrates the 'Insertar celdas' (Insert Cells) dialog box and the resulting spreadsheet state. The dialog box has four options: 'Desplazar las celdas hacia la derecha', 'Desplazar las celdas hacia abajo' (selected), 'Insertar toda una fila', and 'Insertar toda una columna'. The 'Aceptar' (Accept) button is highlighted. To the left, a spreadsheet shows a table with columns A and B, and rows 1 to 4. Row 1 contains 'Elemento1', row 2 contains 'Elemento2', row 3 contains 'Elemento3', and row 4 is empty. To the right, the same spreadsheet is shown after the 'Insertar celdas' operation, where the cells have been shifted down, and the new row 4 contains 'Elemento3'.</p>	

Este ejemplo, muestra como tras insertar una celda desplazando hacia abajo el resto de celdas, la ejecución del procedimiento prueba\_Undo vuelve a dejar las celdas en su posición original deshaciendo la última acción realizada.

## 2.1.2.48 UNION

### 2.1.2.48.1 Descripción

Devuelve la unión de dos o más rangos.

Permite unir dos rangos o más.

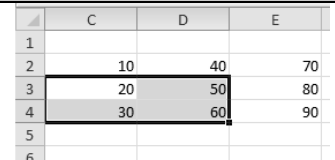
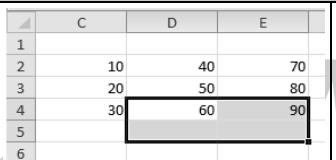
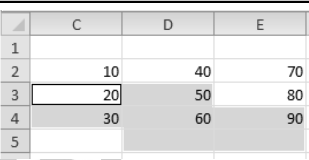
### 2.1.2.48.2 Sintaxis:

Union(Arg1, Arg2, Arg3, Arg4, Arg5, Arg6, Arg7, Arg8, Arg9, Arg10, Arg11, Arg12, Arg13, Arg14, Arg15, Arg16, Arg17, Arg18, Arg19, Arg20, Arg21, Arg22, Arg23, Arg24, Arg25, Arg26, Arg27, Arg28, Arg29, Arg30)

### 2.1.2.48.3 Argumentos

Argumento	Comentario
Arg1	Obligatorio. Rango a unir.
Arg2	Obligatorio. Rango a unir con el anterior. Al menos, se han de utilizar 2 rangos para poder realizar la unión.
Arg3 Arg30	- Opcional.

### 2.1.2.48.4 Ejemplo

Código			
<pre> Sub prueba_Union()     Dim rangoRes As Range     Dim rango1 As Range     Dim rango2 As Range     Set rango1 = Worksheets("MiHoja").Range("C3:D4")     Set rango2 = Worksheets("MiHoja").Range("D4:E5")     Application.Worksheets("MiHoja").Activate     Set rangoRes = Application.Union(rango1, rango2)     If Not (rangoRes Is Nothing) Then         rangoRes.Select     End If End Sub </pre>			
Resultado			
			

Este ejemplo muestra el resultado de la unión de los rangos **C3:D4** y **D4:E5**.

## 2.1.2.49 VOLATILE

### 2.1.2.49.1 Descripción

Permite indicar que una función de usuario determinada es volátil. Indicar que una función es volátil significa que cualquier cambio realizado en cualquier celda forzará una actualización de las celdas que utilicen esta función.

### 2.1.2.49.2 Sintaxis:

Volatile(Volatile)

### 2.1.2.49.3 Argumentos

Argumento	Comentario
Volatile	True indica que la función es volátil y es el valor por defecto.

### 2.1.2.49.4 Ejemplo

Código
<pre>Function miHora()     Application.Volatile (True)     miHora = Now End Function</pre>

En este ejemplo, definimos la función para que, cada vez que se produzca algún cambio en alguna celda, las celdas que utilicen esta función se recalculen. Por ejemplo, podríamos suponer que en una celda he utilizado la función miHora para almacenar la fecha del último cambio.

## 2.1.2.50 WAIT

### 2.1.2.50.1 Descripción

Permite realizar una pausa hasta un determinado momento.

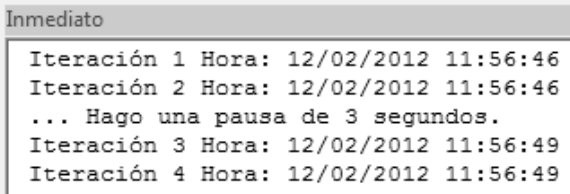
### 2.1.2.50.2 Sintaxis:

Wait(Time)

### 2.1.2.50.3 Argumentos

Argumento	Comentario
Time	Hora hasta la que el procedimiento se detendrá.

### 2.1.2.50.4 Ejemplo

Código
<pre> Sub prueba_Wait()     Dim n As Integer     For n = 1 To 4         Debug.Print ("Iteración " &amp; n &amp; " Hora: " &amp; Now)         If n = 2 Then             Debug.Print ("... Hago una pausa de 3 segundos.")             Application.Wait (Now + TimeValue("0:00:03"))         End If     Next n End Sub </pre>
Resultado
 <pre> Immediate Iteración 1 Hora: 12/02/2012 11:56:46 Iteración 2 Hora: 12/02/2012 11:56:46 ... Hago una pausa de 3 segundos. Iteración 3 Hora: 12/02/2012 11:56:49 Iteración 4 Hora: 12/02/2012 11:56:49 </pre>

En este ejemplo se ha producido una pausa de 3 segundos tras la segunda iteración.

## 2.1.3 Eventos

### 2.1.3.1 AFTERCALCULATE

Evento que se dispara cuando se han realizado todas las actividades pendientes y de cálculo.

### 2.1.3.2 NEWWORKBOOK

Evento que se dispara cuando se crea un nuevo libro.

### **2.1.3.3 PROTECTEDVIEWWINDOWACTIVATE**

Evento que se dispara cuando se activa una ventana Vista protegida.

### **2.1.3.4 PROTECTEDVIEWWINDOWBEFORECLOSE**

Evento que se dispara cuando antes de que se cierre una o libro de una ventana Vista protegida.

### **2.1.3.5 PROTECTEDVIEWWINDOWBEFOREEDIT**

Evento que se dispara antes de habilitar la edición en el libro de la ventana que tiene la Vista protegida especificada.

### **2.1.3.6 PROTECTEDVIEWWINDOWDEACTIVATE**

Evento que se dispara cuando se desactiva una ventana Vista protegida.

### **2.1.3.7 PROTECTEDVIEWWINDOWOPEN**

Evento que se dispara cuando se abre un libro en una ventana de tipo Vista protegida.

### **2.1.3.8 PROTECTEDVIEWWINDOWRESIZE**

Evento que se dispara al cambiar el tamaño de una ventana Vista protegida.

### **2.1.3.9 SHEETACTIVATE**

Evento que se dispara cuando al activar una hoja.

### **2.1.3.10 SHEETBEFOREDOUBLECLICK**

Evento que se dispara cuando se hace un doble clic en una hoja de cálculo y se produce antes de la acción de doble clic determinada.

### **2.1.3.11 SHEETBEFORERIGHTCLICK**

Evento que se dispara al hacer clic con el botón secundario del mouse en una hoja de cálculo y se produce antes de la acción determinada.

### **2.1.3.12 SHEETCALCULATE**

Evento que se dispara después de actualizar una hoja de cálculo.

### **2.1.3.13 SHEETCHANGE**

Evento que se dispara al modificar una celda de cualquier hoja.

**2.1.3.14 SHEETDEACTIVATE**

Evento que se dispara cuando se desactiva una hoja.

**2.1.3.15 SHEETFOLLOWHYPERLINK**

Evento que se dispara cuando se hace clic en un hipervínculo de MS Excel.

**2.1.3.16 SHEETPIVOTTABLEAFTERVALUECHANGE**

Evento que se dispara cuando se calcula de nuevo una celda dentro de una tabla dinámica.

**2.1.3.17 SHEETPIVOTTABLEBEFOREALLOCATECHANGES**

Evento que se dispara cuando se aplican los cambios a una tabla dinámica.

**2.1.3.18 SHEETPIVOTTABLEBEFORECOMMITCHANGES**

Evento que se dispara antes de confirmar los cambios en el origen de datos OLAP de una tabla dinámica.

**2.1.3.19 SHEETPIVOTTABLEBEFOREDISCARDCHANGES**

Evento que se dispara antes de descartar los cambios realizados en una tabla dinámica.

**2.1.3.20 SHEETPIVOTTABLEUPDATE**

Evento que se dispara después de actualizar una hoja del informe de tabla dinámica.

**2.1.3.21 SHEETSELECTIONCHANGE**

Evento que se dispara cuando cuando se cambia la selección en una hoja.

**2.1.3.22 WINDOWACTIVATE**

Evento que se dispara al activar una ventana del libro.

**2.1.3.23 WINDOWDEACTIVATE**

Evento que se dispara al desactivar una ventana del libro.

**2.1.3.24 WINDOWRESIZE**

Evento que se dispara al cambiar el tamaño de cualquier ventana del libro.

### **2.1.3.25 WORKBOOKACTIVATE**

Evento que se dispara al activar un libro.

### **2.1.3.26 WORKBOOKADDININSTALL**

Evento que se dispara al instalar un libro como complemento.

### **2.1.3.27 WORKBOOKADDINUNINSTALL**

Evento que se dispara al desinstalar un libro que ese haya instalado como complemento.

### **2.1.3.28 WORKBOOKAFTERSAVE**

Evento que se dispara después de guardar el libro.

### **2.1.3.29 WORKBOOKAFTERXMLEXPORT**

Evento que se dispara después de que MS Excel guarde o exporte datos XML desde el libro especificado.

### **2.1.3.30 WORKBOOKAFTERXMLIMPORT**

Evento que se dispara tras actualizar una conexión de datos XML, o al importar datos XML a cualquier libro de Microsoft Excel abierto.

### **2.1.3.31 WORKBOOKBEFORECLOSE**

Evento que se dispara antes de que se cierre un libro.

### **2.1.3.32 WORKBOOKBEFOREPRINT**

Evento que se dispara antes de imprimir un libro.

### **2.1.3.33 WORKBOOKBEFORESAVE**

Evento que se dispara antes de guardar un libro.

### **2.1.3.34 WORKBOOKBEFOREXMLEXPORT**

Evento que se dispara antes de que MS Excel guarde o exporte datos XML desde el libro especificado.

### **2.1.3.35 WORKBOOKBEFOREXMLIMPORT**

Evento que se dispara antes de actualizar una conexión de datos XML o antes de importar datos XML a un libro abierto.



### **2.1.3.36 WORKBOOKDEACTIVATE**

Evento que se dispara antes de desactivar cualquier libro abierto.

### **2.1.3.37 WORKBOOKNEWCHART**

Evento que se dispara se crea un nuevo gráfico en cualquier libro.

### **2.1.3.38 WORKBOOKNEWSHEET**

Evento que se dispara al crear una nueva hoja en un libro.

### **2.1.3.39 WORKBOOKOPEN**

Evento que se dispara cuando se abre un libro.

### **2.1.3.40 WORKBOOKPIVOTTABLECLOSECONNECTION**

Evento que se dispara al cerrar la conexión de un informe de tabla dinámica.

### **2.1.3.41 WORKBOOKPIVOTTABLEOPENCONNECTION**

Evento que se dispara al abrir la conexión de un informe de tabla dinámica.

### **2.1.3.42 WORKBOOKROWSETCOMPLETE**

Evento que se dispara cuando al recorrer el conjunto de registros o invocar la acción de conjunto de filas en una tabla dinámica OLAP.

### **2.1.3.43 WORKBOOKSYNC**

Evento que se dispara cuando se sincroniza la copia local de un libro que forma parte de un área de documentos con la copia almacenada en el servidor.

## **2.2 EL OBJETO LIBRO (WORKBOOK).**

Objeto que representa un libro en la aplicación MS Excel

### **2.2.1 Propiedades**

#### **2.2.1.1 ACCURACYVERSION**

Indica si ciertas funciones de la hoja utilizan algoritmos de precisión mas recientes.

### **2.2.1.2 ACTIVECHART**

Proporciona objeto representante del gráfico activo.

### **2.2.1.3 ACTIVESHEET**

Proporciona objeto representante de la hoja activa.

### **2.2.1.4 ACTIVELICER**

Proporciona objeto representante de la segmentación de datos activa en el libro activo o especificado.

### **2.2.1.5 APPLICATION**

Proporciona objeto representante de la aplicación Microsoft Excel.

### **2.2.1.6 AUTOUPDATEFREQUENCY**

Define o consulta los minutos transcurridos entre actualizaciones automáticas del libro compartido.

### **2.2.1.7 AUTOUPDATESAVECHANGES**

Cada vez que el libro se actualiza automáticamente la propiedad vale **True** si los cambios actuales del libro compartido se envían a otros usuarios. Por defecto, vale **True**.

### **2.2.1.8 BUILTINDOCUMENTPROPERTIES**

Proporciona una colección de tipo DocumentProperties que representa todas las propiedades de documento integradas en el libro especificado.

### **2.2.1.9 CALCULATIONVERSION**

Proporciona información de la versión de Excel correspondiente a la última vez que se actualizó.

### **2.2.1.10 CHANGEHISTORYDURATION**

Define o consulta el número de días correspondiente al historial de cambios del libro compartido.

### **2.2.1.11 CHARTS**

Proporciona una colección con las hojas de gráfico del libro especificado.

### **2.2.1.12 CHECKCOMPATIBILITY**

Define o consulta si al guardar el libro se ejecuta o no el comprobador de compatibilidad.

### **2.2.1.13 CODENAME**

Proporciona el nombre de código del objeto.

### **2.2.1.14 COLORS**

Define o consulta los colores de la paleta del libro los cuales son representados por valores RGB.

### **2.2.1.15 COMMANDBARS**

Proporciona objeto con las barras de comandos de MS Excel.

### **2.2.1.16 CONFLICTRESOLUTION**

Define o consulta la forma en que se han de resolver los conflictos producidos al actualiza un libro compartido.

### **2.2.1.17 CONNECTIONS**

Propiedad que establece una conexión entre el libro y un origen de datos ODBC u OLEDB.

### **2.2.1.18 CONNECTIONSDISABLED**

Indica si las conexiones externas o los vínculos del libro están deshabilitadas.

### **2.2.1.19 CONTAINER**

Proporciona objeto que representa la aplicación contenedora del objeto OLE indicado.

### **2.2.1.20 CONTENTTYPEPROPERTIES**

Proporciona una descripción de los metadatos almacenados en el libro.

### **2.2.1.21 CREATEBACKUP**

Devuelve True si al guardar el archivo se crea un archivo de copia de seguridad.

### **2.2.1.22 CREATOR**

Proporciona valor que indica que aplicación creó el objeto.

### **2.2.1.23 CUSTOMDOCUMENTPROPERTIES**

Define o consulta las propiedades de documento personalizadas del libro indicado mediante una colección de tipo DocumentProperties.

### **2.2.1.24 CUSTOMVIEWS**

Proporciona una colección de vistas personalizadas del libro mediante una colección de tipo CustomViews.

### **2.2.1.25 CUSTOMXMLPARTS**

Proporciona el código XML personalizado en el almacén de datos XML mediante una colección CustomXMLParts.

### **2.2.1.26 DATE1904**

Define o consulta si el libro usa el sistema de fechas 1904.

### **2.2.1.27 DEFAULTPIVOTTABLESTYLE**

Define o consulta el estilo de tabla usado como estilo predeterminado para las tablas dinámicas.

### **2.2.1.28 DEFAULTSLICERSTYLE**

Define o consulta el estilo predeterminado para las segmentaciones de datos.

### **2.2.1.29 DEFAULTTABLESTYLE**

Define o consulta el estilo de tabla usado como estilo de tabla predeterminado.

### **2.2.1.30 DISPLAYDRAWINGOBJECTS**

Define o consulta la forma en cómo se muestran las formas.

### **2.2.1.31 DISPLAYINKCOMMENTS**

Define o consulta si se muestran comentarios manuscritos en el libro.

### **2.2.1.32 DOCUMENTINSPECTORS**

Proporciona una colección que representa los módulos de inspector de documento del libro indicado.

### **2.2.1.33 DOCUMENTLIBRARYVERSIONS**

Proporciona una colección que representa la colección de versiones de un libro compartido.

### **2.2.1.34 DONOTPROMPTFORCONVERT**

Define o consulta si se solicita confirmación para convertir un libro que contenga características no compatibles con versiones de Excel anteriores a Excel 2007.

### **2.2.1.35 ENABLEAUTORECOVER**

Define o consulta si los archivos modificados se han con cierta frecuencia.

### **2.2.1.36 ENCRYPTIONPROVIDER**

Define o consulta el nombre del proveedor del algoritmo de cifrado utilizado para cifrar los documentos.

### **2.2.1.37 ENVELOPEVISIBLE**

Define o consulta si están visibles el encabezado de la composición y la barra de herramientas de correo electrónico.

### **2.2.1.38 EXCEL4INTLMACROSHEETS**

Proporciona una colección que representa todas las hojas de macros internacionales de MS Excel 4.0 del libro indicado.

### **2.2.1.39 EXCEL4MACROSHEETS**

Proporciona una colección que representa todas las hojas de macros de MS Excel 4.0 del libro indicado.

### **2.2.1.40 EXCEL8COMPATIBILITYMODE**

Permite comprobar si el libro está en modo de compatibilidad.

### **2.2.1.41 FILEFORMAT**

Proporciona el tipo del libro y formato del archivo mediante la constante XlFileFormat.

### **2.2.1.42 FINAL**

Define o consulta si un libro es final.

### **2.2.1.43 FORCEFULLCALCULATION**

Define o consulta si el libro está realizando el cálculo en modo forzado.

### **2.2.1.44 FULLNAME**

Proporciona el nombre completo (incluyendo la ruta) del libro.

### **2.2.1.45 FULLNAMEURLENCODED**

Proporciona el nombre completo (incluyendo la ruta) del libro.

### **2.2.1.46 HASPASSWORD**

Indica si está protegido mediante una contraseña devolviendo True.

### **2.2.1.47 HASVBPROJECT**

Indica si el libro tiene un proyecto VBA devolviendo True.

### **2.2.1.48 HIGHLIGHTCHANGESONSCREEN**

Indica si los cambios realizados en el libro compartido se resaltan en la pantalla devolviendo True.

### **2.2.1.49 ICONSETS**

Permite filtrar los datos de un libro en función de un icono de celda.

### **2.2.1.50 INACTIVELISTBORDERVISIBLE**

Define o consulta si se han de mostrar los bordes de lista aunque no esté activa. Puede devolver True o False según se muestren o no respectivamente.

### **2.2.1.51 ISADDIN**

Define o consulta si el libro se ejecuta como complemento.

### **2.2.1.52 ISINPLACE**

Devuelve True si el libro se está modificando en el sitio o False si se ha abierto en MS Excel.

### **2.2.1.53 KEEPCCHANGEHISTORY**

Define o consulta si ha de estar habilitado el control de cambios en el libro compartido.

### **2.2.1.54 LISTCHANGESONNEWSHEET**

Define o consulta si los cambios realizados en el libro compartido se han de mostrar en una hoja de cálculo distinta.

### **2.2.1.55 MULTIUSEREDITING**

Indica si el libro está abierto como una lista compartida devolviendo True.

### **2.2.1.56 NAME**

Proporciona el nombre del objeto.

### **2.2.1.57 NAMES**

Proporciona una colección de nombres de rangos nombrados.

### **2.2.1.58 PARENT**

Proporciona el nombre del objeto primario del objeto especificado.

### **2.2.1.59 PASSWORD**

Define o consulta la contraseña con la que debe abrirse el libro indicado. Si se establece un string vacío, el libro no tendrá contraseña.

### **2.2.1.60 PASSWORDENCRYPTIONALGORITHM**

Devuelve el valor correspondiente al algoritmo utilizado para cifrar las contraseñas del libro.

### **2.2.1.61 PASSWORDENCRYPTIONFILEPROPERTIES**

Devuelve True si las propiedades del archivo del libro protegido por contraseña están cifradas.

### **2.2.1.62 PASSWORDENCRYPTIONKEYLENGTH**

Indica la longitud de la clave que MS Excel usa para cifrar las contraseñas del libro.

### **2.2.1.63 PASSWORDENCRYPTIONPROVIDER**

Indica el nombre del proveedor del algoritmo utilizado para cifrar las contraseñas del libro.

### **2.2.1.64 PATH**

Devuelve la ruta de acceso completa del libro.

### **2.2.1.65 PERMISSION**

Proporciona la configuración de los permisos del libro mediante un objeto Permission.

### **2.2.1.66 PERSONALVIEWLISTSETTINGS**

Define o consulta si los valores de filtro y ordenación del libro compartido se han de incluir en la vista personal del usuario.

### **2.2.1.67 PERSONALVIEWPRINTSETTINGS**

Define o consulta si los parámetros de impresión del libro compartido se han de incluir en la vista personal del usuario.

### **2.2.1.68 PRECISIONASDISPLAYED**

Define o consulta si los cálculos del libro se han de realizar utilizando la misma precisión que la usada para mostrar los números. Devuelve True en caso afirmativo.

### **2.2.1.69 PROTECTSTRUCTURE**

Devuelve True si el orden de las hojas dentro del libro está protegido.

### **2.2.1.70 PROTECTWINDOWS**

Devuelve True si las ventanas están protegidas.

### **2.2.1.71 PUBLISHOBJECTS**

Proporciona una colección de objetos que representan elementos del libro que se ha guardado en una página Web.



### **2.2.1.72 READONLY**

Indica si el objeto ha sido abierto sólo de lectura devolviendo True.

### **2.2.1.73 READONLYRECOMMENDED**

Indica si el libro se guardó recomendado de sólo lectura. En caso afirmativo devuelve True.

### **2.2.1.74 REMOVEPERSONALINFORMATION**

Define o consulta si es posible eliminar la información personal del libro. True indica que si es posible.

### **2.2.1.75 RESEARCH**

Proporciona un objeto que contiene el servicio de referencia de un libro.

### **2.2.1.76 REVISIONNUMBER**

Permite consultar cuantas veces el libro se ha guardado como lista compartida. Si se ha abierto en modo exclusivo, este valor devuelve 0.

### **2.2.1.77 SAVED**

Devuelve True si no se han realizado cambios en el libro desde la última vez que se guardó.

### **2.2.1.78 SAVELINKVALUES**

Define o consulta si los valores de los vínculos externos son guardados junto con el libro. En caso afirmativo, devuelve True.

### **2.2.1.79 SERVERPOLICY**

Proporciona un objeto que informa sobre la directiva especificada para un libro almacenado en un servidor que ejecuta Office SharePoint Server 2007 o posterior

### **2.2.1.80 SERVERVIEWABLEITEMS**

Propiedad que indica si los programadores pueden interaccionar con la lista de objetos publicados del libro que se muestran en el servidor.

### **2.2.1.81 SHAREDWORKSPACE**

Proporciona objeto que representa el área de trabajo en que se encuentra un determinado documento. El objeto es de tipo SharedWorkspace.

### **2.2.1.82 SHEETS**

Proporciona una colección con todas las hojas del libro. La colección devuelva es de tipo Sheets.

### **2.2.1.83 SHOWCONFLICTHISTORY**

Devuelve True si la hoja Historial de conflictos está visible en el libro como nua lista compartida.

### **2.2.1.84 SHOWPIVOTCHARTACTIVEFIELDS**

Define o consulta si el Panel de filtros del gráfico dinámico ha de estar visible.

### **2.2.1.85 SHOWPIVOTTABLEFIELDLIST**

Define o consulta si es posible mostrar la lista de campos de la tabla dinámica.

### **2.2.1.86 SIGNATURES**

Proporciona las firmas digitales de un libro.

### **2.2.1.87 SLICERCACHES**

Proporciona un objeto de tipo SlicerCaches asociado al libro. Un SlicerCache representa la memoria caché de segmentación de datos

### **2.2.1.88 SMARTDOCUMENT**

Proporciona un objeto de tipo SmartDocument con la configuración de una solución de documento inteligente.

### **2.2.1.89 STYLES**

Devuelve el conjunto de todos los estilos del libro mediante un objeto de tipo Styles.

### 2.2.1.90 SYNC

Proporciona objeto de tipo Sync mediante el cual se puede acceder a los métodos y propiedades de los documentos asociados a un área de trabajo de documento.

### 2.2.1.91 TABLESTYLES

Proporciona un objeto con los estilos usados en el libro actual. Dicho objeto es una colección de Styles y es del tipo TableStyles.

### 2.2.1.92 TEMPLATEREMOVEEXTDATA

Define o consulta si al guardar el libro como una plantilla, las referencias a datos externos se han de eliminar. En caso afirmativo, devuelve True.

### 2.2.1.93 THEME

Permite consultar el tema aplicado al libro en curso.

### 2.2.1.94 UPDATERELINKS

Define o consulta la constante `XlUpdateLink` la cual indica la configuración de un libro para actualizar vínculos OLE integrados. Los vínculos pueden actualizarse de alguna de las siguientes maneras:

**`xlUpdateLinksAlways`** Los vínculos OLE siempre se actualizan.

**`xlUpdateLinksNever`** Los vínculos OLE incrustados no se actualizan nunca.

**`xlUpdateLinksUserSetting`** Los vínculos OLE incrustados se actualizan según la configuración del usuario.

### 2.2.1.95 UPDATEREMOTEREFERENCES

Define o consulta si las referencias remotas han de ser actualizadas en el libro.

### 2.2.1.96 USERSTATUS

Devuelve información sobre los usuarios que poseen el libro abierto como una lista compartida. Dicha información la devuelve en forma de array de 2 dimensiones en el que se almacena por cada usuario la siguiente información:

**(userN,1)** ... Nombre

**(userN,2)** ... Fecha y hora que el usuario abrió el libro.

**(userN,3)** ...Modo de apertura (1: Exclusive, 2: Shared o compartido)

### **2.2.1.97 VBASIGNED**

Indica True si el proyecto VBA se ha firmado digitalmente.

### **2.2.1.98 VBPROJECT**

Proporciona un objeto con el proyecto VBA del libro.

### **2.2.1.99 WEBOPTIONS**

Proporciona una colección de atributos usador por MS Excel al guardar un documento como página Web o al abrir una página Web. La colección devuelta es de tipo WebOptions.

### **2.2.1.100 WINDOWS**

Proporciona un objeto de tipo Windows con todas las ventanas del libro.

### **2.2.1.101 WORKSHEETS**

Proporciona una colección con representa todas las hojas de cálculo del libro. Dicha colección se devuelve mediante un objeto de tipo Sheets.

### **2.2.1.102 WRITEPASSWORD**

Define o consulta la contraseña de escritura del libro indicado.

### **2.2.1.103 WRITERESERVED**

Indica si el libro está protegido contra escritura y, en caso afirmativo, devuelve True.

### **2.2.1.104 WRITERESERVEDBY**

Proporciona el nombre del usuario que tiene permiso de escritura en el libro

### **2.2.1.105 XMLMAPS**

Proporciona una colección con las asignaciones de esquema agregadas al libro. Dicha colección es de tipo XmlMaps.

## 2.2.1.106 XMLNAMESPACES

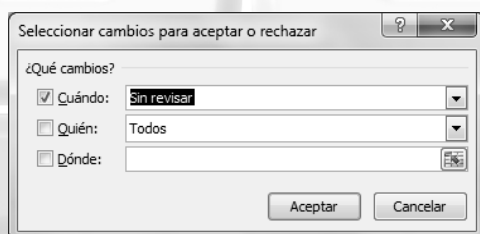
Proporciona una colección de espacios de nombres XML incluidos en el libro. Dicha colección es del tipo XmlNamespaces.

## 2.2.2 Métodos

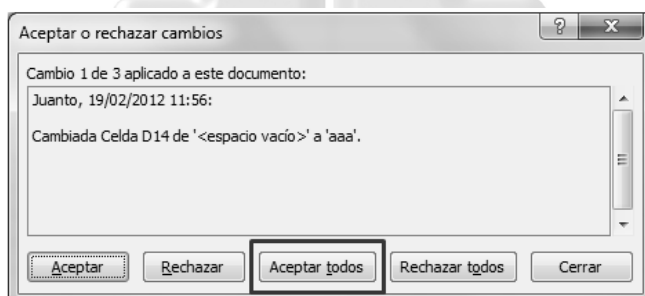
### 2.2.2.1 ACCEPTALLCHANGES

#### 2.2.2.1.1 Descripción

Cuando un libro es compartido, acepta todos los cambios. Equivale a invocar la opción **Control de cambios** situado en la pestaña de **Revisar**:



...y posteriormente aceptar los cambios a través del cuadro de diálogo que aparece en la opción de Aceptar o rechazar cambios:



#### 2.2.2.1.2 Sintaxis:

AcceptAllChanges(When, Who, Where)

Ra-Ma®

### 2.2.2.1.3 Argumentos

Argumento	Comentario
When	Cuando se han de aceptar los cambios.
Who	Quien ha de aceptar los cambios.
Where	Donde se han de aceptar los cambios.

### 2.2.2.1.4 Ejemplo

Código
<pre>Sub prueba_AcceptAllChanges()     Application.Workbooks(1).AcceptAllChanges End Sub</pre>

## 2.2.2.2 ACTIVATE

### 2.2.2.2.1 Descripción

Permite activar la primera ventana del libro indicado.

### 2.2.2.2.2 Sintaxis:

Activate

### 2.2.2.2.3 Argumentos

No tiene.

### 2.2.2.2.4 Ejemplo

Código
<pre>Sub prueba_Activate()     Application.Workbooks(1).Activate End Sub</pre>



Este ejemplo muestra como estando activa la ventana 3, tras la ejecución del procedimiento para a estar activa la ventana 1.

## 2.2.2.3 ADDTOFAVORITES

### 2.2.2.3.1 Descripción

Permite agregar un acceso directo a la carpeta de Favoritos de Internet Explorer.

### 2.2.2.3.2 Sintaxis:


**AddToFavorites**

### 2.2.2.3.3 Argumentos

No tiene.

Ra-Ma<sup>®</sup>

### 2.2.2.3.4 Ejemplo

Código
<pre>Sub prueba_AddToFavorites()     Application.Workbooks(1).AddToFavorites End Sub</pre>
Resultado


### 2.2.2.4 APPLYTHEME

#### 2.2.2.4.1 Descripción

Permite aplicar un tema al libro indicado.

#### 2.2.2.4.2 Sintaxis:

**ApplyTheme (Filename)**

#### 2.2.2.4.3 Argumentos

Argumento	Comentario
Filename	Nombre del fichero que contiene el tema. Normalmente los temas pueden encontrarse en el directorio en el que se halla instalado Office (p.e. "C:\Program Files\Microsoft Office\Document Themes 14")



### 2.2.2.4.4 Ejemplo

Código			
<pre>Sub prueba_ApplyTheme()     Application.Workbooks(1).ApplyTheme ("C:\Program Files\Microsoft Office\Document Themes 14\Austin.thmx") End Sub</pre>			
Resultado			
	A	B	C
1	abcdefghi	ABCDEFGHGI	1234567890
2	b	B	20
3	c	C	30
4	d	D	40
5	e	E	50

En este ejemplo, vemos como se ha cambiado el tema por el nuevo tema “Austin”.

### 2.2.2.5 BREAKLINK

#### 2.2.2.5.1 Descripción

Permite convertir en valores, los resultados de las fórmulas que estén vinculadas a un origen MS Excel u OLE.

#### 2.2.2.5.2 Sintaxis:

BreakLink(Name, Type)

#### 2.2.2.5.3 Argumentos

Argumento	Comentario
<b>Name</b>	Nombre del link a desvincular.
<b>Type</b>	Tipo del link que puede ser alguno de los siguientes: <b>XlLinkTypeExcelLinks:</b> Vínculo a un origen de MS Excel. <b>XlLinkTypeOLELinks:</b> Vínculo a un origen OLE.

### 2.2.2.5.4 Ejemplo

Código													
<pre> Sub prueba_BreakLink()     Dim lLinks As Variant     Dim i As Integer     Dim l As String     lLinks = ActiveWorkbook.LinkSources     For i = LBound(lLinks) To UBound(lLinks)         Debug.Print i &amp; " " &amp; lLinks(i)     Next i     Application.Workbooks(1).BreakLink Name:=lLinks(1),     Type:=xlLinkTypeExcelLinks End Sub </pre>													
Resultado													
<table border="1"> <thead> <tr> <th></th><th>A</th></tr> </thead> <tbody> <tr> <td>1</td><td>10</td></tr> <tr> <td>2</td><td>20</td></tr> </tbody> </table>		A	1	10	2	20	<p>‘Mostrar fórmulas’</p> <table border="1"> <thead> <tr> <th></th><th>A</th></tr> </thead> <tbody> <tr> <td>1</td><td>=C:\Cursos\Curso_Macros\Capitulos\08 - El modelo de objetos\[Juanto.xlsx]Hoja1!\$A\$1</td></tr> <tr> <td>2</td><td>=C:\Cursos\Curso_Macros\Capitulos\08 - El modelo de objetos\[Juanto.xlsx]Hoja1!\$A\$2</td></tr> </tbody> </table>		A	1	=C:\Cursos\Curso_Macros\Capitulos\08 - El modelo de objetos\[Juanto.xlsx]Hoja1!\$A\$1	2	=C:\Cursos\Curso_Macros\Capitulos\08 - El modelo de objetos\[Juanto.xlsx]Hoja1!\$A\$2
	A												
1	10												
2	20												
	A												
1	=C:\Cursos\Curso_Macros\Capitulos\08 - El modelo de objetos\[Juanto.xlsx]Hoja1!\$A\$1												
2	=C:\Cursos\Curso_Macros\Capitulos\08 - El modelo de objetos\[Juanto.xlsx]Hoja1!\$A\$2												
<p>Inmediato</p> <pre>1 C:\Cursos\Curso_Macros\Capitulos\08 - El modelo de objetos\Juanto.xlsx</pre>													
<table border="1"> <thead> <tr> <th></th><th>A</th></tr> </thead> <tbody> <tr> <td>1</td><td>10</td></tr> <tr> <td>2</td><td>20</td></tr> </tbody> </table>		A	1	10	2	20	<p>‘Mostrar fórmulas’</p> <table border="1"> <thead> <tr> <th></th><th>A</th></tr> </thead> <tbody> <tr> <td>1</td><td>10</td></tr> <tr> <td>2</td><td>20</td></tr> </tbody> </table>		A	1	10	2	20
	A												
1	10												
2	20												
	A												
1	10												
2	20												

En este ejemplo, puede apreciarse que las celdas A1 y A2 poseen un vínculo a unas celdas del libro Juanto.xlsx (Hoja1) y que tras ejecutar el procedimiento de ejemplo, dicho vínculo desaparece dejando únicamente los valores asociados al mismo. En la ventana de inmediato pueden observarse los links que tenía la hoja antes de ejecutar el procedimiento.

### 2.2.2.6 CANCHECKIN

#### 2.2.2.6.1 Descripción

Indica si es posible proteger un libro en un servidor.

#### 2.2.2.6.2 Sintaxis:

**CanCheckIn**

#### 2.2.2.6.3 Argumentos

No tiene.

#### 2.2.2.6.4 Ejemplo

Código
<pre>Sub prueba_CanCheckIn()     Debug.Print Workbooks(1).CanCheckIn End Sub</pre>
Resultado
<div>Inmediato</div> <div>Falso</div>

### 2.2.2.7 CHANGEFILEACCESS

#### 2.2.2.7.1 Descripción

Permite cambiar los permisos de acceso al libro indicado.

#### 2.2.2.7.2 Sintaxis:

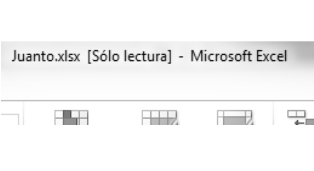

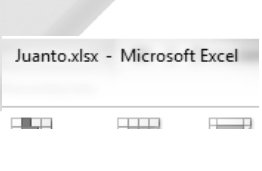

**ChangeFileAccess (Mode, WritePassword, Notify)**

Ra-Ma®

### 2.2.2.7.3 Argumentos

Argumento	Comentario
<b>Mode</b>	Modo de acceso nuevo.
<b>WritePassword</b>	En caso de que el archivo esté protegido contra escritura, especifica contraseña de protección.
<b>Notify</b>	Notifica al usuario si es posible el acceso inmediato al archivo.

### 2.2.2.7.4 Ejemplo

Código			
<pre> Sub prueba_ChangeFileAccess()     Dim miLibro As Excel.Workbook     Set miLibro = Application.Workbooks.Open("c:\temp\Juanto.xlsx")     ' Cambia permisos a sólo escritura.     miLibro.ChangeFileAccess Mode:=xlReadOnly     MsgBox ("pausa")     ' Cambia permisos a lectura / escritura.     miLibro.ChangeFileAccess Mode:=xlReadWrite     MsgBox ("pausa")     Application.Workbooks(2).Close     Set miLibro = Nothing End Sub </pre>			
Resultado			
			

En este ejemplo, puede apreciarse como se abre el libro Juanto.xlsx, se le cambian los permisos de acceso a ‘sólo lectura’ y posteriormente se cambian a ‘lectura/escritura’. Por último se cierra el libro.

## 2.2.2.8 CHANGELINK

### 2.2.2.8.1 Descripción

Cambia un vínculo de un documento a otro.

### 2.2.2.8.2 Sintaxis:

**ChangeLink (Name, NewName, Type)**

### 2.2.2.8.3 Argumentos

Argumento	Comentario
<b>Name</b>	Nombre del vínculo que se desea cambiar.
<b>NewName</b>	Nombre del nuevo vínculo.
<b>Type</b>	Tipo del link que puede ser alguno de los siguientes:  <b>xlLinkTypeExcelLinks</b> Vínculo a un origen de MS Excel.  <b>xlLinkTypeOLELinks</b> Vínculo a un origen OLE.

### 2.2.2.8.4 Ejemplo

Código
<pre> Sub prueba_ChangeLink()     Dim nombre, nuevo As String     nombre = "C:\Cursos\Curso_Macros\Capitulos\08 - El modelo de objetos\Juanto.xlsx"     nuevo = "C:\Cursos\Curso_Macros\Capitulos\08 - El modelo de objetos\Juanto2.xlsx"     Application.Workbooks(1).ChangeLink Name:=nombre, NewName:=nuevo, _     Type:=xlLinkTypeExcelLinks End Sub </pre>

Resultado													
<table> <tr><th></th><th>A</th></tr> <tr><td>1</td><td>10</td></tr> <tr><td>2</td><td>20</td></tr> </table>		A	1	10	2	20	<p>‘Mostrar_fórmulas’</p> <table> <tr><th></th><th>A</th></tr> <tr><td>1</td><td>=[Juanto.xlsx]Hoja1!\$A\$1</td></tr> <tr><td>2</td><td>=[Juanto.xlsx]Hoja1!\$A\$2</td></tr> </table>		A	1	=[Juanto.xlsx]Hoja1!\$A\$1	2	=[Juanto.xlsx]Hoja1!\$A\$2
	A												
1	10												
2	20												
	A												
1	=[Juanto.xlsx]Hoja1!\$A\$1												
2	=[Juanto.xlsx]Hoja1!\$A\$2												
<table> <tr><th></th><th>A</th></tr> <tr><td>1</td><td>1000</td></tr> <tr><td>2</td><td>2000</td></tr> </table>		A	1	1000	2	2000	<p>‘Mostrar_fórmulas’</p> <table> <tr><th></th><th>A</th></tr> <tr><td>1</td><td>=[Juanto2.xlsx]Hoja1!\$A\$1</td></tr> <tr><td>2</td><td>=[Juanto2.xlsx]Hoja1!\$A\$2</td></tr> </table>		A	1	=[Juanto2.xlsx]Hoja1!\$A\$1	2	=[Juanto2.xlsx]Hoja1!\$A\$2
	A												
1	1000												
2	2000												
	A												
1	=[Juanto2.xlsx]Hoja1!\$A\$1												
2	=[Juanto2.xlsx]Hoja1!\$A\$2												

### 2.2.2.9 CHECKIN

#### 2.2.2.9.1 Descripción

Devuelve un libro de un equipo local a un servidor y establece permiso de sólo lectura para que no pueda modificarse en el equipo local. La llamada a este método también cierra el libro.

#### 2.2.2.9.2 Sintaxis:

**CheckIn(SaveChanges, Comments, MakePublic)**

#### 2.2.2.9.3 Argumentos

Argumento	Comentario
<b>SaveChanges</b>	Si se indica True, se protege el documento y se guardan los cambios.
<b>Comments</b>	Comentarios a incluir junto con la protección. Sólo aplicable si <b>SaveChanges</b> es igual a <b>True</b> .
<b>MakePublic</b>	Si se indica True, el documento se publica tras su protección.

### 2.2.2.9.4 Ejemplo

Código
<pre> Sub prueba_CheckIn()     If Workbooks(1).CanCheckIn Then         Workbooks(1).CheckIn SaveChanges:=True, _         Comments:="Mis comentarios", _         MakePublic:=True         Debug.Print "Documento protegido."     Else         Debug.Print "No se puede proteger el documento."     End If End Sub </pre>
Resultado
<div> <div>Inmediato</div> <div>No se puede proteger el documento.</div> </div>

### 2.2.2.10 CHECKINWITHVERSION

#### 2.2.2.10.1 Descripción

Guarda un libro de un equipo local a un servidor y establece el libro local como de solo lectura para que no pueda modificarse localmente.

#### 2.2.2.10.2 Sintaxis:

**CheckInWithVersion**(**SaveChanges**, **Comments**,  
**MakePublic**, **VersionType**)

#### 2.2.2.10.3 Argumentos

Argumento	Comentario
<b>SaveChanges</b>	Si se indica True se guarda el libro en la ubicación del servidor.
<b>Comments</b>	Comentarios a incluir junto con la protección. Sólo aplicable si <b>SaveChanges</b> es igual a <b>True</b> .
<b>MakePublic</b>	Si se indica True, el documento se publica tras su protección.

<b>VersionType</b>	Información sobre versión. Puede escogerse entre alguna de las siguientes:  <b>xlCheckInMajorVersion</b>  <b>xlCheckInMinorVersion</b>  <b>xlCheckInOverwriteVersion</b>
--------------------	--

2.2.2.10.4 Ejemplo

Código
<pre>Sub prueba_CheckInWithVersion()   If Workbooks(1).CanCheckIn Then     Workbooks(1).CheckInWithVersion SaveChanges:=True, _     Comments:="Mis comentarios", _     MakePublic:=True, _     versiontype:=XlCheckInVersionType.xlCheckInOverwriteVersion     Debug.Print "Documento protegido."   Else     Debug.Print "No se puede proteger el documento."   End If End Sub</pre>
Resultado
<div><div>Inmediato</div><div>No se puede proteger el documento.</div></div>

2.2.2.11 CLOSE

2.2.2.11.1 Descripción

Permite cerrar el libro indicado.

2.2.2.11.2 Sintaxis:

Close(SaveChanges, Filename, RouteWorkbook)



### 2.2.2.11.3 Argumentos

Argumento	Comentario
<b>SaveChanges</b>	Si se indica True, significa que el libro se ha de guardar siempre que se haya realizado algún cambio desde que se empezó a trabajar con él o si el libro es nuevo.
<b>Filename</b>	Indica el nombre con el que se desea guardar el libro.
<b>RouteWorkbook</b>	Si se indica True, el libro se distribuirá al destinatario siguiente.

### 2.2.2.11.4 Ejemplo

Código
<pre> Sub prueba_Close()     Dim miLibro As Excel.Workbook     Set miLibro = Application.Workbooks.Open("c:\temp\Juanto.xlsx")     MsgBox ("pausa")     Application.Workbooks(2).Close SaveChanges:=True     Set miLibro = Nothing End Sub </pre>

Este ejemplo muestra como se abre el libro Juanto.xlsx, se realiza una pausa y posteriormente se cierra.

### 2.2.2.12 DELETENUMBERFORMAT

#### 2.2.2.12.1 Descripción

Permite eliminar un formato personalizado de tipo numérico.

#### 2.2.2.12.2 Sintaxis:

**DeleteNumberFormat (NumberFormat)**

#### 2.2.2.12.3 Argumentos

Argumento	Comentario
<b>NumberFormat</b>	Formato a eliminar.

2.2.2.12.4 Ejemplo

Código

```
Sub prueba_DeleteNumberFormat()  
    ActiveWorkbook.DeleteNumberFormat NumberFormat:="0.0000"  
End Sub
```

Resultado

Formato de celdas

Número

Alineación

Fuente

Bordes

Categoría:  
General  
Número  
Moneda  
Contabilidad  
Fecha  
Hora  
Porcentaje  
Fracción  
Científica  
Texto  
Especial  
Personalizada

Muestra  
1000,5000  
Tipo:  
0,0000  
mm:ss  
mm:ss,0  
@  
[h]:mm:ss  
-\* #.##0 €-;  
-\* #.##0 €  
-\* #.##0,00 €  
-\* #.##0,00  
[\$-F400]h:mm:ss  
0,000  
0,0000

Formato de celdas

Número

Alineación

Fuente

Bordes

Categoría:  
General  
Número  
Moneda  
Contabilidad  
Fecha  
Hora  
Porcentaje  
Fracción  
Científica  
Texto  
Especial  
Personalizada

Muestra  
1000,500  
Tipo:  
0,000  
dd/mm/aaaa  
mm:ss  
mm:ss,0  
@  
[h]:mm:ss  
-\* #.##0 €-;  
-\* #.##0 €  
-\* #.##0,00 €  
-\* #.##0,00  
-\$-F400]h:mm:ss  
0,000

En este ejemplo se elimina el formato numérico “0.0000”

2.2.2.13 ENABLECONNECTIONS

2.2.2.13.1 Descripción

Este método permite habilitar conexiones de datos en el libro.

2.2.2.13.2 Sintaxis:

EnableConnections

2.2.2.13.3 Argumentos

No tiene.

#### 2.2.2.13.4 Ejemplo

Código
<pre>Sub prueba_EnableConnections()     Workbooks(1).EnableConnections End Sub</pre>

#### 2.2.2.14 ENDREVIEW

##### 2.2.2.14.1 Descripción

Finaliza la revisión de un archivo enviado para su revisión.

##### 2.2.2.14.2 Sintaxis:

**EndReview**

##### 2.2.2.14.3 Argumentos

No tiene.

##### 2.2.2.14.4 Ejemplo

Código
<pre>Sub prueba_EndReview()     ActiveWorkbook.EndReview End Sub</pre>

#### 2.2.2.15 EXCLUSIVEACCESS

##### 2.2.2.15.1 Descripción

Permite asignar al usuario actual el acceso exclusivo a un libro compartido.

##### 2.2.2.15.2 Sintaxis:

**ExclusiveAccess**

##### 2.2.2.15.3 Argumentos

No tiene.

#### 2.2.2.15.4 Ejemplo

Código
<pre>Sub prueba_ExclusiveAccess()     ActiveWorkbook.ExclusiveAccess End Sub</pre>

#### 2.2.2.16 EXPORTASFIXEDFORMAT

##### 2.2.2.16.1 Descripción

Permite publicar un libro en formato pdf o xps. Para ello, se ha de disponer del complemento pdf instalado.


##### 2.2.2.16.2 Sintaxis:

```
ExportAsFixedFormat (Type,      Filename,      Quality,
IncludeDocProperties,  IgnorePrintAreas,  From,      To,
OpenAfterPublish, FixedFormatExtClassPtr)
```

##### 2.2.2.16.3 Argumentos

Argumento	Comentario
<b>Type</b>	Tipo a utilizar para la exportación. Puede ser: <b>xlTypePDF</b> o <b>xlTypeXPS</b> .
<b>Filename</b>	Nombre del archivo con el que se creará la exportación.
<b>Quality</b>	Calidad con la que se creará el documento. Puede ser: <b>xlQualityStandard</b> o <b>xlQualityMinimum</b> .
<b>IncludeDocProperties</b>	Si se indica True, se incluyen las propiedades en los documentos.
<b>IgnorePrintAreas</b>	Si se indica True se omiten las áreas de impresión.
<b>From</b>	Indica el número de la primera página a publicar.
<b>To</b>	Indica el número de la última página a publicar.
<b>OpenAfterPublish</b>	Si se indica True, el documento resultante se abre una vez exportado.
<b>FixedFormatExtClassPtr</b>	Puntero a la clase FixedFormatExt.

### 2.2.2.16.4 Ejemplo

Código
<pre> Sub prueba_ExportAsFixedFormat ()     ' Se exporta a pdf     ActiveWorkbook.ExportAsFixedFormat Type:=xlTypePDF, _     Filename:="c:\temp\Juanto.pdf", _     Quality:=xlQualityStandard, _     OpenAfterPublish:=True     ' Se exporta a xps y abre documento.     ActiveWorkbook.ExportAsFixedFormat Type:=xlTypeXPS, _     Filename:="c:\temp\Juanto.xps", _     Quality:=xlQualityStandard, _     OpenAfterPublish:=True End Sub </pre>
Resultado


Este ejemplo, exporta el libro activo a pdf y xps con un determinado nombre y posteriormente abre los documentos generados.

### 2.2.2.17 FOLLOWHYPERLINK

#### 2.2.2.17.1 Descripción

Muestra el documento indicado en la aplicación correspondiente tanto si está en la caché como si ha de obtenerlo directamente de la ubicación indicada.

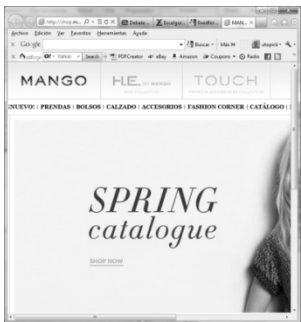

#### 2.2.2.17.2 Sintaxis:

**FollowHyperlink** (**Address**, **SubAddress**, **NewWindow**, **AddHistory**, **ExtraInfo**, **Method**, **HeaderInfo**)

### 2.2.2.17.3 Argumentos

Argumento	Comentario
<b>Address</b>	Dirección del documento destino.
<b>SubAddress</b>	Ubicación dentro del documento.
<b>NewWindow</b>	Abre una nueva ventana para mostrar el documento.
<b>AddHistory</b>	No utilizado.
<b>ExtraInfo</b>	Información adicional para resolver el hipervínculo.
<b>Method</b>	Modo en el que ExtraInfo es adjuntado.
<b>HeaderInfo</b>	Información de encabezado para la solicitud HTTP.

### 2.2.2.17.4 Ejemplo

Código	
<pre>Sub prueba_FollowHyperlink()     ActiveWorkbook.FollowHyperlink Address:="http://www.mango.com"     ActiveWorkbook.FollowHyperlink Address:="c:\temp\index_ph.docx", NewWindow:=True     ActiveWorkbook.FollowHyperlink Address:="c:\temp\juanto.pdf", NewWindow:=True End Sub</pre>	
Resultado	
	

## 2.2.2.18 GETWORKFLOWTASKS

### 2.2.2.18.1 Descripción

Proporciona una colección de objetos de tipo WorkflowTask del libro indicado.

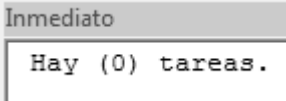
### 2.2.2.18.2 Sintaxis:

**GetWorkflowTasks**

### 2.2.2.18.3 Argumentos

No tiene.

### 2.2.2.18.4 Ejemplo

Código
<pre>Sub prueba_WorkflowTasks()     Dim l As WorkflowTasks     Set l = ActiveWorkbook.GetWorkflowTasks     Debug.Print "Hay (" &amp; l.Count &amp; ") tareas." End Sub</pre>
Resultado


## 2.2.2.19 GETWORKFLOWTEMPLATES

### 2.2.2.19.1 Descripción

Proporciona una colección de objetos de tipo WorkflowTemplate del libro indicado.

### 2.2.2.19.2 Sintaxis:

**GetWorkflowTemplates**

### 2.2.2.19.3 Argumentos

No tiene.

### 2.2.2.19.4 Ejemplo

Código
<pre>Sub prueba_GetWorkflowTemplates()     Debug.Print "Hay (" &amp; ActiveWorkbook.GetWorkflowTemplates.Count     &amp; ") plantillas."     Dim w As Office.workflowTemplates     Set w = ActiveWorkbook.GetWorkflowTemplates     Dim template As WorkflowTemplate     For Each template In w         Debug.Print template.Description     Next End Sub</pre>
Resultado
<div> <div>Inmediato</div> <div>Hay (0) plantillas.</div> </div>

## 2.2.2.20 HIGHLIGHTCHANGESOPTIONS

### 2.2.2.20.1 Descripción

Controla cómo se muestran los cambios en un libro compartido.

### 2.2.2.20.2 Sintaxis:

**HighlightChangesOptions(When, Who, Where)**

### 2.2.2.20.3 Argumentos

Argumento	Comentario
<b>When</b>	<p>Indica qué cambios se han de mostrar utilizando alguna de las siguientes constantes:</p> <p><b>xlAllChanges:</b> Muestra todos los cambios.</p> <p><b>xlNotYetReviewed:</b> Muestra sólo los cambios pendientes de revisión.</p> <p><b>xlSinceMyLastSave:</b> Muestra los cambios realizados desde que el último usuario guardó el libro por última vez.</p>



<b>Who</b>	Indica qué usuario o usuarios se han de mostrar. Puede tener alguno de los siguientes valores:  <b>Todos.</b>  <b>Todos excepto yo.</b>  <b>Nombre usuario.</b>
<b>Where</b>	Area o Rango que se desea analizar.

#### 2.2.2.20.4 Ejemplo

Código
<pre> Sub prueba_HighlightChangesOptions()     Dim miLibro As Excel.Workbook     Set miLibro = Application.Workbooks.Open("C:\temp\Juanto.xlsx")     miLibro.HighlightChangesOptions When:=xlSinceMyLastSave,     Who:="Todos", Where:="A1:D3"     MsgBox ("pausa")     Application.Workbooks(2).Close SaveChanges:=True     Set miLibro = Nothing End Sub </pre>

#### 2.2.2.21 LINKINFO

##### 2.2.2.21.1 Descripción

Devuelve información acerca de la fecha y el estado de actualización del vínculo.

##### 2.2.2.21.2 Sintaxis:

**LinkInfo(Name, LinkInfo, Type, EditionRef)**

Ra-Ma®

### 2.2.2.21.3 Argumentos

Argumento	Comentario
<b>Name</b>	Nombre del vínculo.
<b>LinkInfo</b>	<p>Tipo de información deseado. Puede ser alguno de los siguientes:</p> <p><b>xlEditionDate:</b> Sólo para Macintosh.</p> <p><b>xlLinkInfoStatus:</b> Indica estado del vínculo.</p> <p><b>xlUpdateState:</b> Indica tipo de actualización del vínculo ( automática o manual).</p>
<b>Type</b>	<p>Tipo de vínculo a devolver que puede ser alguno de los siguientes:</p> <p><b>xlLinkInfoOLELinks:</b> Servidor OLE o DDE</p> <p><b>xlLinkInfoPublishers:</b> Publicación</p> <p><b>xlLinkInfoSubscribers:</b> Suscriptor</p>
<b>EditionRef</b>	Indica la referencia a la edición con el estilo FIC1 si el vínculo es una edición.

### 2.2.2.21.4 Ejemplo

Código
<pre> Sub prueba_LinkInfo()     Dim l As Variant     For Each l In ActiveWorkbook.LinkSources         Debug.Print l &amp; " (" &amp; ActiveWorkbook.LinkInfo(l, xlLinkInfoStatus) &amp; ")"     Next End Sub </pre>

Resultado
<p>Inmediato</p> <pre>C:\Cursos\Curso_Macros\Capitulos\08 - El modelo de objetos\Juanto.xlsx (3) C:\Temp\JuantoDDE.xlsx (3)</pre>

## 2.2.2.22 LINKSOURCES

### 2.2.2.22.1 Descripción

Proporciona una colección en forma de matriz de los vínculos (DDE y OLE) que posee el libro indicado.

### 2.2.2.22.2 Sintaxis:

**LinkSources (Type)**

### 2.2.2.22.3 Argumentos

Argumento	Comentario
<b>Type</b>	<p>Tipo de vínculo que puede ser alguno de los siguientes:</p> <p><b>xlExcelLinks:</b> El vínculo es con una hoja de Excel.</p> <p><b>xlOLELinks:</b> El vínculo es con un origen de datos OLE.</p> <p><b>xlPublishers:</b> Sólo Macintosh.</p> <p><b>xlSubscribers:</b> Sólo Macintosh.</p>

### 2.2.2.22.4 Ejemplo

Código
<pre>Sub prueba_LinkSources()     Dim l As Variant     For Each l In ActiveWorkbook.LinkSources         Debug.Print "Link: (" &amp; l &amp; ")"     Next End Sub</pre>

Resultado
<p>Inmediato</p> <pre>Link: (C:\Cursos\Curso_Macros\Capitulos\08 - El modelo de objetos\Juanto.xlsx) Link: (C:\Temp\JuantoDDE.xlsx)</pre>

### 2.2.2.23 LOCKSERVERFILE

#### 2.2.2.23.1 Descripción

Permite bloquear el libro en el servidor impidiendo su modificación.

#### 2.2.2.23.2 Sintaxis:

**LockServerFile**

#### 2.2.2.23.3 Argumentos

No tiene.

#### 2.2.2.23.4 Ejemplo

Código
<pre>Sub prueba_LockServerFile()     ActiveWorkbook.LockServerFile End Sub</pre>

### 2.2.2.24 MERGEWORKBOOK

#### 2.2.2.24.1 Descripción

Permite combinar los cambios que se hayan producido en una copia de un libro compartido sobre el libro abierto que ha sido copiado.

#### 2.2.2.24.2 Sintaxis:

**MergeWorkbook (Filename)**

### 2.2.2.24.3 Argumentos

Argumento	Comentario
<b>Filename</b>	Nombre del fichero que tiene los cambios a combinar con el libro abierto actual.

### 2.2.2.24.4 Ejemplo

Código																																																								
<pre>Sub prueba_MergeWorkbook()     Dim miLibro As Excel.Workbook     Set miLibro = Application.Workbooks.Open("C:\temp\Juanto.xlsx")     miLibro.MergeWorkbook Filename:="C:\temp\Juanto_cmp.xlsx"     Application.Workbooks(2).Close SaveChanges:=True     Set miLibro = Nothing End Sub</pre>																																																								
Resultado																																																								
Juanto.xlsx (Inicio)	Juanto_cmp.xlsx	Juanto.xlsx (Final)																																																						
<table border="1"> <thead> <tr> <th></th><th>A</th><th>B</th></tr> </thead> <tbody> <tr><td>1</td><td>A</td><td>Antes1</td></tr> <tr><td>2</td><td>B</td><td>Antes2</td></tr> <tr><td>3</td><td>C</td><td>Antes3</td></tr> <tr><td>4</td><td></td><td></td></tr> <tr><td>5</td><td></td><td></td></tr> </tbody> </table>		A	B	1	A	Antes1	2	B	Antes2	3	C	Antes3	4			5			<table border="1"> <thead> <tr> <th></th><th>A</th><th>B</th></tr> </thead> <tbody> <tr><td>1</td><td></td><td>Antes1</td></tr> <tr><td>2</td><td>B</td><td>Nuevo2</td></tr> <tr><td>3</td><td></td><td>Antes3</td></tr> <tr><td>4</td><td>D</td><td></td></tr> <tr><td>5</td><td></td><td>Nuevo5</td></tr> </tbody> </table>		A	B	1		Antes1	2	B	Nuevo2	3		Antes3	4	D		5		Nuevo5	<table border="1"> <thead> <tr> <th></th><th>A</th><th>B</th></tr> </thead> <tbody> <tr><td>1</td><td></td><td>Antes1</td></tr> <tr><td>2</td><td>B</td><td>Nuevo2</td></tr> <tr><td>3</td><td></td><td>Antes3</td></tr> <tr><td>4</td><td>D</td><td></td></tr> <tr><td>5</td><td></td><td>Nuevo5</td></tr> </tbody> </table>		A	B	1		Antes1	2	B	Nuevo2	3		Antes3	4	D		5		Nuevo5
	A	B																																																						
1	A	Antes1																																																						
2	B	Antes2																																																						
3	C	Antes3																																																						
4																																																								
5																																																								
	A	B																																																						
1		Antes1																																																						
2	B	Nuevo2																																																						
3		Antes3																																																						
4	D																																																							
5		Nuevo5																																																						
	A	B																																																						
1		Antes1																																																						
2	B	Nuevo2																																																						
3		Antes3																																																						
4	D																																																							
5		Nuevo5																																																						

En este ejemplo, el libro Juanto\_cmp.xlsx es una copia del libro compartido Juanto.xlsx al que se le han realizado diversos cambios (eliminar contenido de A1, incluir 'D' en la celda A4, etc). Al ejecutar el procedimiento, observamos que los cambios realizados sobre Juanto\_cmp.xlsx se repercuten en Juanto.xlsx

### 2.2.2.25 NEWWINDOW

#### 2.2.2.25.1 Descripción

Permite crear una ventana del libro indicado.

### 2.2.2.25.2 Sintaxis:

**NewWindow**

### 2.2.2.25.3 Argumentos

No tiene.

### 2.2.2.25.4 Ejemplo

Código
<pre>Sub prueba_NewWindow()     ActiveWorkbook.NewWindow End Sub</pre>
Resultado



### 2.2.2.26 OPENLINKS

#### 2.2.2.26.1 Descripción

Abre los documentos auxiliares de uno o más vínculos.

Permite abrir los documentos que contienen los vínculos utilizados por el libro indicado.

#### 2.2.2.26.2 Sintaxis:

**OpenLinks (Name, ReadOnly, Type)**

### 2.2.2.26.3 Argumentos

Argumento	Comentario
<b>Name</b>	Nombre del fichero o vínculo a abrir.
<b>ReadOnly</b>	Si indicamos True, se abre sólo de lectura.
<b>Type</b>	<p>Tipo de vínculo que se desea abrir y que puede ser alguno de los siguientes:</p> <p><b>xlExcelLinks</b>: El vínculo es de tipo MS Excel.</p> <p><b>xlOLELinks</b>: El vínculo es de tipo OLE.</p> <p><b>xlPublishers</b>: Sólo Macintosh.</p> <p><b>xlSubscribers</b>: Sólo Macintosh.</p>

### 2.2.2.26.4 Ejemplo

Código
<pre> Sub prueba_OpenLinks()     Dim Links(1 To 5)     Dim i, n As Integer     Dim l As Variant     For Each l In ActiveWorkbook.LinkSources         i = i + 1         Links(i) = l     Next     For n = 1 To i         Debug.Print Links(n)     Next n     ActiveWorkbook.OpenLinks Name:=Links(1) End Sub </pre>
Resultado
<p>Inmediato</p> <pre> C:\Cursos\Curso_Macros\Capitulos\08 - El modelo de objetos\Juanto.xlsx C:\Temp\JuantoDDE.xlsx </pre>

## 2.2.2.27 PIVOTCACHES

### 2.2.2.27.1 Descripción

Devuelve una colección PivotCaches que representa todas las memorias caché de tabla dinámica del libro especificado. Solo lectura.

Proporciona una colección de objetos de tipo PivotCaches. PivotCache representa la colección de memorias utilizadas en las tablas dinámicas existentes en el libro.

### 2.2.2.27.2 Sintaxis:

**PivotCaches**

### 2.2.2.27.3 Argumentos

No tiene.

### 2.2.2.27.4 Ejemplo

Código
<pre>Sub prueba_PivotCaches()     Debug.Print ActiveWorkbook.PivotCaches.Count     Debug.Print ActiveWorkbook.PivotCaches(1).EnableRefresh End Sub</pre>
Resultado
<div>Inmediato</div> <div>1</div> <div>Verdadero</div>

En este ejemplo, se indica cuantos **PivotCaches** hay en el libro y si el usuario puede refrescar la tabla del primer **PivotCache**.

## 2.2.2.28 POST

### 2.2.2.28.1 Descripción

Envía el libro especificado a una carpeta pública. Este método sólo está disponible en un cliente de Microsoft Exchange que esté conectado con un servidor de Microsoft Exchange.



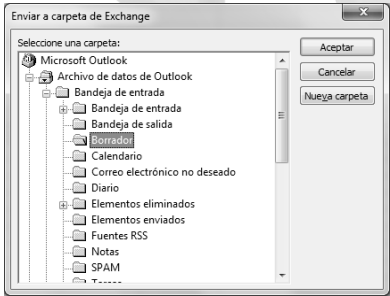
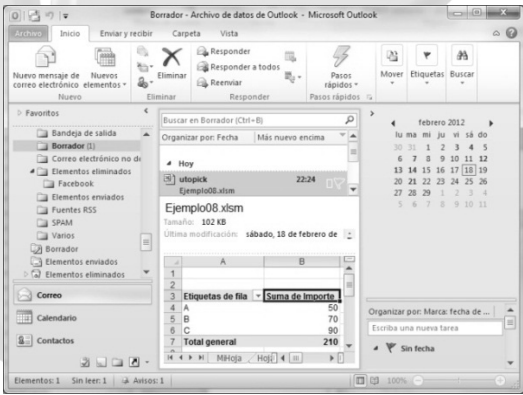
### 2.2.2.28.2 Sintaxis:

**Post (DestName)**

### 2.2.2.28.3 Argumentos

Argumento	Comentario
<b>DestName</b>	Destino del libro. Si no se indica, durante la ejecución se solicitará al usuario.

### 2.2.2.28.4 Ejemplo

Código	
<pre>Sub prueba_Post ()     ActiveWorkbook.Post End Sub</pre>	
Resultado	
	
	

Al ejecutar este procedimiento, se abre un cuadro de diálogo para que podamos seleccionar la carpeta a la que se enviará el libro. A continuación se puede ver cómo efectivamente podemos visualizar el libro en la carpeta 'Borrador'.

Ra-Ma®

## 2.2.2.29 PRINTOUT

### 2.2.2.29.1 Descripción

Permite imprimir el libro eligiendo el rango de páginas a imprimir, el número de copias, la impresora a utilizar, etc...

### 2.2.2.29.2 Sintaxis:

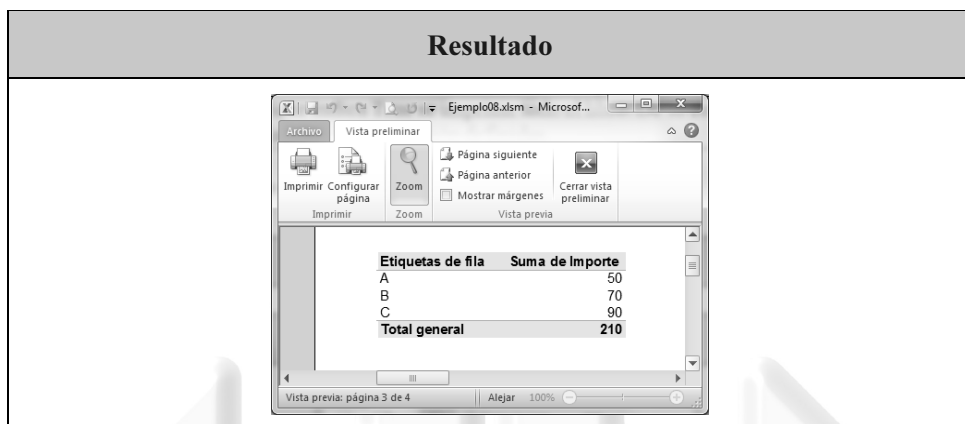
`PrintOut(From, To, Copies, Preview, ActivePrinter, PrintToFile, Collate, PrToFileName, IgnorePrintAreas)`

### 2.2.2.29.3 Argumentos

Argumento	Comentario
<b>From</b>	Primera página a imprimir.
<b>To</b>	Ultima página a imprimir.
<b>Copies</b>	Número de copias.
<b>Preview</b>	Si se indica True se realiza una vista preliminar.
<b>ActivePrinter</b>	Nombre de la impresora a utilizar.
<b>PrintToFile</b>	Si se indica True se la impresión se envía a un archivo.
<b>Collate</b>	Si se indica True se pueden intercalar varias copias.
<b>PrToFileName</b>	Indica el nombre del fichero hacia el que se desea enviar la impresión.
<b>IgnorePrintAreas</b>	Si se indica True se imprime todo el libro. De lo contrario, se imprimen sólo las áreas de impresión definidas.

### 2.2.2.29.4 Ejemplo

Código
<pre>Sub prueba_PrintOut()     ActiveWorkbook.PrintOut Preview:=True End Sub</pre>



En este ejemplo, se imprime el libro por la impresora estándar pero previamente se realiza una previsualización del mismo.

## 2.2.2.30 PRINTPREVIEW

### 2.2.2.30.1 Descripción

Permite realizar una previsualización del libro para ver cómo quedaría impreso.

### 2.2.2.30.2 Sintaxis:

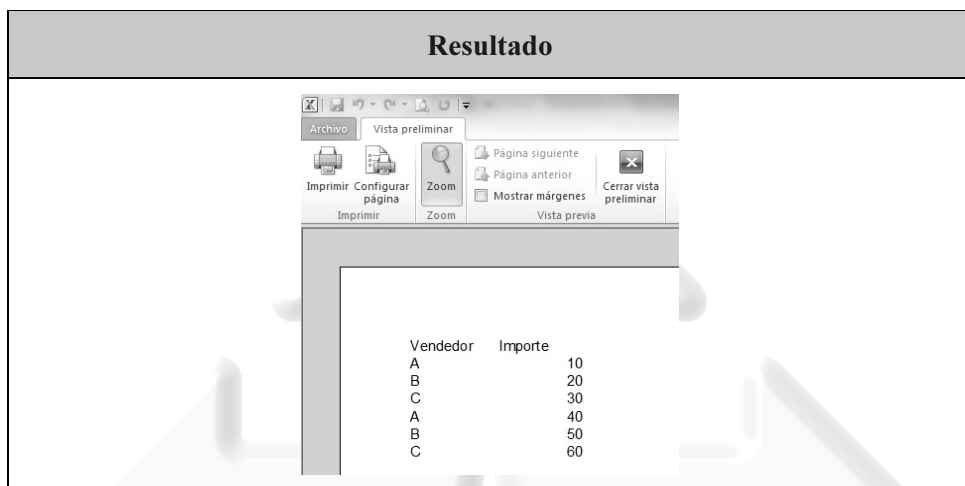
**PrintPreview(EnableChanges)**

### 2.2.2.30.3 Argumentos

Argumento	Comentario
<b>EnableChanges</b>	Si se indica True, el usuario puede realizar cambios durante la vista preliminar en ciertos parámetros como son los márgenes, la orientación de la página, el encabezado y pie, etc... Se trata de un parámetro opcional.

### 2.2.2.30.4 Ejemplo

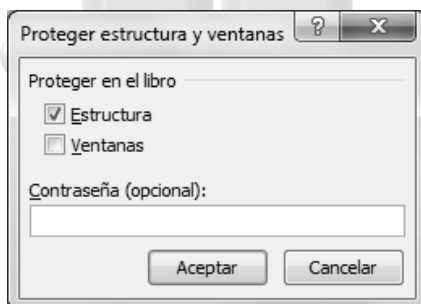
Código
<pre>Sub prueba_PrintPreview()     ActiveWorkbook.PrintPreview EnableChanges:=True End Sub</pre>



### 2.2.2.31 PROTECT

#### 2.2.2.31.1 Descripción

Protege un libro para que no se pueda modificar. Tiene la misma funcionalidad que conseguimos manualmente a partir de la opción **Proteger libro** ubicada en la pestaña **Revisar**:



#### 2.2.2.31.2 Sintaxis:

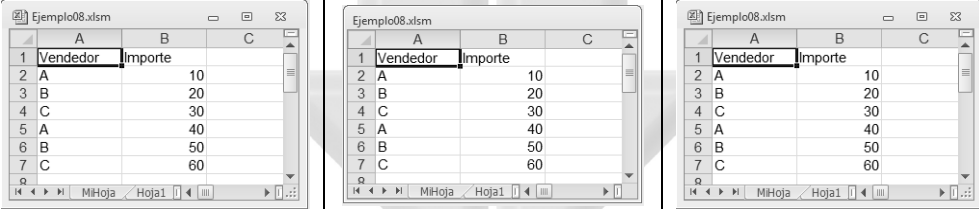
`Protect(Password, Structure, Windows)`

Ra-Ma®

### 2.2.2.31.3 Argumentos

Argumento	Comentario
<b>Password</b>	Permite proteger el libro mediante contraseña (si se desea).
<b>Structure</b>	Si se indica True, se protege las estructura del libro (nombre de las hojas, color, posición, etc...).
<b>Windows</b>	Si se indica True, se protege la posición y tamaño de las ventanas.

### 2.2.2.31.4 Ejemplo

Código
<pre> Sub prueba_Protect()     ActiveWorkbook.Protect Windows:=True     MsgBox ("Pausa")     ActiveWorkbook.Protect Windows:=False     MsgBox ("Pausa") End Sub </pre>
Resultado


En este ejemplo, protegemos las ventanas del libro y las volvemos a desproteger haciendo una pausa para entre cada acción para poder comprobar el resultado de cada acción.

Efectivamente, podemos ver que cuando está protegida desaparece el borde mediante el cual puede redimensionarse la ventana y también desaparecen los botones de **Minimizar**, **Maximizar/Restaurar** y **Cerrar**.

## 2.2.2.32 PROTECTSHARING

### 2.2.2.32.1 Descripción

Permite guardar el libro protegiéndolo con contraseña, impidiendo que sea compartido y sugiriendo que se abra sólo en modo lectura.

### 2.2.2.32.2 Sintaxis:

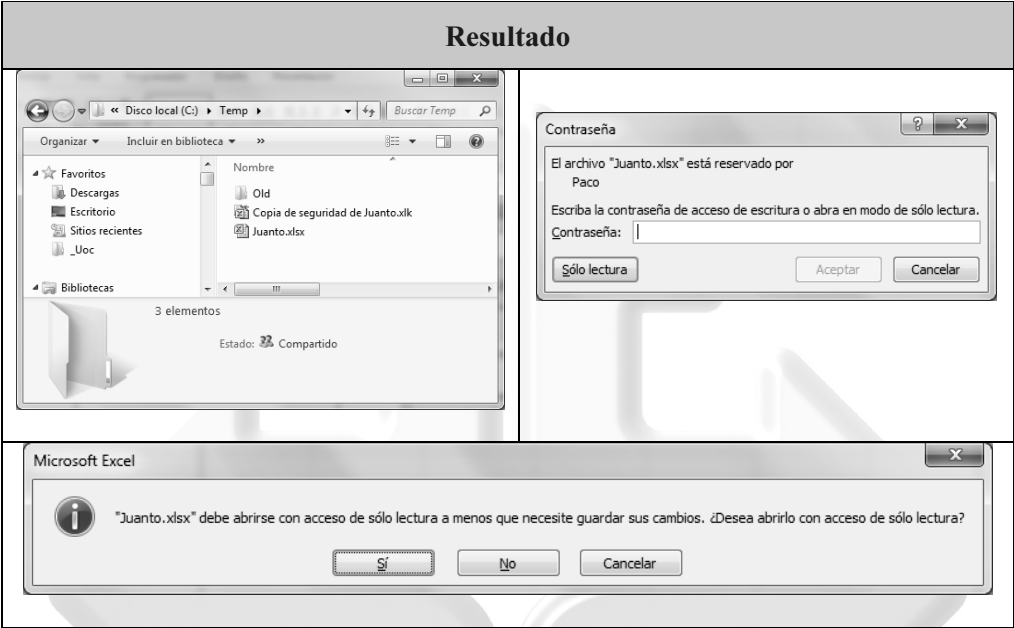
**ProtectSharing**(Filename, Password, WriteResPassword, ReadOnlyRecommended, CreateBackup, SharingPassword, FileFormat)

### 2.2.2.32.3 Argumentos

Argumento	Comentario
<b>Filename</b>	Nombre del archivo guardado.
<b>Password</b>	Contraseña a utilizar para proteger el archivo.
<b>WriteResPassword</b>	Contraseña contra escritura.
<b>ReadOnlyRecommended</b>	Si se indica True, al abrir el archivo se recomendará abrirlo sólo de lectura.
<b>CreateBackup</b>	Si se indica True se crea una copia de seguridad al abrir el archivo.
<b>SharingPassword</b>	Contraseña para impedir compartir el libro.
<b>FileFormat</b>	Formato del archivo.

### 2.2.2.32.4 Ejemplo

Código
<pre> Sub prueba_ProtectSharing()     Dim miLibro As Excel.Workbook     Set miLibro = Application.Workbooks.Open("C:\temp\Juanto.xlsx")     miLibro.ProtectSharing WriteResPassword:="mipwd", _         ReadOnlyRecommended:=True, _         CreateBackup:=True     miLibro.Close SaveChanges:=True     Set miLibro = Nothing End Sub </pre>



En este ejemplo podemos ver cómo se crea una copia de seguridad del archivo, cómo se establece una contraseña en el momento de su apertura y de cómo se sugiere que el libro sea abierto en modo de sólo lectura.

2.2.2.33 PURGECHANGEHISTORYNOW

2.2.2.33.1 Descripción

Permite eliminar las entradas del registro de cambio.

2.2.2.33.2 Sintaxis:

PurgeChangeHistoryNow(Days, SharingPassword)

2.2.2.33.3 Argumentos

Argumento	Comentario
Days	Número de días que hay que mantener los cambios en el registro de cambios.
SharingPassword	Contraseña a utilizar para desproteger el libro.

#### 2.2.2.33.4 Ejemplo

Código
<pre>Sub prueba_PurgeChangeHistoryNow()     Dim miLibro As Excel.Workbook     Set miLibro = Application.Workbooks.Open("C:\temp\Juanto.xlsx")     miLibro.PurgeChangeHistoryNow Days:=10     miLibro.Close SaveChanges:=True     Set miLibro = Nothing End Sub</pre>

En este ejemplo se eliminan los cambios del libro **Juanto.xlsx** que tienen más de 10 días.

#### 2.2.2.34 REFRESHALL

##### 2.2.2.34.1 Descripción

Actualiza todos los rangos de datos externos y los informes de tabla dinámica del libro especificado.

##### 2.2.2.34.2 Sintaxis:

**RefreshAll**

##### 2.2.2.34.3 Argumentos

No tiene.

##### 2.2.2.34.4 Ejemplo

Código
<pre>Sub prueba_RefreshAll()     ActiveWorkbook.RefreshAll End Sub</pre>

Ra-Ma®



Resultado

Situación inicial

	A	B
1	Vendedor	Importe
2	A	10
3	B	20
4	C	30
5	A	40
6	B	50
7	C	60

	A	B
1		
2		
3	Etiquetas de fila ▾ Suma de Importe	
4	A	50
5	B	70
6	C	90
7	Total general	210

Tras cambiar los datos

	A	B
1	Vendedor	Importe
2	A	10
3	B	20
4	C	30
5	A	40
6	B	500
7	C	60

	A	B
1		
2		
3	Etiquetas de fila ▾ Suma de Importe	
4	A	50
5	B	70
6	C	90
7	Total general	210

Tras ejecutar el procedimiento

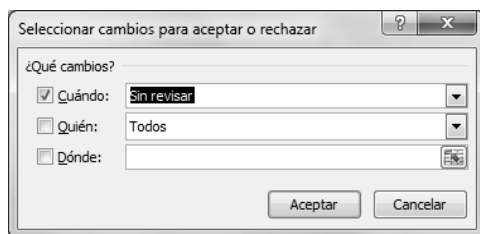
	A	B
1		
2		
3	Etiquetas de fila ▾ Suma de Importe	
4	A	50
5	B	520
6	C	90
7	Total general	660

En este ejemplo podemos observar que tras cambiar los datos que sirven como fuente para la tabla dinámica, dicha tabla no refleja los nuevos valores hasta que ejecutamos el procedimiento.

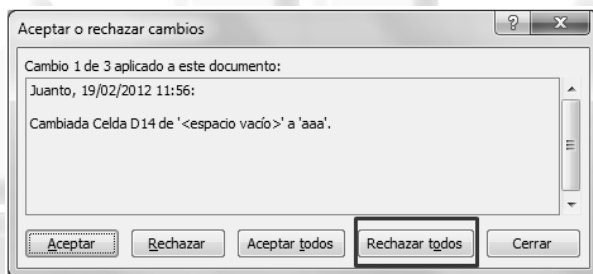
### 2.2.2.35 REJECTALLCHANGES

#### 2.2.2.35.1 Descripción

Permite rechazar los cambios del libro compartido en función de cuando, quien y donde se hicieron. Equivale a invocar la opción **Control de cambios** situado en la pestaña de **Revisar**:



...y posteriormente rechazar los cambios a través del cuadro de diálogo que aparece en la opción de Aceptar o rechazar cambios:



#### 2.2.2.35.2 Sintaxis:

**RejectAllChanges (When, Who, Where)**

#### 2.2.2.35.3 Argumentos

Argumento	Comentario
<b>When</b>	Cuando se han de rechazar los cambios.
<b>Who</b>	Quien ha de rechazar los cambios. Puede ser alguno de los siguientes valores:  <b>Todos.</b>  <b>Todos excepto yo.</b>  <b>Nombre usuario.</b>
<b>Where</b>	Area o Rango donde se han de rechazar los cambios.

### 2.2.2.35.4 Ejemplo

Código
<pre>Sub prueba_RejectAllChanges()     Dim miLibro As Excel.Workbook     Set miLibro = Application.Workbooks.Open("C:\temp\Juanto.xlsx")     miLibro.RejectAllChanges Who:="Paco", Where:="A1:A3"     miLibro.Close SaveChanges:=True     Set miLibro = Nothing End Sub</pre>

En este ejemplo, se rechazan todos los cambios realizados por **Paco** en el rango “**A1:A3**” del libro compartido **Juanto.xlsx** .

### 2.2.2.36 RELOADAS

#### 2.2.2.36.1 Descripción

Permite utilizar una determinada codificación a la hora de cargar un libro basado en un documento HTML.

#### 2.2.2.36.2 Sintaxis:

**ReloadAs (Encoding)**

#### 2.2.2.36.3 Argumentos

Argumento	Comentario
<b>Encoding</b>	Codificación a utilizar para cargar el libro (ver MsoEncoding Enumeration).

### 2.2.2.36.4 Ejemplo

Código
<pre>Sub prueba_ReloadAs()     Dim miLibro As Excel.Workbook     Set miLibro = Application.Workbooks.Open("C:\temp\Juanto.html")     miLibro.ReloadAs Encoding:=msoEncodingUTF8     Set miLibro = Nothing End Sub</pre>

## 2.2.2.37 REMOVEDOCUMENTINFORMATION

### 2.2.2.37.1 Descripción

Quita del libro toda la información del tipo especificado.

Permite eliminar información del tipo indicado en el libro.

### 2.2.2.37.2 Sintaxis:

**RemoveDocumentInformation (RemoveDocInfoType)**

### 2.2.2.37.3 Argumentos

Argumento	Comentario
<b>RemoveDocInfoType</b>	El tipo de información a eliminar. Para mas detalles, consultar enumeración <b>XIRemoveDocInfoType</b> .

### 2.2.2.37.4 Ejemplo

Código
<pre>Sub prueba_RemoveDocumentInformation()     Dim miLibro As Excel.Workbook     Set miLibro = Application.Workbooks.Open("C:\temp\Juanto.xlsx")     miLibro.RemoveDocumentInformation xlRDIDefinedNameComments     miLibro.RemoveDocumentInformation xlRDIRemovePersonalInformation     miLibro.Close SaveChanges:=True     Set miLibro = Nothing End Sub</pre>

En este ejemplo, se elimina la información del libro Juanto.xlsx relacionada con los comentarios con nombre definidos y con la información personal.

## 2.2.2.38 REMOVEUSER

### 2.2.2.38.1 Descripción

Desconecta al usuario especificado del libro compartido.

Permite desconectar a un usuario del libro indicado.

**2.2.2.38.2 Sintaxis:****RemoveUser (Index)****2.2.2.38.3 Argumentos**

Argumento	Comentario
<b>Index</b>	Indice que determina el usuario a desconectar.

**2.2.2.38.4 Ejemplo**

Código
<pre> Sub prueba_RemoveUser()     Dim miLibro As Excel.Workbook     Set miLibro = Application.Workbooks.Open("C:\temp\Juanto.xlsx")     Dim users As Variant     Dim nomUser, fecha, tipo As String     Dim nUser As Integer     users = ActiveWorkbook.UserStatus     For nUser = 1 To UBound(users, 1)         nomUser = users(nUser, 1)         fecha = users(nUser, 2)         Select Case users(nUser, 3)             Case 1                 tipo = "Exclusive"             Case 2                 tipo = "Shared"         End Select         Debug.Print nomUser &amp; " " &amp; fecha &amp; " " &amp; tipo     Next     miLibro.RemoveUser (1)     miLibro.Close     Set miLibro = Nothing End Sub </pre>

Ra-Ma®



Este ejemplo permite visualizar qué usuarios están conectados a un libro compartido, en qué momento lo hicieron y en qué modo. A continuación, vemos como se desconecta un usuario y el mensaje que recibe como consecuencia de su desconexión.

### 2.2.2.39 REPLYWITHCHANGES

#### 2.2.2.39.1 Descripción

Permite enviar un mensaje al autor del libro recibido para su revisión indicándole que dicha revisión ya ha finalizado.

#### 2.2.2.39.2 Sintaxis:

**ReplyWithChanges (ShowMessage)**

#### 2.2.2.39.3 Argumentos

Argumento	Comentario
<b>ShowMessage</b>	Si indicamos True se muestra la ventana del mensaje para que podamos acabar de configurarlo.

### 2.2.2.39.4 Ejemplo

Código
<pre>Sub prueba_ReplyWithChanges()     Dim miLibro As Excel.Workbook     Set miLibro = Application.Workbooks.Open("C:\temp\Juanto.xlsx")     miLibro.ReplyWithChanges showmessage:=False     miLibro.Close     Set miLibro = Nothing End Sub</pre>

### 2.2.2.40 RESETCOLORS

#### 2.2.2.40.1 Descripción

Restaura los colores por defecto de la paleta.

#### 2.2.2.40.2 Sintaxis:

**ResetColors**

#### 2.2.2.40.3 Argumentos

No tiene.

#### 2.2.2.40.4 Ejemplo

Código
<pre>Sub prueba_ResetColors()     ActiveWorkbook.ResetColors End Sub</pre>

### 2.2.2.41 RUNAUTOMACROS

#### 2.2.2.41.1 Descripción

Ejecuta la macro Auto\_Open, Auto\_Close, Auto\_Activate o Auto\_Deactivate adjunta al libro. Este método se incluye por motivos de compatibilidad con versiones anteriores. En el nuevo código de Visual Basic debería utilizar los eventos Open, Close, Activate y Deactivate en lugar de las macros anteriores.


### 2.2.2.41.2 Sintaxis:

**RunAutoMacros (Which)**

### 2.2.2.41.3 Argumentos

Argumento	Comentario
<b>Which</b>	Macro a ejecutar.

### 2.2.2.41.4 Ejemplo

Código
<pre>Sub prueba_RunAutoMacros()     Dim miLibro As Excel.Workbook     Set miLibro = Application.Workbooks.Open("C:\temp\Ventas.xlsm")     miLibro.RunAutoMacros xlAutoOpen     MsgBox ("Pausa")     miLibro.Close     Set miLibro = Nothing End Sub</pre>
Resultado


En este ejemplo, puede apreciarse como se abre el libro Ventas.xlsm, como se ejecuta el método Auto\_Open en el que simplemente se ha colocado un mensaje, y a continuación como se muestra el mensaje indicando ‘Pausa’ que hemos colocado an

### 2.2.2.42 SAVE

#### 2.2.2.42.1 Descripción

Guarda los cambios del libro especificado.



**2.2.2.42.2 Sintaxis:****Save****2.2.2.42.3 Argumentos**

No tiene.

**2.2.2.42.4 Ejemplo**

Código
<pre> Sub prueba_Save()     Dim miLibro As Excel.Workbook     Set miLibro = Application.Workbooks.Open("C:\temp\Ventas.xlsm")     miLibro.Worksheets(1).Cells(5, 1) = Now     miLibro.Save     MsgBox ("Pausa")     miLibro.Close     Set miLibro = Nothing End Sub </pre>

En este ejemplo, abrimos el libro Ventas.xlsm e insertamos la fecha/hora del sistema en la celda A5 y a continuación, salvamos el libro y lo cerramos después del mensaje de ‘Pausa’.

**2.2.2.43 SAVEAS****2.2.2.43.1 Descripción**

Guarda los cambios del libro en un archivo diferente.

**2.2.2.43.2 Sintaxis:**

**SaveAs (FileName, FileFormat, Password, WriteResPassword, ReadOnlyRecommended, CreateBackup, AddToMru, TextCodepage, TextVisualLayout, Local)**

**2.2.2.43.3 Argumentos**

Argumento	Comentario
<b>FileName</b>	Nombre del archivo
<b>FileFormat</b>	Formato con el que se guardará el archivo. Puede ser alguno de los formatos representados por la constante XlFileFormat.

<b>Password</b>	Contraseña a utilizar para la apertura.
<b>WriteResPassword</b>	Contraseña a utilizar contra escritura.
<b>ReadOnlyRecommended</b>	Si True, se sugiere abrir solo de lectura.
<b>CreateBackup</b>	Si True, crea copia de seguridad.
<b>AddToMru</b>	Si True, se incluye en la lista de archivos usados recientemente.
<b>TextCodepage</b>	No usado en versión en inglés de EE.UU de MS Excel.
<b>TextVisualLayout</b>	No usado en versión en inglés de EE.UU de MS Excel.
<b>Local</b>	Si True, guarda los archivos en el idioma de MS Excel sino, los guarda en el idioma de Visual Basic para Aplicaciones (VBA).

#### 2.2.2.43.4 Ejemplo

Código
<pre>Sub prueba_SaveAs ()     ActiveWorkbook.SaveAs Filename:="C:\Temp\jg.xlsm", _         ReadOnlyRecommended:=True, _         AddToMru:=True End Sub</pre>

Guarda el libro en curso en con el nombre CopiaLibro.xlsm, recomienda que solo se abra en modo lectura y lo añade a la lista de archivos recientes.

#### 2.2.2.44 SAVEASXMLDATA

##### 2.2.2.44.1 Descripción

Exporta a un archivo de datos XML los datos asignados a la asignación de esquema XML especificada.

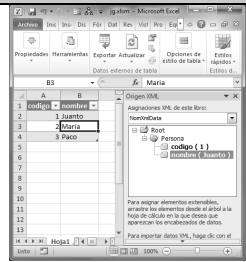

### 2.2.2.44.2 Sintaxis:

**SaveAsXMLData (nomFile, mapa)**

### 2.2.2.44.3 Argumentos

Argumento	Comentario
<b>nomFile</b>	Nombre de fichero del archivo que se guardará.
<b>mapa</b>	Asignación de esquema a aplicar.

### 2.2.2.44.4 Ejemplo

Código
<pre> Sub prueba_SaveAsXMLData()     Dim mapa As XmlMap     Set mapa = ActiveWorkbook.XmlMaps("NomXmlData")     If mapa.IsExportable Then         ActiveWorkbook.SaveAsXMLData "C:\Temp\res.xml", mapa     Else         MsgBox "Mapa " &amp; mapa.Name &amp; "no se puede esxportar."     End If End Sub </pre>
Resultado
  <pre> &lt;?xml version="1.0" encoding="UTF-8" standalone="true"?&gt; &lt;Root xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"&gt;   &lt;Persona&gt;     &lt;codigo&gt;1&lt;/codigo&gt;     &lt;nombre&gt;Juanito&lt;/nombre&gt;   &lt;/Persona&gt;   &lt;Persona&gt;     &lt;codigo&gt;2&lt;/codigo&gt;     &lt;nombre&gt;Maria&lt;/nombre&gt;   &lt;/Persona&gt;   &lt;Persona&gt;     &lt;codigo&gt;3&lt;/codigo&gt;     &lt;nombre&gt;Paco&lt;/nombre&gt;   &lt;/Persona&gt; &lt;/Root&gt; </pre>

En el ejemplo puede observarse un libro con una tabla (con campos código y nombre) que se esporta usando la asignación XML denominada NomXmlData hacia el archivo res.xml cuyo contenido e aprecia en la imagen de la derecha del todo.

## 2.2.2.45 SAVECOPYAS

### 2.2.2.45.1 Descripción

Permite guardar una copia del libro en un archivo pero no modifica el libro abierto en la memoria.

### 2.2.2.45.2 Sintaxis:

**SaveCopyAs (nomFile)**

### 2.2.2.45.3 Argumentos

Argumento	Comentario
<b>nomFile</b>	Nombre del fichero con el que se desea guardar la copia.

### 2.2.2.45.4 Ejemplo

Código
<pre>Sub prueba_SaveCopyAs()     ActiveWorkbook.SaveCopyAs ("C:\Temp\CopiaLibro.xlsm") End Sub</pre>

Este ejemplo guarda una copia del libro en curso con el nombre CopiaLibro.xlsm.

## 2.2.2.46 SENDFAXOVERINTERNET

### 2.2.2.46.1 Descripción

Permite enviar una hoja de cálculo como fax a los destinatarios especificados. Para utilizar el servicio de fax, tenemos que suscribirnos a un proveedor de servicios.

### 2.2.2.46.2 Sintaxis:

**SendFaxOverInternet (receptores,  
asunto, mostrarMensaje)**

### 2.2.2.46.3 Argumentos

Argumento	Comentario
<b>receptores</b>	Lista de destinatarios (números de fax o direcciones de correo electrónico) a enviar el fax separándolos por puntos y comas.
<b>asunto</b>	Descripción del asunto del documento enviado.
<b>mostrarMensaje</b>	Si True, muestra el mensaje de fax antes de enviarse.

### 2.2.2.46.4 Ejemplo

Código
<pre>Sub prueba_SenFaxOverInternet()     ActiveWorkbook.SendFaxOverInternet _         Recipients:="usuario@dominio.es", _         Subject:="Prueba envío Fax", _         showmessage:=True End Sub</pre>

## 2.2.2.47 SENDFORREVIEW

### 2.2.2.47.1 Descripción

Envía un libro en un mensaje de correo electrónico para su revisión a los destinatarios especificados.

### 2.2.2.47.2 Sintaxis:

**SendForReview(Recipients, Subject, ShowMessage, IncludeAttachment)**

### 2.2.2.47.3 Argumentos

Argumento	Comentario
<b>Recipients</b>	Lista de receptores a los que enviar el libro para su revisión. Si hay varios receptores, éstos se separarán mediante punto y coma (;).
<b>Subject</b>	Asunto que acompañará al mensaje.

<b>ShowMessage</b>	Si se indica True, aparecerá el cuadro de diálogo antes de enviar el mensaje para que podamos completar cualquiera de sus características (cuerpo, mas adjuntos, definir diversas opciones, etc)
<b>IncludeAttachment</b>	Indica si el mensaje ha de incluir la ubicación a un servidor o los datos adjuntos.

#### 2.2.2.47.4 Ejemplo

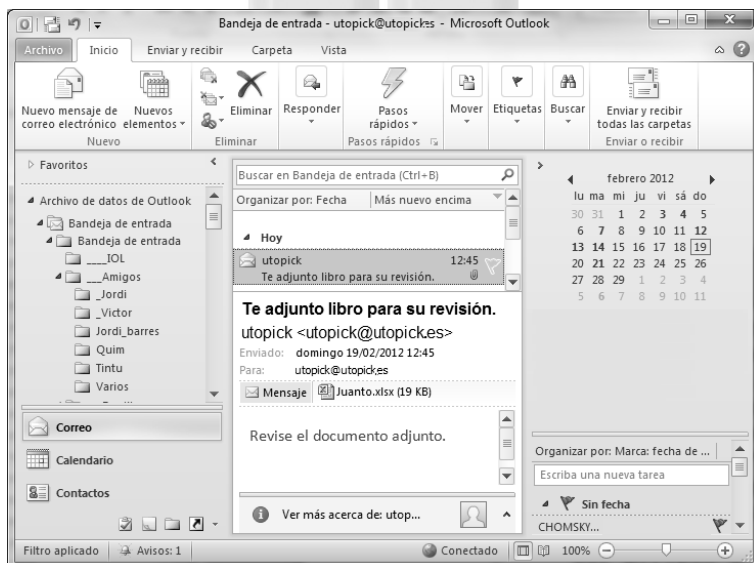
##### Código

```

Sub prueba_SendForReview()
    Dim miLibro As Excel.Workbook
    Set miLibro = Application.Workbooks.Open("C:\temp\Juanto.xlsx")
    miLibro.SendForReview _
        Recipients:="utopick@arrakis.es", _
        Subject:="Te adjunto libro para su revisión.", _
        showMessage:=False, _
        IncludeAttachment:=True
    miLibro.Close
    Set miLibro = Nothing
End Sub

```

##### Resultado



En este ejemplo, se puede comprobar como efectivamente se envía el libro para su revisión sin mostrar la pantalla para modificar el mensaje de forma que todo sucede silenciosamente.

## 2.2.2.48 SENDMAIL

### 2.2.2.48.1 Descripción

Permite enviar el libro indicado por correo electrónico.

### 2.2.2.48.2 Sintaxis:

**SendMail(Recipients, Subject, ReturnReceipt)**

### 2.2.2.48.3 Argumentos

Argumento	Comentario
<b>Recipients</b>	Destinatarios del correo.
<b>Subject</b>	Asunto del correo.
<b>ReturnReceipt</b>	Si True, solicita acuse de recibo. Por defecto su valor es False.

### 2.2.2.48.4 Ejemplo

Código
<pre>Sub prueba_sendmail()     Dim des(1 To 2)     des(1) = "utopick@utopick.es"     des(2) = "amigo@dominio.net"     ActiveWorkbook.SendMail Recipients:=des, _                              Subject:="Es una prueba", _                              ReturnReceipt:=True End Sub</pre>

En este ejemplo, el libro se envía por correo a 2 destinatarios y solicita acuse de recibo.

## 2.2.2.49 SETLINKONDATA

### 2.2.2.49.1 Descripción

Permite indicar cual será el procedimiento que debe ejecutarse cuando se actualice un vínculo DDE.

### 2.2.2.49.2 Sintaxis:

**SetLinkOnData (Name, Procedure)**

### 2.2.2.49.3 Argumentos

Argumento	Comentario
<b>Name</b>	Nombre del vínculo sobre el que se quiere establecer el procedimiento.
<b>Procedure</b>	Nombre del procedimiento que se ejecutará al actualizar el vínculo.

### 2.2.2.49.4 Ejemplo

Código
<pre>Sub prueba_SetLinkOnData()     Dim canal As Integer     canal = Application.DDEInitiate(app:="WinWord", topic:="C:\TEMP\JUANTO.DOC")     Dim miRango As Range     Set miRango = Range("A1:A3")     Application.DDEPoke canal, "\EndOfDoc", miRango     ActiveWorkbook.SetLinkOnData "C:\TEMP\JUANTO.DOC", "MiMacro"     Application.DDETerminate canal End Sub</pre>

## 2.2.2.50 SETPASSWORDENCRYPTIONOPTIONS

### 2.2.2.50.1 Descripción

Permite definir las opciones de cifrado de contraseñas para libros.



### 2.2.2.50.2 Sintaxis:

```
SetPasswordEncryptionOptions (PasswordEncryptionProvider,
                               PasswordEncryptionAlgorithm,
                               PasswordEncryptionKeyLength,
                               PasswordEncryptionFileProperties)
```

### 2.2.2.50.3 Argumentos

Argumento	Comentario
<b>PasswordEncryptionProvider</b>	<p>Proveedor utilizado al cifrar contraseñas para el libro. Por ejemplo:</p> <p>Microsoft Enhanced Cryptographic Provider v1.0.</p> <p>Microsoft Strong Cryptographic Provider.</p> <p>Microsoft RSA SChannel Cryptographic Provider.</p> <p>Microsoft Enhanced RSA and AES Cryptographic .Provider (Prototype).</p>
<b>PasswordEncryptionAlgorithm</b>	Algoritmo utilizado para cifrar contraseñas.
<b>PasswordEncryptionKeyLength</b>	Longitud a utilizar para la clave del algoritmo (p.e. 56, 128)
<b>PasswordEncryptionFileProperties</b>	Si True, se cifran las propiedades del archivo.

### 2.2.2.50.4 Ejemplo

Código
<pre>Sub prueba_SetPasswordOptions()     ActiveWorkbook.SetPasswordEncryptionOptions _         PasswordEncryptionProvider:="Microsoft Strong Cryptographic Provider", _         PasswordEncryptionAlgorithm:="RC4", _         PasswordEncryptionKeyLength:=128, _         PasswordEncryptionFileProperties:=True End Sub</pre>

### 2.2.2.51 TOGGLEFORMSDESIGN

#### 2.2.2.51.1 Descripción

El método `ToggleFormsDesign` se usa para cambiar Excel al modo de diseño cuando se usan controles de formularios.

Permite cambiar Excel a modo diseño cuando se utilizan controles de formularios. Equivale a hacer clic sobre el icono:



#### 2.2.2.51.2 Sintaxis:

`ToggleFormsDesign`

#### 2.2.2.51.3 Argumentos

No tiene.

#### 2.2.2.51.4 Ejemplo

Código
<pre>Sub prueba_ToggleFormsDesign()     ActiveWorkbook.ToggleFormsDesign End Sub</pre>

### 2.2.2.52 UNPROTECT

#### 2.2.2.52.1 Descripción

Elimina la protección de una hoja o de un libro. Este método no tiene efecto si la hoja o el libro no están protegidos.

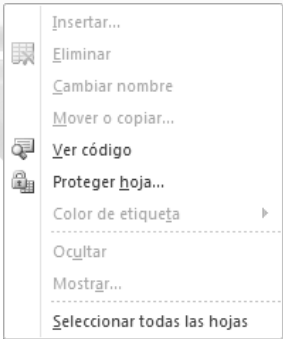
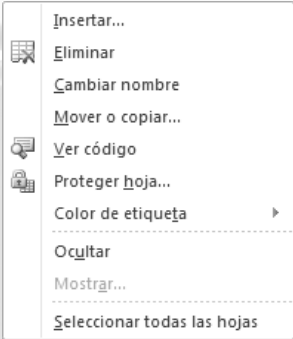
#### 2.2.2.52.2 Sintaxis:

`Unprotect (Password)`

### 2.2.2.52.3 Argumentos

Argumento	Comentario
Password	Palabra de paso a utilizar para desbloquear el libro.

### 2.2.2.52.4 Ejemplo

Código	
<pre>Sub prueba_Unprotect ()     ActiveWorkbook.Unprotect End Sub</pre>	
Resultado	
	

## 2.2.2.53 UNPROTECTSHARING

### 2.2.2.53.1 Descripción

Desactiva la protección de libro compartido y lo guarda.

Permite desproteger un libro compartido y guardarlo al mismo tiempo. Equivale a hacer clic sobre la opción **Desproteger libro compartido** de la pestaña **Revisar**.



### 2.2.2.53.2 Sintaxis:

**UnprotectSharing (SharingPassword)**

### 2.2.2.53.3 Argumentos

Argumento	Comentario
<b>SharingPassword</b>	Palabra de paso a utilizar para desproteger el libro.

### 2.2.2.53.4 Ejemplo

Código	
<pre>Sub prueba_UnprotectSharing()     Dim miLibro As Excel.Workbook     Set miLibro = Application.Workbooks.Open("c:\temp\Juanto.xlsx")     miLibro.UnprotectSharing     Application.Workbooks(2).Close SaveChanges:=True     Set miLibro = Nothing End Sub</pre>	
Resultado	
 Desproteger libro compartido	 Proteger libro compartido

Este ejemplo muestra como cambia la opción de **Desproteger libro compartido** a **Proteger libro compartido** tras la ejecución de este procedimiento.

## 2.2.2.54 UPDATEFROMFILE

### 2.2.2.54.1 Descripción

Permite actualizar la versión de un libro cargado en memoria que se ha abierto de sólo lectura si éste ha cambiado realmente.

### 2.2.2.54.2 Sintaxis:

**UpdateFromFile**

### 2.2.2.54.3 Argumentos

No tiene.

### 2.2.2.54.4 Ejemplo

Código
<pre>Sub prueba_UpdateFromFile()     ActiveWorkbook.UpdateFromFile End Sub</pre>

## 2.2.2.55 UPDATELINK

### 2.2.2.55.1 Descripción

Actualiza un vínculo (o vínculos) de Microsoft Excel, DDE u OLE.

### 2.2.2.55.2 Sintaxis:

**UpdateLink**(Name, Type)

### 2.2.2.55.3 Argumentos

Argumento	Comentario
<b>Name</b>	Vínculo que se desea actualizar.
<b>Type</b>	Tipo de vínculo que puede ser alguno de los siguientes:  <b>xlLinkTypeExcelLinks:</b> El vínculo es con una hoja de Excel. <b>xlLinkTypeOLELinks:</b> El vínculo es con un origen de datos OLE.

### 2.2.2.55.4 Ejemplo

Código
<pre>Sub prueba_UpdateLink()     Dim lLinks As Variant     Dim i As Integer     lLinks = ActiveWorkbook.LinkSources     For i = LBound(lLinks) To UBound(lLinks)         Debug.Print i &amp; " " &amp; lLinks(i)     Next i     ActiveWorkbook.UpdateLink Name:=lLinks(1) End Sub</pre>
Resultado
<pre>Inmediato 1 C:\Temp\JuanitoDDE.xlsx 2 C:\Cursos\Curso_Macros\Capitulos\08 - El modelo de objetos\Juanito.xlsx</pre>

En este ejemplo, se actualiza el vínculo asociado a **C:\Temp\JuanitoDDE.xlsx**.

## 2.2.2.56 WEBPAGEPREVIEW

### 2.2.2.56.1 Descripción

Permite realizar una previsualización del libro a través del navegador por defecto para ver como quedaría si el libro fuese guardado como una página Web.


### 2.2.2.56.2 Sintaxis:

**WebPagePreview**

### 2.2.2.56.3 Argumentos

No tiene.

### 2.2.2.56.4 Ejemplo

Código
<pre>Sub prueba_WebPagePreview()     ActiveWorkbook.WebPagePreview End Sub</pre>
Resultado


## 2.2.2.57 XMLIMPORT

### 2.2.2.57.1 Descripción

Importa un archivo de datos XML al libro actual.


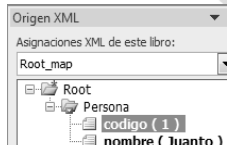
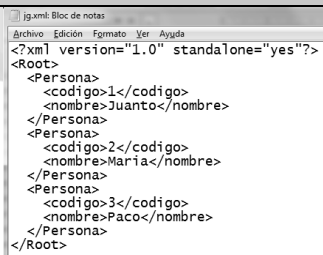
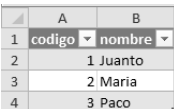
### 2.2.2.57.2 Sintaxis:

**XmlImport(Url, ImportMap, Overwrite, Destination)**

### 2.2.2.57.3 Argumentos

Argumento	Comentario
<b>Url</b>	Dirección para acceder al archivo XML.
<b>ImportMap</b>	Esquema a aplicar cuando se importa el archivo.
<b>Overwrite</b>	Si True sobrescribe los datos.
<b>Destination</b>	Rango en el que se creará la lista.

### 2.2.2.57.4 Ejemplo

Código			
<pre> Sub prueba_XmlImport()     Dim path As String     Dim r As Variant     Dim xmp As XmlMap     path = "C:\Temp\jg.XML"     Set xmp = Application.Workbooks(1).XmlMaps(1)     r = ActiveWorkbook.XmlImport(path, xmp) End Sub </pre>			
Resultado			
			

En este ejemplo, puede apreciarse como se importan los datos asociados a la asignación XML cargada en primer lugar en el libro tras ejecutar el procedimiento indicado.

### 2.2.2.58 XMLIMPORTXML

#### 2.2.2.58.1 Descripción

Permite importar una secuencia de datos XML previamente cargada en memoria.

#### 2.2.2.58.2 Sintaxis:

`XmlImportXml (Data, ImportMap, Overwrite, Destination)`

#### 2.2.2.58.3 Argumentos

Argumento	Comentario
<b>Data</b>	Datos a importar.
<b>ImportMap</b>	Esquema a aplicar en la importación.
<b>Overwrite</b>	Si True sobrescribe los datos.
<b>Destination</b>	Rango en el que se creará la lista.

### 2.2.3 Eventos

#### 2.2.3.1 ACTIVATE

Evento producido al activar un libro, hoja o gráfico incrustado.

#### 2.2.3.2 ADDININSTALL

Producido cuando se instala un libro como complemento.

#### 2.2.3.3 ADDINUNINSTALL

Producido cuando se desinstala un libro como complemento.

#### 2.2.3.4 AFTERSAVE

Producido al guardar un libro.

#### 2.2.3.5 AFTERXMLEXPORT

Producido al guardar o exportar datos XML desde un libro.



### **2.2.3.6 AFTERXMLIMPORT**

Producido al importar nuevos datos XML o al actualizar una conexión de datos XML.

### **2.2.3.7 BEFORECLOSE**

Se produce antes de cerrar el libro.

### **2.2.3.8 BEFOREPRINT**

Producido antes de imprimir el libro.

### **2.2.3.9 BEFORESAVE**

Producido antes de guardar el libro.

### **2.2.3.10 BEFOREXMLEXPORT**

Se produce antes de que Microsoft Excel guarde o exporte datos XML desde el libro especificado.

### **2.2.3.11 BEFOREXMLIMPORT**

Producido antes de importar nuevos datos XML o de actualizar una conexión de datos XML existente.

### **2.2.3.12 DEACTIVATE**

Producido al desactivar un libro, una hoja o un gráfico.

### **2.2.3.13 NEWCHART**

Producido al crear un nuevo gráfico.

### **2.2.3.14 NEWSHEET**

Producido al crear una nueva hoja.

### **2.2.3.15 OPEN**

Producido al abrir el libro

### **2.2.3.16 PIVOTTABLECLOSECONNECTION**

Se produce después de cerrar la conexión de un informe de tabla dinámica.

### **2.2.3.17 PIVOTTABLEOPENCONNECTION**

Producido al abrir la conexión de un informe de tabla dinámica.

### **2.2.3.18 ROWSETCOMPLETE**

Producido al haber recorrido el conjunto de registros o al invocar la acción de conjunto de filas en una tabla dinámica OLAP.

### **2.2.3.19 SHEETACTIVATE**

Producido al activar una hoja.

### **2.2.3.20 SHEETBEFOREDOUBLECLICK**

Producido al hacer doble clic en una hoja justo antes de que se produzca la acción predeterminada asociada al doble clic.

### **2.2.3.21 SHEETBEFORERIGHTCLICK**

Producido al hacer clic en una hoja con el botón secundario del mouse (ratón) justo antes de que se produzca la acción predeterminada asociada al clic de dicho botón secundario.

### **2.2.3.22 SHEETCALCULATE**

Producido después de calcular cualquier hoja del libro.

### **2.2.3.23 SHEETCHANGE**

Producido al cambiar el contenido de una celda.

### **2.2.3.24 SHEETDEACTIVATE**

Producido al desactivar una hoja.

### **2.2.3.25 SHEETFOLLOWHYPERLINK**

Producido al hacer clic sobre un hipervínculo.

### **2.2.3.26 SHEETPIVOTTABLEAFTERVALUECHANGE**

Producido al recalcular o editar una celda que contiene fórmulas dentro de una tabla dinámica.

### **2.2.3.27 SHEETPIVOTTABLEBEFOREALLOCATECHANGES**

Producido antes de que se apliquen los cambios en una tabla dinámica.

**2.2.3.28 SHEETPIVOTTABLEBEFORECOMMITCHANGES**

Producido antes de que se confirmen los cambios en una tabla dinámica cuyo origen de datos sea OLAP.

**2.2.3.29 SHEETPIVOTTABLEBEFOREDISCARDCHANGES**

Producido antes de descartar los cambios realizados en una tabla dinámica.

**2.2.3.30 SHEETPIVOTTABLECHANGESYNC**

Producido después de realizar los cambios en una tabla dinámica.

**2.2.3.31 SHEETPIVOTTABLEUPDATE**

Producido después de que se actualice la hoja del informe de tabla dinámica.

**2.2.3.32 SHEETSELECTIONCHANGE**

Producido al cambiar la selección en una hoja excepto en una hoja de gráfico.

**2.2.3.33 SYNC**

Producido al sincronizar la copia local de un libro que forma parte de un área de documentos con la copia almacenada en el servidor.

**2.2.3.34 WINDOWACTIVATE**

Producido al activar una ventana del libro.

**2.2.3.35 WINDOWDEACTIVATE**

Producido cuando se desactiva una ventana del libro.

**2.2.3.36 WINDOWRESIZE**

Producido al cambiar el tamaño de una ventana del libro.

Ra-Ma®

## **2.3 EL OBJETO HOJA (WORKSHEET).**

### **2.3.1 Propiedades**

#### **2.3.1.1 APPLICATION**

Proporciona un objeto que representa al creador del objeto indicado.

#### **2.3.1.2 AUTOFILTER**

Si se ha activado el filtrado, esta propiedad proporciona un objeto de tipo AutoFilter.

#### **2.3.1.3 AUTOFILTERMODE**

Devuelve True si están visibles las flechas desplegables en la hoja.

#### **2.3.1.4 CELLS**

Proporciona un objeto de tipo Range que representa a todas las celdas de la hoja indicada.

#### **2.3.1.5 CIRCULARREFERENCE**

En el caso de que haya referencias circulares, devuelve el rango que contiene la primera referencia circular.

#### **2.3.1.6 CODENAME**

Proporciona el nombre de la hoja indicada.

#### **2.3.1.7 COLUMNS**

Proporciona un objeto de tipo Range que representa a todas las columnas de la hoja indicada.

#### **2.3.1.8 COMMENTS**

Proporciona un objeto de tipo Comments con todos los comentarios de la hoja indicada.

#### **2.3.1.9 CONSOLIDATIONFUNCTION**

Indica el código de la función que se está utilizando en la consolidación actual y puede ser alguna de las siguientes:

**xlAverage** (-4106) Promedio.

**xlCount** (-4112) Contar.

**xlCountNums** (-4113) Contar sólo los valores numéricos.

**xlMax** (-4136) Máximo.

**xlMin** (-4139) Mínimo.

**xlProduct** (-4149) Multiplicar.

**xlStDev** (-4155) Desviación estándar, basada en una muestra.

**xlStDevP** (-4156) Desviación estándar, basada en toda la población.

**xlSum** (-4157) Suma.

**xlUnknown** (1000) No se especifica ninguna función de subtotal.

**xlVar** (-4164) Variación, basada en una muestra.

(-4165) Variación, basada en toda la población.

### 2.3.1.10 CONSOLIDATIONOPTIONS

Proporciona información en forma de matriz sobre las opciones de consolidación. Dichas opciones son las siguientes:

- 1 Usar rótulos en la fila superior.
- 2 Usar rótulos en la columna izquierda.
- 3 Crear vínculos con los datos de origen.

### 2.3.1.11 CONSOLIDATIONSOURCES

Proporciona una matriz con los nombres de los rangos existentes en la consolidación actual. Si no hay ninguno devuelve Empty.

### 2.3.1.12 CREATOR

Proporciona un número que indica cual fue la aplicación que creó el objeto.

### 2.3.1.13 CUSTOMPROPERTIES

Proporciona objeto con la información de identificador asociada a una hoja de cálculo.

### 2.3.1.14 DISPLAYPAGEBREAKS

Devuelve True si se visualizan los saltos de página.

### 2.3.1.15 DISPLAYRIGHTTOLEFT

Indica si la hoja de cálculo especificada se muestra de derecha a izquierda en lugar de mostrarse de izquierda a derecha. Si definimos esta propiedad como True, observaremos que la primer columna (A) se sitúa a la derecha de la pantalla y la disposición del resto (B, C, etc ...) se irá realizando hacia la izquierda.

### 2.3.1.16 ENABLEAUTOFILTER

Define o consulta si están habilitadas las flechas de filtro automático al estar activada la protección de solo interfaz de usuario.

### 2.3.1.17 ENABLECALCULATION

Define o consulta si MS Excel ha de actualizar la hoja automáticamente o no. Equivale a seleccionar **Automático** para **Opciones de cálculo** situado en la pestaña **Fórmulas**.

### 2.3.1.18 ENABLEFORMATCONDITIONSCALCULATION

Define o consulta si se utilizarán los formatos condicionales automáticamente.

### 2.3.1.19 ENABLEOUTLINING

Permite definir o consultar si en una hoja que esté protegida es posible expandir o colapsar las agrupaciones de un esquema.



	A	B	C
1	Nombre	Provincia	Importe
5	<b>Total Juan</b>		1125
6	Maria	Barcelona	20
7	Maria	Madrid	200
8	Maria	Valencia	2000
9	<b>Total Maria</b>		2220
13	<b>Total Paco</b>		3350
14	<b>Total general</b>		6695

### 2.3.1.20 ENABLEPIVOTTABLE

Define o consulta si las acciones y controles de una tabla dinámica deben estar habilitados cuando se activa la protección de solo interfaz de usuario. True permite habilitar dichos controles.



	A
1	
2	
3	Etiquetas de fila ▼
4	Barcelona
5	Madrid
6	Juan
7	100
8	Maria
9	200
10	Paco
11	300
12	Valencia
13	(en blanco)
14	(en blanco)
15	(en blanco)
16	Total general

### 2.3.1.21 ENABLESELECTION

Devuelve o establece qué se puede seleccionar en la hoja. XlEnableSelection de lectura y escritura.

Define o consulta qué es lo que se puede seleccionar en la hoja. Puede definirse alguno de los siguientes valores:

**xlNoRestrictions** (0) Se puede seleccionar cualquier celda.

**xlNoSelection** (-4142) No se puede seleccionar nada.

**xlUnlockedCells** (1) Sólo se pueden seleccionar las celdas que no estén bloqueadas.

### 2.3.1.22 FILTERMODE

Si la hoja de cálculo tiene asignado algún filtro y dicho filtro se está aplicando, devuelve True.

### 2.3.1.23 HPAGEBREAKS

Proporciona una colección de tipo HPageBreaks mediante la cual podemos obtener información sobre los saltos de página horizontales existentes en la hoja.

### **2.3.1.24 HYPERLINKS**

Proporciona una colección de Hyperlinks existentes en la hoja de cálculo.

### **2.3.1.25 INDEX**

Devuelve el número de hoja que posee la hoja indicada en el libro al que pertenece.

### **2.3.1.26 LISTOBJECTS**

Proporciona una colección objetos de tipo ListObject existentes en la hoja.

### **2.3.1.27 MAILENVELOPE**

Permite tratar el encabezado de correo electrónico para un documento.

### **2.3.1.28 NAME**

Define o consulta el nombre de la hoja indicada.

### **2.3.1.29 NAMES**

Proporciona una colección de tipo Names que representa todos los nombres específicos de hoja indicada.

### **2.3.1.30 NEXT**

Proporciona un objeto que representa la siguiente hoja del libro en el que se halla la hoja indicada.

### **2.3.1.31 OUTLINE**

Proporciona un objeto de tipo Outline que representa el esquema de la hoja indicada.

### **2.3.1.32 PAGESETUP**

Proporciona un objeto de tipo PageSetup con todos los valores de configuración de página del objeto (margenes izquierdo e inferior, tamaño de papel, etc.).

### **2.3.1.33 PARENT**

Proporciona el objeto primario de la hoja.



### **2.3.1.34 PREVIOUS**

Proporciona un objeto que representa hoja anterior del libro en el que se halla la hoja indicada.

### **2.3.1.35 PRINTEDCOMMENTPAGES**

Indidca el número de páginas de comentarios que serán impresas para la hoja de cálculo actual.

### **2.3.1.36 PROTECTCONTENTS**

Indica si la hoja está protegida.

### **2.3.1.37 PROTECTDRAWINGOBJECTS**

Indica si las formas están protegidas.

### **2.3.1.38 PROTECTION**

Proporciona un objeto de tipo Proteccion mediante el cual puede consultar o definir las siguientes opciones de protección:

AllowDeletingColumns

AllowDeletingRows

AllowFiltering

AllowFormattingCells

AllowFormattingColumns

AllowFormattingRows

AllowInsertingColumns

AllowInsertingHyperlinks

AllowInsertingRows

AllowSorting

AllowUsingPivotTables

### **2.3.1.39 PROTECTIONMODE**

Si True, indica que está activada la protección de solo interfaz de usuario.

### **2.3.1.40 PROTECTSCENARIOS**

Si True, indica que están protegidos los escenarios de la hoja.

### **2.3.1.41 QUERYTABLES**

Proporciona una colección de tipo QueryTables con todas las tablas de consulta de la hoja.

### **2.3.1.42 RANGE**

Proporciona un objeto que representa un rango de la hoja.

### **2.3.1.43 ROWS**

Proporciona un objeto de tipo Range que representa a todas las filas de la hoja indicada.

### **2.3.1.44 SCROLLAREA**

Define o consulta el rango en el que podemos desplazarnos dentro de la hoja. Si definimos un rango en esta propiedad, observaremos que el usuario sólo puede desplazarse dentro del rango indicado.

### **2.3.1.45 SHAPES**

Proporciona una colección que representa las distintas formas existentes en la hoja indicada.

### **2.3.1.46 SORT**

Permite obtener los valores ordenados de la hoja.

### **2.3.1.47 STANDARDHEIGHT**

Proporciona el alto predeterminado de las filas de la hoja.

### **2.3.1.48 STANDARDWIDTH**

Define o consulta el ancho predeterminado de las columnas de la hoja.

### **2.3.1.49 TAB**

Proporciona un objeto de tipo Tab de la hoja indicada.

### 2.3.1.50 TRANSITIONEXPEVAL

Define o consulta si MS Excel utiliza las reglas de evaluación de expresiones de Lotus 1-2-3 en al hoja.

### 2.3.1.51 TRANSITIONFORMENTRY

Define o consulta si MS Excel utiliza las reglas de inserción de fórmulas de Lotus 1-2-3 en al hoja.

### 2.3.1.52 TYPE

Indica el tipo de hoja de cálculo que puede ser alguno de los siguientes:

**xlChart** (-4109) Gráfico

**xlDialogSheet** (-4116) Hoja de diálogo

**xlExcel4IntlMacroSheet** (4) Hoja de macros internacional de Excel versión 4

**xlExcel4MacroSheet** (3) Hoja de macros de Excel versión 4

**xlWorksheet** (-4167) Hoja de cálculo

### 2.3.1.53 USEDRange

Proporciona un objeto de tipo Range que representa el rango usado en la hoja indicada.

### 2.3.1.54 VISIBLE

Devuelve o establece un valor **xlSheetVisibility** que determina si el objeto está visible.

Define o consulta si una hoja es visible o no. Puede tener alguno de los siguientes valores:

**xlSheetHidden** (0) Oculta la hoja. Sin embargo, esta hoja puede volver a mostrarse por el usuario accediendo la ficha Inicio-> Grupo de controles Celdas -> Formato-> Visibilidad -> Ocultar y mostrar -> Mostrar Hoja.

**xlSheetVeryHidden** (2) Oculta el objeto de forma que solo puede volver a visualizarlo estableciendo esta propiedad en True, es decir, el usuario no puede mostrar la hoja ya que necesariamente se ha de hacer desde el entorno de

programación o mediante una macro preparada para cambiar el valor de esta propiedad.

**xlSheetVisible** (-1) Visualiza la hoja.

### 2.3.1.55 VPAGEBREAKS

Proporciona una colección de tipo VPageBreaks mediante la cual podemos obtener información sobre los saltos de página verticales existentes en la hoja.

## 2.3.2 Métodos

### 2.3.2.1 ACTIVATE

#### 2.3.2.1.1 Descripción

Activa la hoja actual.

#### 2.3.2.1.2 Sintaxis:

Activate

#### 2.3.2.1.3 Argumentos

No tiene.

#### 2.3.2.1.4 Ejemplo

Código
<pre>Sub prueba_Activate()     ActiveWorkbook.Worksheets(1).Activate     MsgBox ("Pausa")     ActiveWorkbook.Worksheets("Hoja5").Activate     MsgBox ("Pausa") End Sub</pre>

En este ejemplo, se activa la primera hoja de la colección de hojas del libro activo, se hace una pausa y a continuación se activa la hoja denominada Hoja5.

### 2.3.2.2 CALCULATE

#### 2.3.2.2.1 Descripción

Permite calcular la hoja indicada.

**2.3.2.2.2 Sintaxis:****Calculate****2.3.2.2.3 Argumentos**

No tiene.

**2.3.2.2.4 Ejemplo**

Código
<pre>Sub prueba_Calculate()     ActiveSheet.Calculate End Sub</pre>

Este ejemplo permite recalcular la hoja activa.

**2.3.2.3 CHARTOBJECTS****2.3.2.3.1 Descripción**

Proporciona una colección de gráficos que se hallan integrados en la hoja.

**2.3.2.3.2 Sintaxis:****ChartObjects (Index)****2.3.2.3.3 Argumentos**

Argumento	Comentario
<b>Index</b>	Número del gráfico o nombre del mismo a tratar. Puede indicarse una matriz para tratar varios gráficos.

**2.3.2.3.4 Ejemplo**

Código
<pre>Sub prueba_ChartObjects()     Dim c As ChartObject     For Each c In ActiveSheet.ChartObjects         With c             Debug.Print .Name &amp; " " &amp; .Chart.ChartType         End With     Next End Sub</pre>



### 2.3.2.4 CHECKSPELLING

#### 2.3.2.4.1 Descripción

Comprueba la ortografía de un objeto.


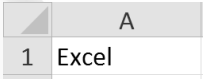
#### 2.3.2.4.2 Sintaxis:

`CheckSpelling(CustomDictionary, IgnoreUppercase, AlwaysSuggest, SpellLang)`

#### 2.3.2.4.3 Argumentos

Argumento	Comentario
<b>CustomDictionary</b>	Nombre del archivo del diccionario a utilizar si la palabra no se halla en el diccionario principal.
<b>IgnoreUppercase</b>	Si se indica True, no se analizan las palabras escritas totalmente en mayúsculas. Si se indica False si que se analizan. Por defecto se usa la configuración actual.
<b>AlwaysSuggest</b>	Si True, al encontrar un error se muestra una lista de sugerencias.
<b>SpellLang</b>	Idioma del diccionario utilizado. (Ver MsoLanguageID Enumeration para obtener una lista de los idiomas pibles).

### 2.3.2.4.4 Ejemplo

Código	
<pre>Sub prueba_CheckSpelling()     ActiveSheet.CheckSpelling End Sub</pre>	
Resultado	
	
	

En este ejemplo puede observarse como al encontrar un error ortográfico, se muestra el cuadro de diálogo para sugerir palabras a utilizar y, una vez escogida la palabra adecuada, el error queda corregido.

### 2.3.2.5 CIRCLEINVALID

#### 2.3.2.5.1 Descripción

Permite rodear con un círculo los datos no son válidos. Equivale a ejecutar en la pestaña de **Datos** y grupo de **Herramientas de datos** la siguiente opción:



Rodear con un círculo datos no válidos

#### 2.3.2.5.2 Sintaxis:

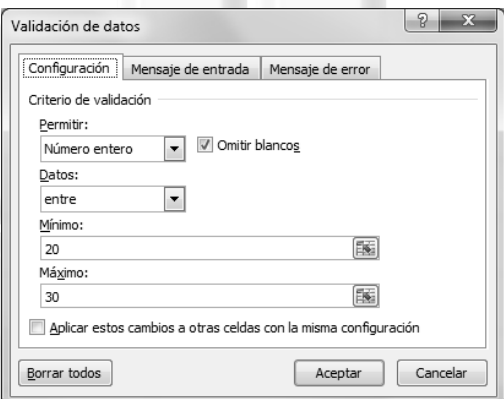
**CircleInvalid**

#### 2.3.2.5.3 Argumentos

No tiene.

Ra-Ma<sup>®</sup>

### 2.3.2.5.4 Ejemplo

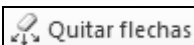
Código																						
<pre>Sub prueba_CircleInvalid()   Dim h As Worksheet   Set h = ActiveWorkbook.ActiveSheet   h.CircleInvalid End Sub</pre>																						
Resultado																						
<table border="1" style="margin: auto; border-collapse: collapse;"> <thead> <tr style="background-color: #cccccc;"> <th></th> <th>A</th> </tr> </thead> <tbody> <tr><td>1</td><td>10</td></tr> <tr><td>2</td><td>20</td></tr> <tr><td>3</td><td>30</td></tr> <tr><td>4</td><td>15</td></tr> </tbody> </table>		A	1	10	2	20	3	30	4	15		<table border="1" style="margin: auto; border-collapse: collapse;"> <thead> <tr style="background-color: #cccccc;"> <th></th> <th>A</th> </tr> </thead> <tbody> <tr><td>1</td><td>10</td></tr> <tr><td>2</td><td>20</td></tr> <tr><td>3</td><td>30</td></tr> <tr><td>4</td><td>15</td></tr> </tbody> </table>		A	1	10	2	20	3	30	4	15
	A																					
1	10																					
2	20																					
3	30																					
4	15																					
	A																					
1	10																					
2	20																					
3	30																					
4	15																					

Este ejemplo muestra como tras aplicar una validación de datos a un determinado rango y ejecutar el procedimiento, los valores que no están comprendidos entre 20 y 30 son marcados con un círculo.

### 2.3.2.6 CLEARARROWS

#### 2.3.2.6.1 Descripción

Elimina las flechas de rastreo que haya podido dejar alguna acción de auditoría. Equivale a la opción situada en **Auditoria de fórmulas** de la pestaña de **Fórmulas**:



#### 2.3.2.6.2 Sintaxis:

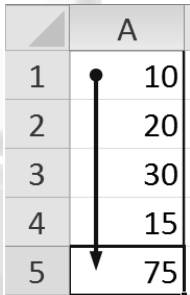
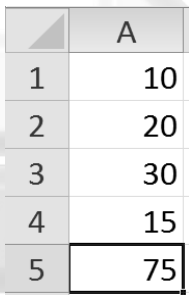
**ClearArrows**



### 2.3.2.6.3 Argumentos

No tiene.

### 2.3.2.6.4 Ejemplo

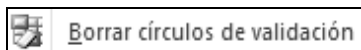
Código	
<pre>Sub prueba_ClearArrows()     ActiveWorkbook.ActiveSheet.ClearArrows End Sub</pre>	
Resultado	
	

Podemos observar como después de ejecutar el procedimiento, desaparece la flecha que había dejado la opción de **Rastrear precedentes** que mostraba de donde se obtenía el resultado de A5.

### 2.3.2.7 CLEARCIRCLES

#### 2.3.2.7.1 Descripción

Permite eliminar los círculos de los datos no válidos. Equivale a ejecutar en la pestaña de **Datos** y grupo de **Herramientas de datos** la siguiente opción:



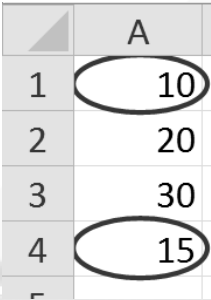
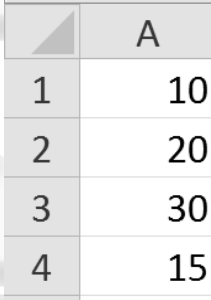
#### 2.3.2.7.2 Sintaxis:

ClearCircles

#### 2.3.2.7.3 Argumentos

No tiene.

### 2.3.2.7.4 Ejemplo

Código	
<pre>Sub prueba_ClearCircles()     ActiveWorkbook.ActiveSheet.ClearCircles End Sub</pre>	
Resultado	
	

### 2.3.2.8 COPY

#### 2.3.2.8.1 Descripción

Permite copiar la hoja en otra parte del libro.

#### 2.3.2.8.2 Sintaxis:

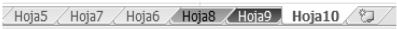

**Copy(Before, After)**

#### 2.3.2.8.3 Argumentos

Argumento	Comentario
<b>Before</b>	La hoja se copiará antes de la hoja indicada en Before.
<b>After</b>	La hoja se copiará después de la hoja indicada en Before.

Solo se puede indicar Before o After pero no ambas.

### 2.3.2.8.4 Ejemplo

Código	
<pre>Sub prueba_Copy()     ActiveWorkbook.Worksheets("Hoja5").Copy _         After:=ActiveWorkbook.Worksheets("Hoja10") End Sub</pre>	
Resultado	
	

### 2.3.2.9 DELETE

#### 2.3.2.9.1 Descripción

Permite eliminar la hoja indicada.

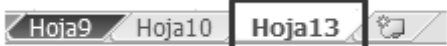

#### 2.3.2.9.2 Sintaxis:

**Delete**

#### 2.3.2.9.3 Argumentos

No tiene.

#### 2.3.2.9.4 Ejemplo

Código	
<pre>Sub prueba_Delete()     ActiveWorkbook.Worksheets("Hoja13").Delete End Sub</pre>	
Resultado	
	

### 2.3.2.10 EVALUATE

#### 2.3.2.10.1 Descripción

Permite evaluar una expresión.

#### 2.3.2.10.2 Sintaxis:

**Evaluate (Name)**

#### 2.3.2.10.3 Argumentos

Argumento	Comentario
<b>Name</b>	Expresión a evaluar.

#### 2.3.2.10.4 Ejemplo

Código			
<pre>Sub prueba_Evaluate()     Debug.Print "Raíz cúbica 27^(1/3): " &amp;     ActiveWorkbook.ActiveSheet.Evaluate("27^(1/3)")     Dim s As String     s = "sum(" &amp; "A1:A4" &amp; ")"     Debug.Print "Suma ( " &amp; s &amp; " ) " &amp;     ActiveWorkbook.ActiveSheet.Evaluate(s) End Sub</pre>			
Resultado			
<table><tr><th>Inmediato</th></tr><tr><td>Raíz cúbica 27^(1/3): 3</td></tr><tr><td>Suma ( sum(A1:A4) ) 75</td></tr></table>	Inmediato	Raíz cúbica 27^(1/3): 3	Suma ( sum(A1:A4) ) 75
Inmediato			
Raíz cúbica 27^(1/3): 3			
Suma ( sum(A1:A4) ) 75			

### 2.3.2.11 EXPORTASFIXEDFORMAT

#### 2.3.2.11.1 Descripción

Permite exportar la hoja indicada con el formato especificado.

### 2.3.2.11.2 Sintaxis:

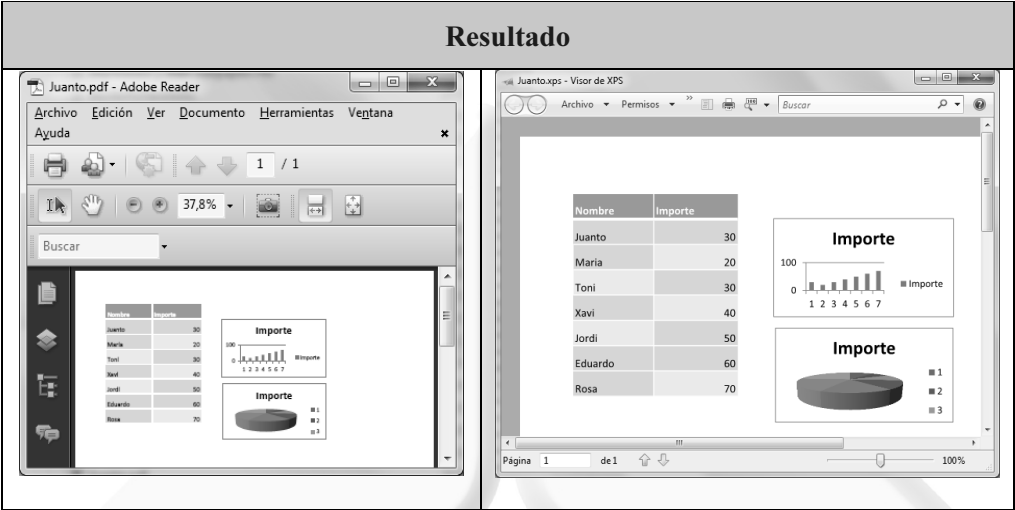
**ExportAsFixedFormat (Type, Filename, Quality, IncludeDocProperties, IgnorePrintAreas, From, To, OpenAfterPublish)**

### 2.3.2.11.3 Argumentos

Argumento	Comentario
<b>Type</b>	Tipo a utilizar para la exportación. Puede ser: <b>xlTypePDF</b> o <b>xlTypeXPS</b> .
<b>Filename</b>	Nombre del archivo con el que se creará la exportación.
<b>Quality</b>	Calidad con la que se creará el documento. Puede ser: <b>xlQualityStandard</b> o <b>xlQualityMinimum</b> .
<b>IncludeDocProperties</b>	Si se indica True, se incluyen las propiedades en los documentos.
<b>IgnorePrintAreas</b>	Si se indica True se omiten las áreas de impresión.
<b>From</b>	Indica el número de la primera página a publicar.
<b>To</b>	Indica el número de la última página a publicar.
<b>OpenAfterPublish</b>	Si se indica True, el documento resultante se abre una vez exportado.

### 2.3.2.11.4 Ejemplo

Código
<pre> Sub prueba_ExportAsFixedFormat ()     ' Se exporta a pdf     ActiveWorkbook.ActiveSheet.ExportAsFixedFormat Type:=xlTypePDF, -     Filename:="c:\temp\Juanto.pdf", _     Quality:=xlQualityStandard, _         OpenAfterPublish:=True     ' Se exporta a xps y abre documento.     ActiveWorkbook.ActiveSheet.ExportAsFixedFormat Type:=xlTypeXPS, -     Filename:="c:\temp\Juanto.xps", _     Quality:=xlQualityStandard, _         OpenAfterPublish:=True End Sub </pre>



### 2.3.2.12 MOVE

#### 2.3.2.12.1 Descripción

Permite desplazar la hoja a otra posición dentro del libro.

#### 2.3.2.12.2 Sintaxis:

**Move (Before, After)**

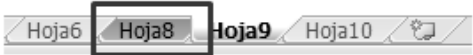
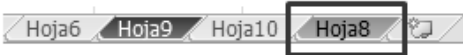
#### 2.3.2.12.3 Argumentos

Argumento	Comentario
Before	La hoja se moverá antes de la hoja indicada en Before.
After	La hoja se moverá después de la hoja indicada en Before.

Solo se puede indicar Before o After pero no ambas.

#### 2.3.2.12.4 Ejemplo

Código
<pre>Sub prueba_Move()     ActiveWorkbook.Worksheets("Hoja8").Move _         After:=ActiveWorkbook.Worksheets("Hoja10") End Sub</pre>

Resultado	
	

En este ejemplo, se ha movido la Hoja8 después de la Hoja10.

### 2.3.2.13 OLEOBJECTS

#### 2.3.2.13.1 Descripción

Proporciona una colección objetos OLE (una colección OLEObjects) que se hallan en la hoja.

#### 2.3.2.13.2 Sintaxis:

OLEObjects(Index)

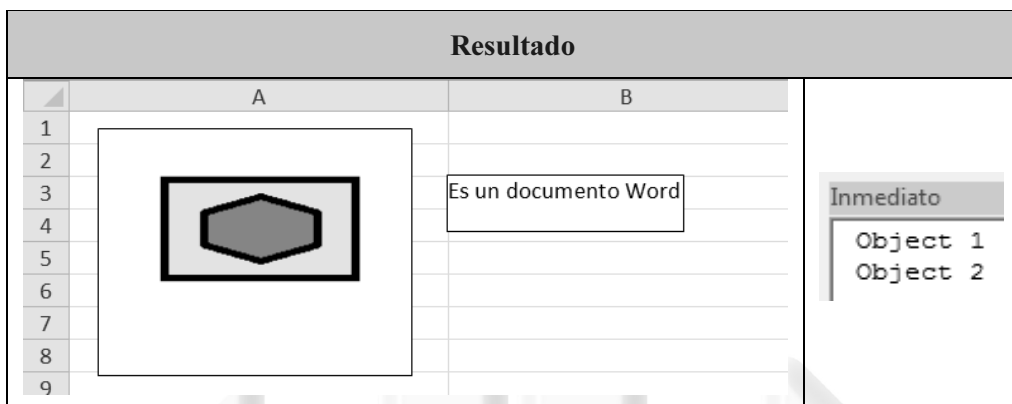
#### 2.3.2.13.3 Argumentos

Argumento	Comentario
<b>Index</b>	Indice del objeto a tratar (número o nombre).

#### 2.3.2.13.4 Ejemplo

Código
<pre> Sub prueba_OLEObjects()     Dim c As OLEObject     For Each c In ActiveSheet.OLEObjects         With c             Debug.Print .Name         End With     Next End Sub </pre>

Ra-Ma®



En este ejemplo, podemos ver como obtenemos un listado de los objetos existentes en la hoja activa que en este caso se trata de un dibujo de Paint y de un documento Word.

### 2.3.2.14 PASTE

#### 2.3.2.14.1 Descripción

Pega el contenido del Portapapeles en la hoja.

Permite copiar el contenido del portatapeles en la hoja indicada.

#### 2.3.2.14.2 Sintaxis:

**Paste (Destination, Link)**

#### 2.3.2.14.3 Argumentos

Argumento	Comentario
<b>Destination</b>	Rango sobre el que se ha de pegar el contenido del portapapeles.
<b>Link</b>	Establece un vínculo con el origen de los datos pegados si se indica True.

#### 2.3.2.14.4 Ejemplo

Código
<pre> Sub prueba_Paste()     ActiveWorkbook.ActiveSheet.Range("A1:B4").Copy     ActiveWorkbook.ActiveSheet.Paste Destination:=Range("D1:D4") End Sub </pre>



Resultado										
	A	B	C	D		A	B	C	D	E
1	Vendedor	Importe			1	Vendedor	Importe		Vendedor	Importe
2	A	10			2	A	10		A	10
3	B	20			3	B	20		B	20
4	C	30			4	C	30		C	30

Podemos observar como tras copiar el contenido del rango A1:B4, éste es copiado sobre D1:D4.

### 2.3.2.15 PASTESPECIAL

#### 2.3.2.15.1 Descripción

Permite copiar el contenido del portatapeles con un determinado formato.

#### 2.3.2.15.2 Sintaxis:

`PasteSpecial(Format, Link, DisplayAsIcon, IconFileName, IconIndex, IconLabel, NoHTMLFormatting)`

#### 2.3.2.15.3 Argumentos

Argumento	Comentario
<b>Format</b>	<p>Formato a utilizar. Puede ser alguno de los siguientes:</p> <ul style="list-style-type: none"> <li>1 Todo</li> <li>2 Fórmulas</li> <li>3 Valores</li> <li>4 Formatos</li> <li>5 Comentarios</li> <li>6 Validaciones</li> <li>7 Todo excepto bordes</li> <li>8 Ancho columnas</li> <li>11 Formatos de números y fórmulas</li> <li>12 Formatos de números y valores</li> <li>13 Todos los formatos condicionales de combinación</li> </ul>

<b>Link</b>	True indica que se establece un vínculo con el origen de los datos pegados.
<b>DisplayAsIcon</b>	Si True, los datos pegados se muestran como un icono.
<b>IconFileName</b>	Nombre del archivo que contiene el icono a utilizar si DisplayAsIcon es True.
<b>IconIndex</b>	Índice del icono para el archivo de iconos.
<b>IconLabel</b>	Rótulo del icono.
<b>NoHTMLFormatting</b>	Si True, se quita el formato, los hipervínculos y las imágenes de HTML.

### 2.3.2.15.4 Ejemplo

Código
<pre> Sub prueba_PasteSpecial()     Dim h As Worksheet     Set h = ActiveWorkbook.ActiveSheet     ActiveWorkbook.ActiveSheet.Range("B2").Select     h.PasteSpecial (1) ' Todo     ActiveWorkbook.ActiveSheet.Range("C2").Select     h.PasteSpecial (2) ' Fórmulas     ActiveWorkbook.ActiveSheet.Range("D2").Select     h.PasteSpecial (3) ' Valores     ActiveWorkbook.ActiveSheet.Range("E2").Select     h.PasteSpecial (4) ' Formatos     ActiveWorkbook.ActiveSheet.Range("F2").Select     h.PasteSpecial (5) ' Comentarios     ActiveWorkbook.ActiveSheet.Range("G2").Select     h.PasteSpecial (6) ' Validaciones     ActiveWorkbook.ActiveSheet.Range("H2").Select     h.PasteSpecial (7) ' Todo excepto bordes     ActiveWorkbook.ActiveSheet.Range("I2").Select     h.PasteSpecial (8) ' Ancho columnas     ActiveWorkbook.ActiveSheet.Range("J2").Select     h.PasteSpecial (11) ' Formatos de números y fórmulas     ActiveWorkbook.ActiveSheet.Range("K2").Select     h.PasteSpecial (12) ' Formatos de números y valores     ActiveWorkbook.ActiveSheet.Range("L2").Select     h.PasteSpecial (13) ' Todos los formatos condicionales de     combinación End Sub </pre>

Resultado												
1	A	B	C	D	E	F	G	H	I	J	K	L
2	1.000,00	Paco: comentario	2	3	4	5	6	7	8	11	12	13
3	2.000,00											
4	3.000,00											
5	6.000,00											

1	A	B	C	D	E	F	G	H	I	J	K	L
2	1.000,00	1.000,00	1000	1000	4	5	6	7	8	11	12	13
3	2.000,00	2.000,00	2000	2000				1.000,00		1.000,00	1.000,00	1.000,00
4	3.000,00	3.000,00	3000	3000				2.000,00		2.000,00	2.000,00	2.000,00
5	6.000,00	6.000,00	6000	6000				3.000,00		3.000,00	3.000,00	3.000,00
								6.000,00		6.000,00	6.000,00	6.000,00

En este ejemplo, vemos como se copia el contenido de rango A2:A5 sobre diversos rangos usando un formato diferente en cada uno de los mismos. En la celda A2 hay un comentario, en la celda A3 una validación de datos y en la celda A4 un formato condicional.

### 2.3.2.16 PIVOTTABLES

#### 2.3.2.16.1 Descripción

Proporciona una colección de objetos que representan las tablas dinámicas existentes en la hoja indicada.

#### 2.3.2.16.2 Sintaxis:

**PivotTables (Index)**

#### 2.3.2.16.3 Argumentos

Argumento	Comentario
<b>Index</b>	Número o nombre del informe.

#### 2.3.2.16.4 Ejemplo

Código
<pre> Sub prueba_PivotTables()     Dim i As Integer     For i = 1 To ActiveWorkbook.ActiveSheet.PivotTables.Count         Debug.Print ActiveWorkbook.ActiveSheet.PivotTables(i).Name     Next i End Sub </pre>

Resultado
<div>Inmediato</div> <div>Tabla dinámica1</div>

### 2.3.2.17 PIVOTTABLEWIZARD

#### 2.3.2.17.1 Descripción

Permite crear una tabla dinámica sin utilizar el asistente.

#### 2.3.2.17.2 Sintaxis:

```
PivotTableWizard(SourceType, SourceData,
TableDestination, TableName, RowGrand, ColumnGrand,
SaveData, HasAutoFormat, AutoPage, Reserved,
BackgroundQuery, OptimizeCache, PageFieldOrder,
PageFieldWrapCount, ReadData, Connection)
```

#### 2.3.2.17.3 Argumentos

Argumento	Comentario
<b>SourceType</b>	<p>Valor que indica el origen de los datos y puede ser alguno de los siguientes:</p> <p><b>xlConsolidation</b> (3) Rangos de consolidación.</p> <p><b>xlDatabase</b> (1) Base de datos de Microsoft Excel.</p> <p><b>xlExternal</b> (2) Datos de otra aplicación.</p> <p><b>xlPivotTable</b> (-4148) Mismo origen que otro informe de tabla dinámica.</p> <p><b>xlScenario</b> (4) Datos basados en escenarios creados con el Administrador de escenarios.</p>
<b>SourceData</b>	Datos para el informe. Puede utilizarse un objeto Range.

<b>TableDestination</b>	Lugar donde colocar el informe de tabla dinámica.
<b>TableName</b>	Nombre del nuevo informe.
<b>RowGrand</b>	True indica que se han de mostrar los totales generales de las filas en el informe.
<b>ColumnGrand</b>	True indica que se han de mostrar los totales generales de las columnas en el informe.
<b>SaveData</b>	True indica que se han de guardar los datos con el informe. Si False, sólo se guarda la definición.
<b>HasAutoFormat</b>	True indica que se ha de dar formato automáticamente al informe cuando se actualice o cuando los campos se desplazan.
<b>AutoPage</b>	True indica que se crea un campo de página para la consolidación si <b>SourceType</b> tiene el valor <b>xlConsolidation</b> .
<b>Reserved</b>	No usado.
<b>BackgroundQuery</b>	True indica que se realizan consultas en segundo plano.
<b>OptimizeCache</b>	True indica que se optimiza la caché de la tabla dinámica al ser construida.
<b>PageFieldOrder</b>	Orden en que se agregan los campos de página al diseño del informe de tabla dinámica. Puede ser alguno de los siguientes valores de la constante <b>XlOrder</b> : <b>xlDownThenOver</b> o <b>xlOverThenDown</b> .
<b>PageFieldWrapCount</b>	Número de campos de página ubicados en cada columna o fila del informe de tabla dinámica.
<b>ReadData</b>	True indica que se crea una caché de tabla dinámica con todos los registros de la base de datos externa.
<b>Connection</b>	Cadena con la configuración para establecer una conexión a un origen de datos ODBC.

2.3.2.17.4 Ejemplo

Código

```
Sub prueba_PivotTableWizard()  
    ActiveSheet.PivotTableWizard SourceType:=xlDatabase, _  
        sourceData:=ActiveSheet.Range("A1:C10"), _  
        TableDestination:=ActiveSheet.Range("E1"), _  
        TableName:="Informe de pruebas", _  
        RowGrand:=True, _  
        ColumnGrand:=True, _  
        SaveData:=True, _  
        HasAutoFormat:=True  
End Sub
```

Resultado

	A	B	C
1	Nombre	Provincia	Importe
2	Juan	Barcelona	25
3	Juan	Madrid	100
4	Juan	Valencia	1000
5	Maria	Barcelona	20
6	Maria	Madrid	200
7	Maria	Valencia	2000
8	Paco	Barcelona	50
9	Paco	Madrid	300
10	Paco	Valencia	3000

En este ejemplo puede observarse como se crea una tabla dinámica a partir del rango indicado.

2.3.2.18 PRINTOUT

2.3.2.18.1 Descripción

Permite imprimir la hoja indicada.

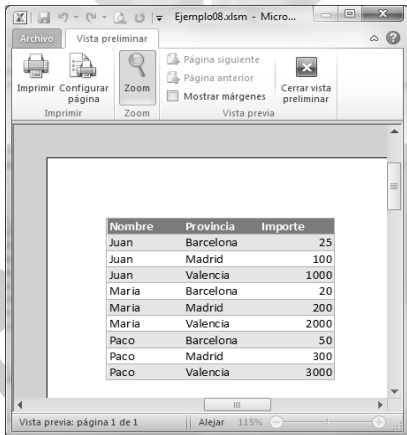
2.3.2.18.2 Sintaxis:

PrintOut(From, To, Copies, Preview, ActivePrinter, PrintToFile, Collate, PrToFileName, IgnorePrintAreas)

### 2.3.2.18.3 Argumentos

Argumento	Comentario
<b>From</b>	Primera página a imprimir.
<b>To</b>	Ultima página a imprimir.
<b>Copies</b>	Número de copias.
<b>Preview</b>	Si se indica True se realiza una vista preliminar.
<b>ActivePrinter</b>	Nombre de la impresora a utilizar.
<b>PrintToFile</b>	Si se indica True se la impresión se envía a un archivo.
<b>Collate</b>	Si se indica True se pueden intercalar varias copias.
<b>PrToFileName</b>	Indica el nombre del fichero hacia el que se desea enviar la impresión.
<b>IgnorePrintAreas</b>	Si se indica True se imprime todo el libro. De lo contrario, se imprimen sólo las áreas de impresión definidas.

### 2.3.2.18.4 Ejemplo

Código
<pre>Sub prueba_PrintOut ()     ActiveSheet.PrintOut Preview:=True End Sub</pre>
Resultado


### 2.3.2.19 PRINTPREVIEW

#### 2.3.2.19.1 Descripción

Permite realizar una previsualización de la hoja para ver cómo quedaría impresa.

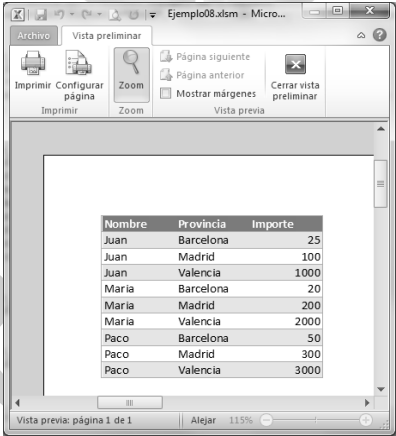
#### 2.3.2.19.2 Sintaxis:

**PrintPreview(EnableChanges)**

#### 2.3.2.19.3 Argumentos

Argumento	Comentario
<b>EnableChanges</b>	Si se indica True, el usuario puede realizar cambios durante la vista preliminar en ciertos parámetros como son los márgenes, la orientación de la página, el encabezado y pie, etc... Se trata de un parámetro opcional.

#### 2.3.2.19.4 Ejemplo

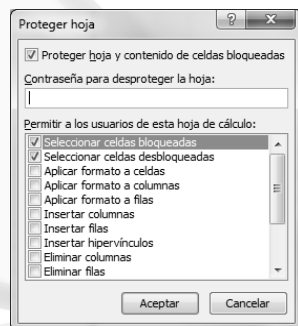
Código
<pre>Sub prueba_PrintPreview()     ActiveSheet.PrintPreview EnableChanges:=True End Sub</pre>
Resultado




## 2.3.2.20 PROTECT

### 2.3.2.20.1 Descripción

Protege una hoja para que no se pueda modificar. Tiene la misma funcionalidad que conseguimos manualmente a partir de la opción **Proteger hoja** ubicada en la pestaña **Revisar**:



### 2.3.2.20.2 Sintaxis:

```
Protect (Password,      DrawingObjects,      Contents,
Scenarios,      UserInterfaceOnly,      AllowFormattingCells,
AllowFormattingColumns,      AllowFormattingRows,
AllowInsertingColumns,      AllowInsertingRows,
AllowInsertingHyperlinks,      AllowDeletingColumns,
AllowDeletingRows,      AllowSorting,      AllowFiltering,
AllowUsingPivotTables)
```

### 2.3.2.20.3 Argumentos

Argumento	Comentario
<b>Password</b>	Contraseña a utilizar para la hoja.
<b>DrawingObjects</b>	Si True, se protegen las formas.
<b>Contents</b>	Si True, se protege el contenido.
<b>Scenarios</b>	Si True, se protegen los escenarios.
<b>UserInterfaceOnly</b>	Si True, se protege la interfaz de usuario.

<b>AllowFormattingCells</b>	Si True se permite formatear cualquier celda la hoja.
<b>AllowFormattingColumns</b>	Si True se permite rio formatear cualquier columna la hoja.
<b>AllowFormattingRows</b>	Si True se permite formatear cualquier fila la hoja.
<b>AllowInsertingColumns</b>	Si True se permite insertar columnas.
<b>AllowInsertingRows</b>	Si True se permite insertar filas.
<b>AllowInsertingHyperlinks</b>	Si True se permiten insertar hiperlinks.
<b>AllowDeletingColumns</b>	Si True se permite borrar columnas.
<b>AllowDeletingRows</b>	Si True se permite borrar filas.
<b>AllowSorting</b>	Si True se permite ordenar.
<b>AllowFiltering</b>	Si True se permite establecer filtros.
<b>AllowUsingPivotTables</b>	Si True se permite utilizar informes de tabla dinámica.

#### 2.3.2.20.4 Ejemplo

Código
<pre>Sub prueba_Protect()     ActiveSheet.Protect Password="hola" End Sub</pre>

Ests ejemplo protege la hoja y le establece una contraseña para su desprotección.

#### 2.3.2.21 RESETALLPAGEBREAKS

##### 2.3.2.21.1 Descripción

Permite restablecer los saltos de página de la hoja indicada.

### 2.3.2.21.2 Sintaxis:

**ResetAllPageBreaks**

### 2.3.2.21.3 Argumentos

No tiene.

### 2.3.2.21.4 Ejemplo

Código	
Sub prueba_ResetAllPageBreaks (	
ActiveSheet.ResetAllPageBreaks	
End Sub	

Resultado	

En este ejemplo puede comprobarse como una hoja que tiene los saltos de página configurados manualmente, recupera los saltos por defecto tras ejecutar el procedimiento.

## 2.3.2.22 SAVEAS

### 2.3.2.22.1 Descripción

Guarda los cambios del gráfico o de la hoja de cálculo en un archivo diferente.

### 2.3.2.22.2 Sintaxis:

**SaveAs (FileName, FileFormat, Password, WriteResPassword, ReadOnlyRecommended, CreateBackup, AddToMru, TextCodepage, TextVisualLayout, Local)**

### 2.3.2.22.3 Argumentos

Argumento	Comentario
<b>FileName</b>	Nombre del archivo
<b>FileFormat</b>	Formato con el que se guardará el archivo. Puede ser alguno de los formatos representados por la constante <b>XIFileFormat</b> .
<b>Password</b>	Contraseña a utilizar para la apertura.
<b>WriteResPassword</b>	Contraseña a utilizar contra escritura.
<b>ReadOnlyRecommended</b>	Si True, se sugiere abrir solo de lectura.
<b>CreateBackup</b>	Si True, crea copia de seguridad.
<b>AddToMru</b>	Si True, se incluye en la lista de archivos usados recientemente.
<b>TextCodepage</b>	No usado en versión en inglés de EE.UU de MS Excel.
<b>TextVisualLayout</b>	No usado en versión en inglés de EE.UU de MS Excel.
<b>Local</b>	Si True, guarda los archivos en el idioma de MS Excel sino, los guarda en el idioma de Visual Basic para Aplicaciones (VBA).

#### 2.3.2.22.4 Ejemplo

Código
<pre>Sub prueba_SaveAs()     ActiveSheet.SaveAs Filename:="C:\Temp\jg.xls", _         ReadOnlyRecommended:=True, _         AddToMru:=True End Sub</pre>

Este ejemplo, guarda el libro en curso incluyéndolo en la lista de ‘recientemente usados’ y proponiendo en la apertura del mismo que sea sólo de lectura.

#### 2.3.2.23 ESCENARIOS

##### 2.3.2.23.1 Descripción

Proporciona un objeto o una colección de objetos de tipo Scenario existentes en la hoja indicada.

##### 2.3.2.23.2 Sintaxis:

**Scenarios (Index)**

##### 2.3.2.23.3 Argumentos

Argumento	Comentario
<b>Index</b>	Número o nombre del escenario a tratar.

#### 2.3.2.23.4 Ejemplo

Código
<pre>Sub prueba_Scenarios()     Debug.Print "Hay " &amp; ActiveSheet.Scenarios.Count &amp; "     escenario/s."     Dim c As Scenario     For Each c In ActiveSheet.Scenarios         With c             Debug.Print "Escenario: " &amp; .Name         End With     Next End Sub</pre>

Resultado	
	<div>Inmediato</div> <div> Hay 2 escenario/s.  Escenario: VENTAS1  Escenario: VENTAS2 </div>

En este ejemplo puede observarse que en la hoja activa hay 2 escenarios.

### 2.3.2.24 SELECT

#### 2.3.2.24.1 Descripción

Selecciona el objeto.

#### 2.3.2.24.2 Sintaxis:

**Select (Replace)**

#### 2.3.2.24.3 Argumentos

Argumento	Comentario
<b>Replace</b>	Si es True, la selección actual es reemplazada con con el objeto especificado.

#### 2.3.2.24.4 Ejemplo

Código
<pre>Sub prueba_Select()     ActiveWorkbook.Worksheets("Hoja9").Select End Sub</pre>

Tras la ejecución del procedimiento, puede observarse que se selecciona la Hoja9.

### 2.3.2.25 SETBACKGROUNDPICTURE

#### 2.3.2.25.1 Descripción

Permite incluir una imagen como fondo de la hoja.


#### 2.3.2.25.2 Sintaxis:

**SetBackgroundPicture (Filename)**

#### 2.3.2.25.3 Argumentos

Argumento	Comentario
Filename	Nombre de fichero a utilizar como fondo de la hoja indicada.

#### 2.3.2.25.4 Ejemplo

Código
<pre>Sub prueba_SetBackgroundPicture()     ActiveSheet.SetBackgroundPicture ("C:\Temp\epoca2.bmp") End Sub</pre>
Resultado


Al ejecutar este procedimiento, vemos como la imgame indicada se coloca como fondo de la hoja.

### 2.3.2.26 SHOWALLDATA

#### 2.3.2.26.1 Descripción

Permite visualizar todas las filas sobre las que se ha aplicado un autofiltro. Equivale a elegir la opción de mostrar ‘Todas’ las filas.

#### 2.3.2.26.2 Sintaxis:

**ShowAllData**

#### 2.3.2.26.3 Argumentos

No tiene.

#### 2.3.2.26.4 Ejemplo

## Código

```
Sub prueba_ShowAllData()  
    ActiveSheet.ShowAllData  
End Sub
```

## Resultado

	A	B	C
1	Nombre	Provincia	Importe
2	Juan	Barcelona	25
5	Maria	Barcelona	20
8	Paco	Barcelona	50

	A	B	C
1	Nombre	Provincia	Importe
2	Juan	Barcelona	25
3	Juan	Madrid	100
4	Juan	Valencia	1000
5	Maria	Barcelona	20
6	Maria	Madrid	200
7	Maria	Valencia	2000
8	Paco	Barcelona	50
9	Paco	Madrid	300
10	Paco	Valencia	3000

### 2.3.2.27 SHOWDATAFORM

#### 2.3.2.27.1 Descripción

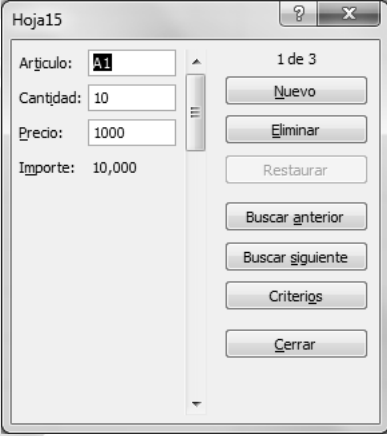
Visualiza un formulario estándar mediante el cual se pueden modificar los datos de la hoja indicada.



**2.3.2.27.2 Sintaxis:****ShowDataForm****2.3.2.27.3 Argumentos**

No tiene

**2.3.2.27.4 Ejemplo**

Código																													
<pre>Sub prueba_ShowDataForm()     ActiveSheet.ShowDataForm End Sub</pre>																													
Resultado																													
<div style="display: flex; justify-content: space-between;"> <table border="1" style="width: 45%;"> <thead> <tr> <th></th> <th>A</th> <th>B</th> <th>C</th> <th>D</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Articulo</td> <td>Cantidad</td> <td>Precio</td> <td>Importe</td> </tr> <tr> <td>2</td> <td>A1</td> <td>10</td> <td>1.000</td> <td>10.000</td> </tr> <tr> <td>3</td> <td>A2</td> <td>20</td> <td>2.000</td> <td>40.000</td> </tr> <tr> <td>4</td> <td>A3</td> <td>30</td> <td>3.000</td> <td>90.000</td> </tr> </tbody> </table> <div style="width: 50%;">  </div> </div>						A	B	C	D	1	Articulo	Cantidad	Precio	Importe	2	A1	10	1.000	10.000	3	A2	20	2.000	40.000	4	A3	30	3.000	90.000
	A	B	C	D																									
1	Articulo	Cantidad	Precio	Importe																									
2	A1	10	1.000	10.000																									
3	A2	20	2.000	40.000																									
4	A3	30	3.000	90.000																									

**2.3.2.28 UNPROTECT****2.3.2.28.1 Descripción**

Permite eliminar la protección de la hija indicada siempre que previamente ésta haya sido protegida.

**2.3.2.28.2 Sintaxis:****Unprotect (Password)**

### 2.3.2.28.3 Argumentos

Argumento	Comentario
<b>Password</b>	Palabara de paso a utilizar para la desprotección.

### 2.3.2.28.4 Ejemplo

Código
<pre>Sub prueba_Unprotect()     ActiveSheet.Unprotect End Sub</pre>

Tras la ejecución de este procedimiento, la hoja queda desprotegida.

## 2.3.2.29 XMLDATAQUERY

### 2.3.2.29.1 Descripción

Devuelve un objeto Range que representa las celdas asignadas a una determinada expresión XPath. Devuelve Nothing si la expresión XPath especificada no se ha asignado a la hoja de cálculo o si el rango asignado está vacío.

### 2.3.2.29.2 Sintaxis:

**XmlDataQuery(XPath, SelectionNamespaces, Map)**

### 2.3.2.29.3 Argumentos

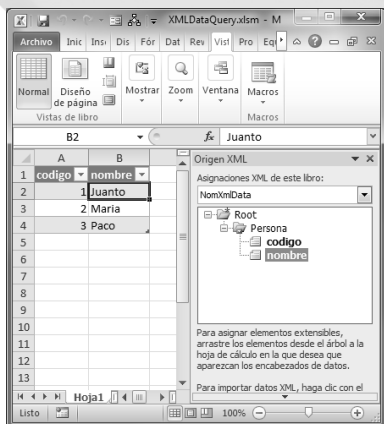
Argumento	Comentario
<b>XPath</b>	Ruta sobre la que se realiza la consulta (XPath).
<b>SelectionNamespaces</b>	Espacio de nombres a los que se refiere XPath.
<b>Map</b>	Asignación XML agregada previamente al libro.

### 2.3.2.29.4 Ejemplo

#### Código

```
Sub prueba_XMLDataQuery()  
    Dim path As String  
    path = "/Root/Persona"  
    Dim namespaces As String  
    namespaces = "xmlns=http://www.w3.org/2001/XMLSchema"  
    Dim r As Range  
    Set r = ActiveSheet.XmlDataQuery(path, namespaces)  
    If r Is Nothing Then  
        MsgBox "No es posible asociar el mapa indicado"  
        Exit Sub  
    End If  
    Debug.Print r.Cells(1, 1).Value  
    Debug.Print r.Cells(1, 2).Value  
    Debug.Print r.Address  
End Sub
```

#### Resultado



Inmediato  
1  
Juanto  
\$A\$2:\$A\$4, \$B\$2:\$B\$4

### 2.3.2.30 XMLMAPQUERY

#### 2.3.2.30.1 Descripción

Proporciona un objeto de tipo Range con las celdas asociadas a una expresión XPATH.

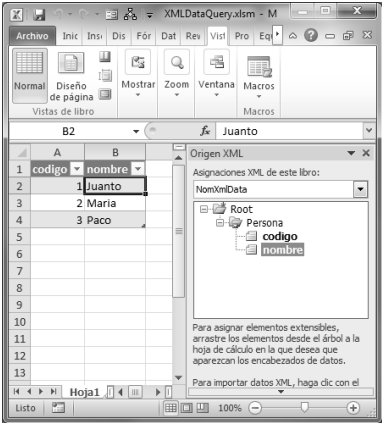
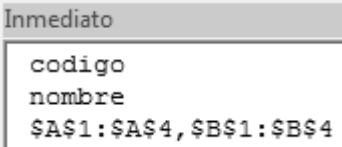
#### 2.3.2.30.2 Sintaxis:

**XmlMapQuery(XPath, SelectionNamespaces, Map)**

2.3.2.30.3 Argumentos

Argumento	Comentario
XPath	Ruta sobre la que se realiza la consulta (XPath).
SelectionNamespaces	Espacio de nombres a los que se refiere XPath.
Map	Asignación XML a utilizar si se desea que la búsqueda de XPath se realice sobre la misma.

2.3.2.30.4 Ejemplo

Código	
<pre>Sub prueba_XmlMapQuery() Dim path As String path = "/Root/Persona" Dim namespaces As String namespaces = "xmlns=http://www.w3.org/2001/XMLSchema" Dim r As Range Set r = ActiveSheet.XmlMapQuery(path, namespaces) If r Is Nothing Then     MsgBox "No es posible asociar el mapa indicado"     Exit Sub End If Debug.Print r.Cells(1, 1).Value Debug.Print r.Cells(1, 2).Value Debug.Print r.Address End Sub</pre>	
Resultado	
	

### **2.3.3 Eventos**

#### **2.3.3.1 ACTIVATE**

Evento producido al activar un libro, hoja o gráfico incrustado.

#### **2.3.3.2 BEFOREDOUBLECLICK**

Producido al hacer doble clic en una hoja. Se produce antes de la acción predeterminada.

#### **2.3.3.3 BEFORERIGHTCLICK**

Producido al hacer clic con el botón secundario del mouse en una hoja. Se produce antes de la acción predeterminada.

#### **2.3.3.4 CALCULATE**

Producido después de que una hoja se actualice.

#### **2.3.3.5 CHANGE**

Producido al modificar una celda por parte del usuario o de un vínculo externo.

#### **2.3.3.6 DEACTIVATE**

Producido al desactivar un libro, una hoja o un gráfico.

#### **2.3.3.7 FOLLOWHYPERLINK**

Producido al hacer clic sobre un hipervínculo.

#### **2.3.3.8 PIVOTTABLEAFTERVALUECHANGE**

Producido al calcular o editar un rango dentro de una tabla dinámica.

#### **2.3.3.9 PIVOTTABLEBEFOREALLOCATECHANGES**

Producido antes de que se apliquen los cambios a una tabla dinámica.

#### **2.3.3.10 PIVOTTABLEBEFORECOMMITCHANGES**

Producido antes de que se confirmen los cambios en el origen de datos OLAP de una tabla dinámica.

### 2.3.3.11 PIVOTTABLEBEFOREDISCARDCHANGES

Producido antes de descartar los cambios hechos en una tabla dinámica.

### 2.3.3.12 PIVOTTABLECHANGESYNC

Producido tras realizar cambios en una tabla dinámica.

### 2.3.3.13 PIVOTTABLEUPDATE

Producido tras actualizar un informe de tabla dinámica.

### 2.3.3.14 SELECTIONCHANGE

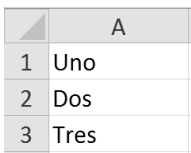
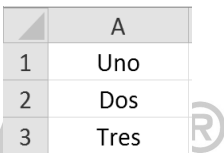
Producido al cambiar la selección de una hoja.

## 2.4 EL OBJETO RANGO (RANGE).

### 2.4.1 Propiedades

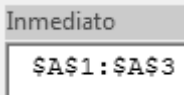
#### 2.4.1.1 ADDINDENT

Define o consulta si hay que aplicar sangría automáticamente a un rango que tenga aplicado un formato de alineación vertical u horizontal.

Código	
<pre>Sub prueba_AddIndent()     ActiveSheet.Range("A1:A3").HorizontalAlignment = xlHAlignDistributed     ActiveSheet.Range("A1:A3").AddIndent = True End Sub</pre>	
Resultado	
	
Inicio	Fin

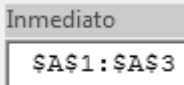
### 2.4.1.2 ADDRESS

Proporciona un string que representa la referencia del rango indicado.

Código	
<pre>Sub prueba_Address()     Debug.Print ActiveSheet.Range("A1:A3").Address End Sub</pre>	
Resultado	
	

### 2.4.1.3 ADDRESSLOCAL

Proporciona un string que representa la referencia del rango indicado en el idioma del usuario.

Código	
<pre>Sub prueba_AddressLocal()     Debug.Print ActiveSheet.Range("A1:A3").AddressLocal End Sub</pre>	
Resultado	
	

### 2.4.1.4 ALLOWEDIT

Indica mediante un valor de tipo Boolean si un rango puede ser modificado o no en una hoja protegida.

Código	
<pre>Sub prueba_AllowEdit()     Debug.Print ActiveSheet.Range("A1:A3").AllowEdit End Sub</pre>	

Resultado
En una hoja protegida tenemos que: <div>Inmediato Falso</div>

### 2.4.1.5 APPLICATION

Proporciona objeto representante de la aplicación Microsoft Excel.

Código
<pre>Sub prueba_Application()     Debug.Print ActiveSheet.Range("A1:A3").Application End Sub</pre>
Resultado
<div>Inmediato Microsoft Excel</div>

### 2.4.1.6 AREAS

Proporciona una colección de objetos de tipo Areas con todos los rangos que se hayan seleccionados.

Código
<pre>Sub prueba_Areas()     Debug.Print Selection.Areas.Count     Dim r As Range     For Each r In Selection.Areas         Debug.Print r.Address     Next End Sub</pre>



Resultado			
	A	B	C
1	Uno	10	A
2	Dos	20	B
3	Tres	30	C

### 2.4.1.7 BORDERS

Devuelve una colección Borders que representa los bordes de un estilo o un rango de celdas (incluido un rango definido como parte de un formato condicional).

Proporciona una colección de objetos de tipo Borders con todos los bordes de un estilo o rango.

Código																							
<pre>Sub prueba_Borders()     Debug.Print ActiveSheet.Range("A1").Borders.LineStyle     Debug.Print ActiveSheet.Range("B2").Borders.LineStyle     Debug.Print ActiveSheet.Range("C3").Borders.LineStyle End Sub</pre>																							
Resultado																							
<table><tr><td></td><td>A</td><td>B</td><td>C</td></tr><tr><td>1</td><td>Uno</td><td>10</td><td>A</td></tr><tr><td>2</td><td>Dos</td><td>20</td><td>B</td></tr><tr><td>3</td><td>Tres</td><td>30</td><td>C</td></tr></table>			A	B	C	1	Uno	10	A	2	Dos	20	B	3	Tres	30	C	<table><tr><td>Inmediato</td></tr><tr><td>-4115</td></tr><tr><td>-4119</td></tr><tr><td>-4142</td></tr></table>		Inmediato	-4115	-4119	-4142
	A	B	C																				
1	Uno	10	A																				
2	Dos	20	B																				
3	Tres	30	C																				
Inmediato																							
-4115																							
-4119																							
-4142																							

### 2.4.1.8 CELLS

Proporciona un objeto que representa todas las celdas del rango indicado.

Código
<pre>Sub prueba_Cells()     Dim r As Range     Dim f, c, t As Double     Set r = Range("A1:C3")</pre>

```

For f = 1 To r.Rows.Count
    For c = 1 To r.Columns.Count
        If IsNumeric(r.Cells(f, c).Value) Then
            t = t + r.Cells(f, c).Value
            r.Cells(f, c).Interior.ColorIndex = 34
        End If
    Next c
Next f
Debug.Print "Total : " & Format(t, "#.00")
End Sub

```

### Resultado

	A	B	C
1	Uno	10	A
2	20	Dos	B
3	Tres	30	5

Inmediato  
 Total : 65,00

	A	B	C
1	Uno	10	A
2	20	Dos	B
3	Tres	30	5

En este ejemplo, se suman todas las celdas que contienen un valor numérico y se cambia el color del fondo a las mismas.

#### 2.4.1.9 CHARACTERS

Proporciona un objeto de tipo Characters que permite tratar el texto del objeto indicado.

### Código

```

Sub prueba_Characters()
    Dim r As Range
    Dim f, c As Double
    Dim s As Range
    Set r = Range("A1:C3")
    For f = 1 To r.Rows.Count
        For c = 1 To r.Columns.Count
            If Not IsNumeric(r.Cells(f, c).Value) Then
                r.Cells(f, c).Characters(2, 1).Font.Color = -
16776961
                Set s = Cells(f, c)
                Debug.Print s.Characters.Text
            End If
        Next c
    Next f
End Sub

```

Resultado											
	A	B	C		A	B	C	Inmediato			
1	Uno	10	A	1	Uno	10	A	Uno			
2	20	Dos	B	2	20	Dos	B	A			
3	Tres	30	5	3	Tres	30	5	Dos			
								B			
								Tres			

Este ejemplo pone en rojo la segunda letra de cada cadena de texto hallada en el rango tratado e imprime en la ventana de Inmediato el contenido de aquellas celdas que contienen letras.

#### 2.4.1.10 COLUMN

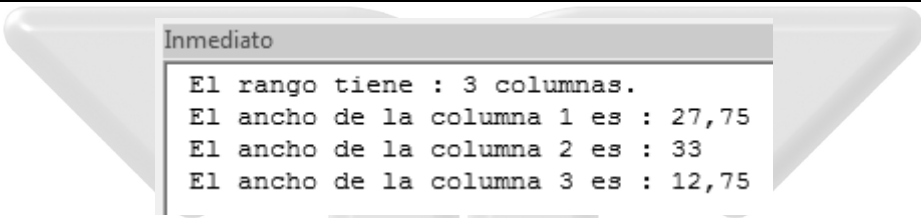
Indica el número de columna correspondiente a la primera columna del rango indicado.

Código
<pre> Sub prueba_Column()     Dim r As Range     Set r = Range("B1:C3")     Debug.Print "La primera columna es: " &amp; r.Column End Sub </pre>
Resultado
<div>Inmediato</div> <div>La primera columna es: 2</div>

Ra-Ma®

### 2.4.1.11 COLUMNS

Proporciona un objeto de tipo Range representando todas las columnas del rango indicado.

Código
<pre>Sub prueba_Columns()     Dim r As Range     Dim n As Integer     Set r = Range("A1:C3")     Debug.Print "El rango tiene : " &amp; r.Columns.Count &amp; " columnas."     For n = 1 To r.Columns.Count         Debug.Print "El ancho de la columna " &amp; n &amp; " es : " &amp; r.Columns(n).Width         Next n     End Sub</pre>
Resultado


### 2.4.1.12 COLUMNWIDTH

Devuelve o establece el ancho de las columnas del rango especificado. Variant de lectura y escritura.

Define o consulta el ancho de las columnas del rango indicado siempre y cuando todas las columnas tengan el mismo ancho. De lo contrario, devuelve null.

Código
<pre>Sub prueba_ColumnWidth()     Dim r As Range     Set r = Range("A1")     Debug.Print "El ancho es : " &amp; r.ColumnWidth     End Sub</pre>

Resultado
<div>Inmediato</div> <div>El ancho es : 4,57</div>

### 2.4.1.13 COMMENT

Proporciona un objeto de tipo Comment que representa el comentario asociado a la celda de la esquina superior del rango indicado.

Código
<pre>Sub prueba_Comment()     Debug.Print "Comentario : " &amp; Range("A1").Comment.Text End Sub</pre>
Resultado
<div>Inmediato</div> <div>Comentario : Paco: Es una prueba de comentario.</div>

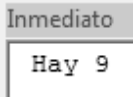
### 2.4.1.14 COUNT

Indica el número de celdas del rango indicado.

Código
<pre>Sub prueba_Count()     Debug.Print "Hay " &amp; Range("A1:C2").Count &amp; " celdas." End Sub</pre>
Resultado
<div>Inmediato</div> <div>Hay 6 celdas.</div>

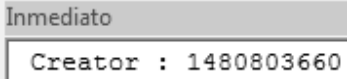
### 2.4.1.15 COUNTLARGE

Indica el número de celdas del rango indicado. Es similar a Count pero permite obtener almacenar valores mayores (por ejemplo Excel 2007 y 2010 contienen mas celdas en una hoja que en versiones anteriores).

Código
<pre>Sub prueba_CountLarge()     Debug.Print "Hay " &amp; Range("A1:C3").CountLarge End Sub</pre>
Resultado


### 2.4.1.16 CREATOR

Proporciona valor que indica que aplicación creó el objeto.

Código
<pre>Sub prueba_Creator()     Debug.Print "Creator : " &amp; Range("A1:C3").Creator End Sub</pre>
Resultado


### 2.4.1.17 CURRENTARRAY

Proporciona un objeto de tipo Range con el rango que contiene todos los elementos de la matriz a la que pertenece la celda indicada.


Código
<pre>Sub prueba_CurrentArray()     Range("C2").CurrentArray.Select End Sub</pre>

Resultado																																						
<div>C2</div> <div><div></div><div>fx</div><div>{=A1:A3*B1:B3}</div></div> <table><tr><th></th><th>A</th><th>B</th><th>C</th></tr><tr><td>1</td><td>10</td><td>1000</td><td>10000</td></tr><tr><td>2</td><td>20</td><td>2000</td><td>40000</td></tr><tr><td>3</td><td>30</td><td>3000</td><td>90000</td></tr></table>					A	B	C	1	10	1000	10000	2	20	2000	40000	3	30	3000	90000	<div>C1</div> <div><div></div><div>fx</div><div>{=A1:A3*B1:B3}</div></div> <table><tr><th></th><th>A</th><th>B</th><th>C</th></tr><tr><td>1</td><td>10</td><td>1000</td><td>10000</td></tr><tr><td>2</td><td>20</td><td>2000</td><td>40000</td></tr><tr><td>3</td><td>30</td><td>3000</td><td>90000</td></tr></table>				A	B	C	1	10	1000	10000	2	20	2000	40000	3	30	3000	90000
	A	B	C																																			
1	10	1000	10000																																			
2	20	2000	40000																																			
3	30	3000	90000																																			
	A	B	C																																			
1	10	1000	10000																																			
2	20	2000	40000																																			
3	30	3000	90000																																			
Inicio				Fin																																		

En este ejemplo, se puede observar que la celda C2 pertenece a un array ( $\{=A1:A3*B1:B3\}$ ) y que al ejecutar el procedimiento se selecciona todo el array completo.

#### 2.4.1.18 CURRENTREGION

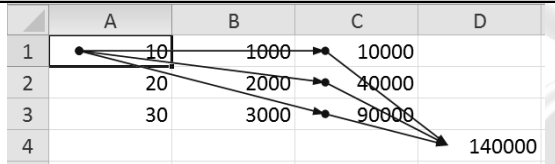
Devuelve un objeto de tipo Range que representa la región a la que pertenece un determinado rango. Se entiende por región, el rango que está delimitado por filas y columnas vacías.

Código																			
<pre>Sub prueba_CurrentRegion()     Range("B2").CurrentRegion.Select End Sub</pre>																			
Resultado																			
<table><tr><td></td><td>A</td><td>B</td><td>C</td></tr><tr><td>1</td><td>10</td><td>1000</td><td>10000</td></tr><tr><td>2</td><td>20</td><td>2000</td><td>40000</td></tr><tr><td>3</td><td>30</td><td>3000</td><td>90000</td></tr></table>					A	B	C	1	10	1000	10000	2	20	2000	40000	3	30	3000	90000
	A	B	C																
1	10	1000	10000																
2	20	2000	40000																
3	30	3000	90000																
<table><tr><td></td><td>A</td><td>B</td><td>C</td></tr><tr><td>1</td><td>10</td><td>1000</td><td>10000</td></tr><tr><td>2</td><td>20</td><td>2000</td><td>40000</td></tr><tr><td>3</td><td>30</td><td>3000</td><td>90000</td></tr></table>					A	B	C	1	10	1000	10000	2	20	2000	40000	3	30	3000	90000
	A	B	C																
1	10	1000	10000																
2	20	2000	40000																
3	30	3000	90000																
Inicio		Fin 																	

### 2.4.1.19 DEPENDENTS

Devuelve un objeto Range que representa el rango que contiene todas las celdas dependientes de una celda. Puede ser una selección múltiple (una unión de objetos Range) si hay más de una celda dependiente. Objeto Range de solo lectura.

Proporciona un objeto de tipo Range que representa todas las celdas dependientes de una celda.

Código				
<pre>Sub prueba_Dependents()     Debug.Print Range("A1").Dependents.Address End Sub</pre>				
Resultado				
				
				<div>Inmediato</div> <div>\$C\$1:\$C\$3,\$D\$4</div>

En este ejemplo, puede observarse cómo se devuelven las celdas dependientes de la celda A1. Efectivamente, C1 posee el valor resultante de  $A1 * B1$  y D4, la fórmula =SUMA(C1:C3).

### 2.4.1.20 DIRECTDEPENDENTS

Devuelve un objeto Range que representa el rango que contiene todas las celdas dependientes directamente de una celda.

Código				
<pre>Sub prueba_DirectDependents()     Debug.Print Range("A1").DirectDependents.Address End Sub</pre>				

Ra-Ma®



Resultado			
	A	B	C
1	10	1000	=A1*B1
2	20	2000	=A2*B2
3	30	3000	=A3*B3

Inmediato  
 \$C\$1

En este ejemplo, vemos que la celda que depende directamente de A1 es C1.

#### 2.4.1.21 DIRECTPRECEDENTS

Devuelve un objeto Range que representa el rango que contiene todas las celdas que preceden directamente a una celda.

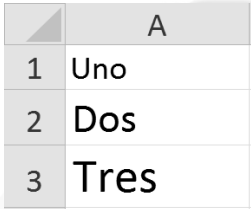
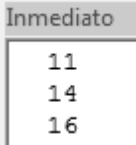
Código			
<pre>Sub prueba_DirectPrecedents()     Debug.Print Range("C1").DirectPrecedents.Address End Sub</pre>			
Resultado			
	A	B	C
1	10	1000	=A1*B1
2	20	2000	=A2*B2
3	30	3000	=A3*B3

Inmediato  
 \$A\$1:\$B\$1

En este ejemplo, vemos que la celda C1 es precedida por las celdas A1 y B1.

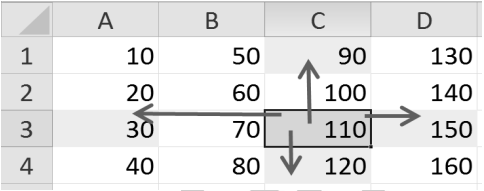
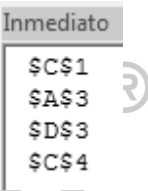
#### 2.4.1.22 DISPLAYFORMAT

Devuelve un objeto DisplayFormat que representa la configuración de visualización del rango especificado. Solo lectura.

Código	
<pre>Sub prueba_DisplayFormat()     Debug.Print Range("A1").DisplayFormat.Font.Size     Debug.Print Range("A2").DisplayFormat.Font.Size     Debug.Print Range("A3").DisplayFormat.Font.Size End Sub</pre>	
Resultado	
	
<b>Inicio</b>	<b>Fin</b>

### 2.4.1.23 END

Permite obtener la celda situada al final de la región en la que se halla el rango indicado. La celda devuelta dependerá de la dirección que indiquemos para la búsqueda la cual puede ser alguna de las siguientes: xlUp, xlToLeft, xlToRight, xlDown

Código	
<pre>Sub prueba_End()     Debug.Print Range("C3").End(xlUp).Address     Debug.Print Range("C3").End(xlToLeft).Address     Debug.Print Range("C3").End(xlToRight).Address     Debug.Print Range("C3").End(xlDown).Address End Sub</pre>	
Resultado	
	
<b>Inicio</b>	<b>Fin</b>

### 2.4.1.24 ENTIRECOLUMN

Proporciona un objeto de tipo Range con las columnas contenidas en el rango indicado.

Código				
<pre>Sub prueba_EntireColumn()     Range("B1:C4").EntireColumn.Select End Sub</pre>				
Resultado				
	A	B	C	D
1	10	50	90	130
2	20	60	100	140
3	30	70	110	150
4	40	80	120	160
Inicio			Fin	

### 2.4.1.25 ENTIREROW

Proporciona un objeto de tipo Range con las filas contenidas en el rango indicado.

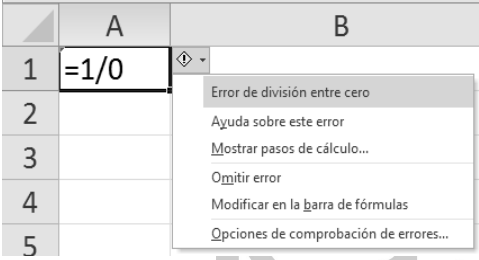
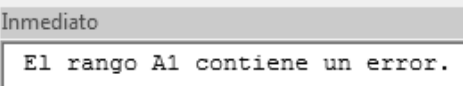
Código				
<pre>Sub prueba_EntireRow()     Range("B2:C3").EntireRow.Select End Sub</pre>				
Resultado				
	A	B	C	D
1	10	50	90	130
2	20	60	100	140
3	30	70	110	150
4	40	80	120	160
Inicio			Fin	

2.4.1.26 ERRORS

Permite al usuario tener acceso a las opciones de comprobación de errores.

Las constantes de errores que pueden comprobarse son:

- xlEvaluateToError = 1
- xlTextDate = 2
- xlNumberAsText= 3
- xlInconsistentFormula = 4
- xlOmittedCells = 5
- xlUnlockedFormulaCells = 6
- xlEmptyCellReferences = 7
- xlListDataValidation = 8

Código	
<pre>Sub prueba_Errors()   If Range("A1").Errors.Item(xlEvaluateToError).Value = True Then     Debug.Print "El rango A1 contiene un error."   End If End Sub</pre>	
Resultado	
	
Inicio	Fin

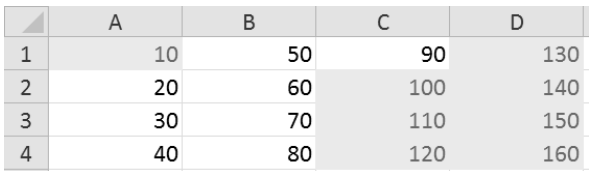

### 2.4.1.27 FONT

Devuelve un objeto Font que representa la fuente del objeto especificado.

Código	
<pre>Sub prueba_Font ()     Debug.Print Range("A1").Font.Name     Debug.Print Range("A2").Font.Name     Debug.Print Range("A3").Font.Name End Sub</pre>	
Resultado	
	
<b>Inicio</b>	<b>Fin</b>

### 2.4.1.28 FORMATCONDITIONS

Proporciona una colección de objetos con los formatos condicionales existentes en el rango indicado.

Código	
<pre>Sub prueba_FormatConditions()     Dim f As FormatCondition     For Each f In Range("A1:D4").FormatConditions         Debug.Print f.Formula1     Next End Sub</pre>	
Resultado	
	
<b>Inicio</b>	<b>Fin</b>

En este ejemplo, se aplica un formato condicional a aquellas celdas dentro del rango A1:D4 que contienen el número 1.

2.4.1.29 FORMULA

Devuelve o establece un valor de tipo Variant que representa la fórmula del objeto en notación de estilo A1 y en el lenguaje de la macro.

Define o consulta la fórmula o f

Código																					
<pre>Sub prueba_Formula()   Dim s As Variant   For Each s In Range("C1:C3").formula     Debug.Print s   Next End Sub</pre>																					
Resultado																					
<table><tr><th></th><th>A</th><th>B</th><th>C</th></tr><tr><td>1</td><td>10</td><td>1000</td><td>=A1*B1</td></tr><tr><td>2</td><td>20</td><td>2000</td><td>=A2*B2</td></tr><tr><td>3</td><td>30</td><td>3000</td><td>=A3*B3</td></tr></table>		A	B	C	1	10	1000	=A1*B1	2	20	2000	=A2*B2	3	30	3000	=A3*B3	<table><tr><th>Inmediato</th></tr><tr><td>=A1*B1</td></tr><tr><td>=A2*B2</td></tr><tr><td>=A3*B3</td></tr></table>	Inmediato	=A1*B1	=A2*B2	=A3*B3
	A	B	C																		
1	10	1000	=A1*B1																		
2	20	2000	=A2*B2																		
3	30	3000	=A3*B3																		
Inmediato																					
=A1*B1																					
=A2*B2																					
=A3*B3																					
Inicio	Fin																				

2.4.1.30 FORMULAARRAY

Define o consulta la fórmula de matriz asociada a un rango.

Código
<pre>Sub prueba_FormulaArray()   Debug.Print Range("C2").FormulaArray End Sub</pre>

Resultado																								
<table><tr><th></th><th>A</th><th>B</th><th>C</th></tr><tr><td>1</td><td>10</td><td>1000</td><td>=A1:A3*B1:B3</td></tr><tr><td>2</td><td>20</td><td>2000</td><td>=A1:A3*B1:B3</td></tr><tr><td>3</td><td>30</td><td>3000</td><td>=A1:A3*B1:B3</td></tr><tr><td>.</td><td></td><td></td><td></td></tr></table>					A	B	C	1	10	1000	=A1:A3*B1:B3	2	20	2000	=A1:A3*B1:B3	3	30	3000	=A1:A3*B1:B3	.				<div>Inmediato</div> <div>=A1:A3*B1:B3</div>
	A	B	C																					
1	10	1000	=A1:A3*B1:B3																					
2	20	2000	=A1:A3*B1:B3																					
3	30	3000	=A1:A3*B1:B3																					
.																								
Inicio				Fin																				

#### 2.4.1.31 FORMULAHIDDEN

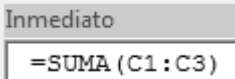
Define o consulta si la fórmula ha de ocultarse cuando la hoja esté protegida.

Código				
<pre>Sub prueba_FormulaHidden()   Range("C2").FormulaHidden = True End Sub</pre>				
Resultado				
	A	B	C	
1	10	1000	=A1*B1	
2	20	2000	=A2*B2	
3	30	3000	=A3*B3	
Inicio		Fin		

Al ejecutar proteger la hoja y el procedimiento, vemos que la fórmula del Rango C2 no se visualiza.

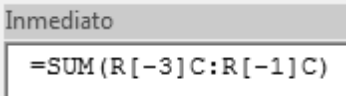
#### 2.4.1.32 FORMULALOCAL

Define o consulta la fórmula con referencias de estilo A1 en el idioma asociado al usuario.

Código
<pre>Sub prueba_FormulaLocal()     Debug.Print Range("C4").FormulaLocal End Sub</pre>
Resultado


#### 2.4.1.33 FORMULAR1C1

Define o consulta la fórmula con referencias de estilo R1C1 en el lenguaje de la macro.

Código
<pre>Sub prueba_FormulaR1C1()     Debug.Print Range("C4").FormulaR1C1 End Sub</pre>
Resultado


#### 2.4.1.34 FORMULAR1C1LOCAL

Define o consulta la fórmula con referencias de estilo R1C1 en el lenguaje del usuario.

Código
<pre>Sub prueba_FormulaR1C1Local()     Debug.Print Range("C4").FormulaR1C1Local End Sub</pre>



Resultado
<div>Inmediato</div> <div>=SUMA (F[-3]C:F[-1]C)</div>

### 2.4.1.35 HASARRAY

Indica mediante un valor de tipo Boolean si la celda indicada pertenece a una fórmula de matriz.

Código
<pre>Sub prueba_HasArray()     If Range("C2").HasArray = True Then         Debug.Print Range("C2").FormulaArray     End If End Sub</pre>
Resultado
<div>Inmediato</div> <div>=A1:A3*B1:B3</div>

### 2.4.1.36 HASFORMULA

Devuelve True si todas las celdas del rango contienen fórmulas. Devuelve False si no hay ninguna fórmula en ninguna celda del rango. En cualquier otro caso, devuelve null.

Código
<pre>Sub prueba_HasFormula()     Debug.Print Range("A1:A3").HasFormula     Debug.Print Range("C1:C3").HasFormula     Debug.Print Range("A1:C3").HasFormula End Sub</pre>

Resultado																				
<table><tr><th></th><th>A</th><th>B</th><th>C</th></tr><tr><td>1</td><td>10</td><td>1000</td><td>=A1*B1</td></tr><tr><td>2</td><td>20</td><td>2000</td><td>=A2*B2</td></tr><tr><td>3</td><td>30</td><td>3000</td><td>=A3*B3</td></tr></table>				A	B	C	1	10	1000	=A1*B1	2	20	2000	=A2*B2	3	30	3000	=A3*B3	<div>Inmediato</div> <div>Falso</div> <div>Verdadero</div> <div>Nulo</div>	
	A	B	C																	
1	10	1000	=A1*B1																	
2	20	2000	=A2*B2																	
3	30	3000	=A3*B3																	
Inicio			Fin																	

2.4.1.37 HEIGHT

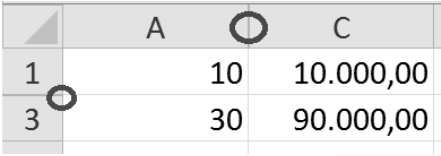
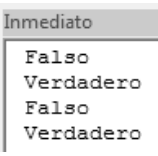
Define o consulta el alto en puntos de un rango.

Código
<pre>Sub prueba_Height()   Debug.Print Range("A1:A3").Height   Debug.Print Range("B1:B2").Height End Sub</pre>
Resultado
<div>Inmediato</div> <div>45</div> <div>30</div>

2.4.1.38 HIDDEN

Define o consulta si las filas o columnas del rango indicado están ocultas.

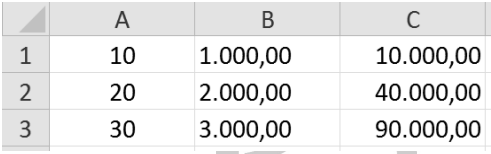
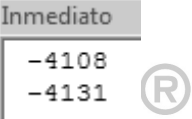
Código
<pre>Sub prueba_Hidden()   Debug.Print Range("A1:C3").Columns("A").Hidden   Debug.Print Range("A1:C3").Columns("B").Hidden   Debug.Print Range("A1:C3").Rows("1").Hidden   Debug.Print Range("A1:C3").Rows("2").Hidden End Sub</pre>

Resultado			
			
Inicio		Fin	

2.4.1.39 HORIZONTALALIGNMENT

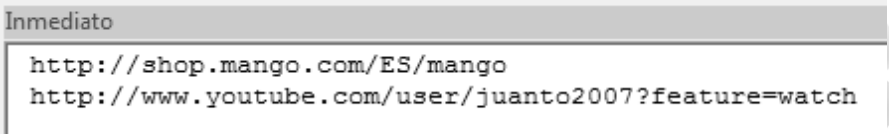
Define o consulta el tipo de alineación horizontal del rango indicado. Puede tener alguno de los siguientes valores:

- 4108 xlCenter
- 4117 xlDistributed
- 4130 xlJustify
- 4131 xlLeft
- 4152 xlRight

Código			
<pre>Sub prueba_HorizontalAlignment()     Debug.Print Range("A1:A3").HorizontalAlignment     Debug.Print Range("B1:B3").HorizontalAlignment End Sub</pre>			
Resultado			
			
Inicio		Fin	

### 2.4.1.40 HYPERLINKS

Proporciona una colección de objetos de tipo Hyperlink con todos los enlaces existentes en el rango indicado.

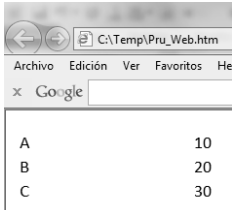
Código
<pre>Sub prueba_Hyperlinks()   Dim h As Hyperlink   For Each h In Range("A1:A2").Hyperlinks     Debug.Print h.Address   Next End Sub</pre>
Resultado


### 2.4.1.41 ID

Define o consulta las etiquetas de identificación asociadas a una celda cuando pertenecen a una hoja que ha sido guardada como página web.

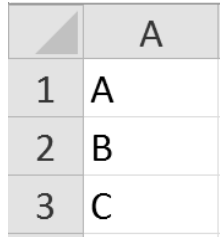
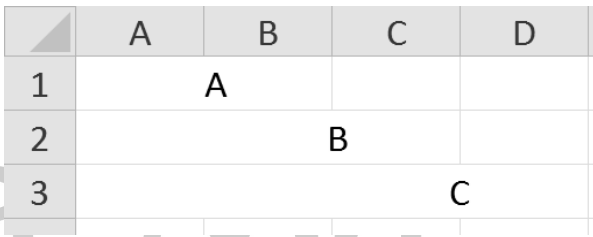
Código
<pre>Sub prueba_ID()   ActiveSheet.Range("A1").ID = "T_1"   ActiveSheet.Range("A2").ID = "T_2"   ActiveSheet.Range("A3").ID = "T_3" End Sub</pre>

Ra-Ma®

Resultado	
	<pre> &lt;body link=blue vlink=purple&gt;  &lt;table border=0 cellpadding=0 cellspacing=0 width=160 style='border-collapse: collapse;table-layout:fixed;width:120pt'&gt; &lt;col width=80 span=2 style='width:60pt'&gt; &lt;tr height=20 style='height:15.0pt'&gt; &lt;td height=20 width=80 style='height:15.0pt;width:60pt' id="T_1"&gt;A&lt;/td&gt; &lt;td align=right width=80 style='width:60pt'&gt;10&lt;/td&gt; &lt;/tr&gt; &lt;tr height=20 style='height:15.0pt'&gt; &lt;td height=20 style='height:15.0pt' id="T_2"&gt;B&lt;/td&gt; &lt;td align=right&gt;20&lt;/td&gt; &lt;/tr&gt; &lt;tr height=20 style='height:15.0pt'&gt; &lt;td height=20 style='height:15.0pt' id="T_3"&gt;C&lt;/td&gt; &lt;td align=right&gt;30&lt;/td&gt; &lt;/tr&gt; &lt;![if supportMisalignedColumns]&gt; &lt;tr height=0 style='display:none'&gt; &lt;td width=80 style='width:60pt'&gt;&lt;/td&gt; &lt;td width=80 style='width:60pt'&gt;&lt;/td&gt; &lt;/tr&gt; &lt;![endif]&gt; &lt;/table&gt;  &lt;/body&gt; </pre>

#### 2.4.1.42 INDENTLEVEL

Define o consulta el nivel de sangría del rango indicado.

Código	
<pre> Sub prueba_IndentLevel()     ActiveSheet.Range("A1").IndentLevel = 5     ActiveSheet.Range("A2").IndentLevel = 10     ActiveSheet.Range("A3").IndentLevel = 15 End Sub </pre>	
Resultado	
	
<b>Inicio</b>	<b>Fin</b>

2.4.1.43 INTERIOR

Proporciona un objeto de tipo Interior mediante el cual podemos modificar o consultar el interior de un rango.

Código																	
<pre>Sub prueba_Interior()   Range("A1").Interior.Color = 1000000000   Range("A2").Interior.Color = 2000000000   Range("A3").Interior.Color = 3000000000 End Sub</pre>																	
Resultado																	
<table><tr><td></td><td>A</td></tr><tr><td>1</td><td>A</td></tr><tr><td>2</td><td>B</td></tr><tr><td>3</td><td>C</td></tr></table>		A	1	A	2	B	3	C	<table><tr><td></td><td>A</td></tr><tr><td>1</td><td>A</td></tr><tr><td>2</td><td>B</td></tr><tr><td>3</td><td>C</td></tr></table>		A	1	A	2	B	3	C
	A																
1	A																
2	B																
3	C																
	A																
1	A																
2	B																
3	C																
Inicio	Fin																

2.4.1.44 ITEM

Devuelve un objeto Range que representa un rango desplazado del rango especificado.

Proporciona un objeto de tipo Range que contiene las celdas que se hallan a partir del desplazamiento indicado en forma de fila y/o columna. El primer parámetro de Item hace referencia a la fila y el segundo a la columna. La columna puede indicarse con un número o con una letra pero siempre relativa a la primera celda del rango indicado.

Código
<pre>Sub prueba_Item()   Dim r As Range   Set r = Range("B1:B3")   Debug.Print r.Item(1)   Debug.Print r.Item(2)   Debug.Print r.Item(2, 2)   Debug.Print r.Item(2, "C") End Sub</pre>

Resultado					
	A	B	C	D	
1	11	12	13	14	
2	21	22	23	24	
3	31	32	33	34	
4	41	42	43	44	
Inicio			Fin		

En este ejemplo puede observarse que Item(1) contiene el valor 12 y que Item(2,"C"), contiene el valor 24 ya que muestra el valor situado en la celda que se halla en la segunda fila del rango B1:B3 y en la tercera columna ("C" equivale a 3).

#### 2.4.1.45 LEFT

Proporciona la distancia en puntos existente entre el borde izquierdo de la columna A y el borde izquierdo del rango.

Código
<pre>Sub prueba_Left()     Debug.Print Range("C1:C3").Left End Sub</pre>
Resultado
<div>Inmediato</div> <div>75</div>

#### 2.4.1.46 LISTHEADERROWS

Indica el número de filas de encabezado que posee el rango indicado.

Código
<pre>Sub prueba_ListHeaderRows()     Debug.Print Range("C1:C5").ListHeaderRows End Sub</pre>

Resultado									
<table><tr><td>C</td></tr><tr><td>Total</td></tr><tr><td>Importe</td></tr><tr><td>10.000,00</td></tr><tr><td>40.000,00</td></tr><tr><td>90.000,00</td></tr></table>	C	Total	Importe	10.000,00	40.000,00	90.000,00	<table><tr><td>Inmediato</td></tr><tr><td>2</td></tr></table>	Inmediato	2
C									
Total									
Importe									
10.000,00									
40.000,00									
90.000,00									
Inmediato									
2									
Inicio	Fin								

#### 2.4.1.47 LISTOBJECT

Proporciona un objeto de tipo ListObject para el rango indicado.

Código																		
<pre>Sub prueba_ListObject()     Debug.Print Range("A1:B4").ListObject End Sub</pre>																		
Resultado																		
<table><tr><td></td><td>A</td><td>B</td></tr><tr><td>1</td><td>Provincia</td><td>Importe</td></tr><tr><td>2</td><td>Barcelona</td><td>1000</td></tr><tr><td>3</td><td>Tarragona</td><td>2000</td></tr><tr><td>4</td><td>Madrid</td><td>1500</td></tr></table>		A	B	1	Provincia	Importe	2	Barcelona	1000	3	Tarragona	2000	4	Madrid	1500	<table><tr><td>Inmediato</td></tr><tr><td>Ventas</td></tr></table>	Inmediato	Ventas
	A	B																
1	Provincia	Importe																
2	Barcelona	1000																
3	Tarragona	2000																
4	Madrid	1500																
Inmediato																		
Ventas																		
Inicio	Fin																	

#### 2.4.1.48 LOCATIONINTABLE

Proporciona un valor que permite identificar en qué parte de la tabla dinámica se halla la celda correspondiente a la esquina superior del rango indicado. El valor puede ser alguno de los representados por la constante `XlLocationInTable`:



- 4153 xlRowHeader
- 4110 xlColumnHeader
- 2 xlPageHeader
- 3 xlDataHeader
- 4 xlRowItem
- 5 xlColumnItem
- 6 xlPageItem
- 7 xlDataItem
- 8 xlTableBody

Código																											
<pre>Sub prueba_LocationInTable()   Debug.Print Range("E2").LocationInTable   Debug.Print Range("E3").LocationInTable   Debug.Print Range("F8").LocationInTable End Sub</pre>																											
Resultado																											
<table><tr><th>E</th><th>F</th></tr><tr><td colspan="2">Etiquetas de fila    Suma de Importe</td></tr><tr><td>Juan</td><td>1360</td></tr><tr><td>  Barcelona</td><td>80</td></tr><tr><td>  Londres</td><td>320</td></tr><tr><td>  Madrid</td><td>220</td></tr><tr><td>  Paris</td><td>740</td></tr><tr><td>Maria</td><td>1770</td></tr><tr><td>  Barcelona</td><td>150</td></tr><tr><td>  Londres</td><td>740</td></tr><tr><td>  Madrid</td><td>230</td></tr><tr><td>  Paris</td><td>650</td></tr><tr><td>Total general</td><td>3130</td></tr></table>	E	F	Etiquetas de fila    Suma de Importe		Juan	1360	Barcelona	80	Londres	320	Madrid	220	Paris	740	Maria	1770	Barcelona	150	Londres	740	Madrid	230	Paris	650	Total general	3130	<div>Inmediato</div> <div>-4153</div> <div>4</div> <div>8</div>
E	F																										
Etiquetas de fila    Suma de Importe																											
Juan	1360																										
Barcelona	80																										
Londres	320																										
Madrid	220																										
Paris	740																										
Maria	1770																										
Barcelona	150																										
Londres	740																										
Madrid	230																										
Paris	650																										
Total general	3130																										
Inicio	Fin																										

2.4.1.49 LOCKED

Define o consulta si el rango está bloqueado.

Código	
<pre>Sub prueba_Locked()     Debug.Print Range("A1").Locked End Sub</pre>	
Resultado	
	<div>Inmediato</div> <div>Verdadero</div>

2.4.1.50 MERGEAREA

Proporciona un rango que representa el rango combinado que a su vez contiene el rango indicado.

Código															
<pre>Sub prueba_MergeArea()     Debug.Print Range("A1").MergeArea.Address End Sub</pre>															
Resultado															
<table><tr><th></th><th>A</th><th>B</th><th>C</th></tr><tr><td>1</td><td colspan="3">ROTULO</td></tr><tr><td>2</td><td>10</td><td>20</td><td>30</td></tr></table>		A	B	C	1	ROTULO			2	10	20	30	<table><tr><td>Inmediato</td></tr><tr><td>\$A\$1:\$C\$1</td></tr></table>	Inmediato	\$A\$1:\$C\$1
	A	B	C												
1	ROTULO														
2	10	20	30												
Inmediato															
\$A\$1:\$C\$1															
Inicio	Fin														

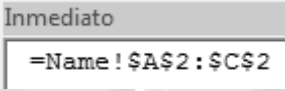
2.4.1.51 MERGECELLS

Define o consulta mediante un Boolean si el rango indicado contiene celdas combinadas.

Código															
<pre>Sub prueba_MergeCells()     Debug.Print Range("A1").MergeCells End Sub</pre>															
Resultado															
<table><tr><td></td><td>A</td><td>B</td><td>C</td></tr><tr><td>1</td><td colspan="3">ROTULO</td></tr><tr><td>2</td><td>10</td><td>20</td><td>30</td></tr></table>		A	B	C	1	ROTULO			2	10	20	30	<table><tr><td>Inmediato</td></tr><tr><td>Verdadero</td></tr></table>	Inmediato	Verdadero
	A	B	C												
1	ROTULO														
2	10	20	30												
Inmediato															
Verdadero															
Inicio	Fin														

### 2.4.1.52 NAME

Define o consulta el nombre del objeto.

Código	
<pre>Sub prueba_Name()     Debug.Print Range("A2:C2").Name End Sub</pre>	
Resultado	
	

### 2.4.1.53 NEXT

Proporciona un objeto de tipo Range que representa la siguiente celda del rango indicado.

Código	
<pre>Sub prueba_Next()     Dim r As Range     Set r = Range("A1")     Debug.Print r.Address     Set r = r.Next     Debug.Print r.Address     Set r = r.Next     Debug.Print r.Address End Sub</pre>	

Resultado												
<table><tr><th></th><th>A</th><th>B</th><th>C</th></tr><tr><th>1</th><td>20</td><td>20</td><td>30</td></tr></table>				A	B	C	1	20	20	30	<div>Inmediato</div> <div>\$A\$1</div> <div>\$B\$1</div> <div>\$C\$1</div>	
	A	B	C									
1	20	20	30									
Inicio			Fin									

2.4.1.54 NUMBERFORMAT

Define o consulta el código del formato del rango indicado.

Código									
<pre>Sub prueba_NumberFormat()   Debug.Print Range("A1").NumberFormat   Debug.Print Range("A2").NumberFormat   Debug.Print Range("A3").NumberFormat End Sub</pre>									
Resultado									
<table><tr><th></th><th>A</th></tr><tr><th>1</th><td>20</td></tr><tr><th>2</th><td>20,00</td></tr><tr><th>3</th><td>3.000</td></tr></table>		A	1	20	2	20,00	3	3.000	<div>Inmediato</div> <div>General</div> <div>0.00</div> <div>#,##0</div>
	A								
1	20								
2	20,00								
3	3.000								
Inicio	Fin								

2.4.1.55 NUMBERFORMATLOCAL

Define o consulta el código del formato en el idioma del usuario del rango indicado.

Código	
<pre>Sub prueba_NumberFormatLocal()   Debug.Print Range("A1").NumberFormatLocal   Debug.Print Range("A2").NumberFormatLocal   Debug.Print Range("A3").NumberFormatLocal End Sub</pre>	

Resultado									
<table border="1"> <thead> <tr> <th></th><th>A</th></tr> </thead> <tbody> <tr> <td>1</td><td>20</td></tr> <tr> <td>2</td><td>20,00</td></tr> <tr> <td>3</td><td>3.000</td></tr> </tbody> </table>		A	1	20	2	20,00	3	3.000	<div>Inmediato</div> <div>Estándar</div> <div>0,00</div> <div>###0</div>
	A								
1	20								
2	20,00								
3	3.000								
<b>Inicio</b>	<b>Fin</b>								

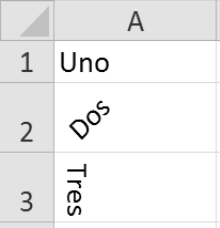
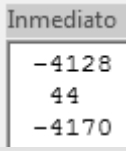
### 2.4.1.56 OFFSET

Proporciona un rango el cual está desplazado en una serie de columnas y filas con respecto al rango indicado.

Código																																																		
<pre>Sub prueba_Offset()   Dim r As Range   Set r = Range("A1:A2").Offset(2, 1)   Debug.Print r.Address   r.Borders.Color = 3000 End Sub</pre>																																																		
Resultado																																																		
<table><tr><th></th><th>A</th><th>B</th></tr><tr><td>1</td><td>Provincia</td><td>Vendedor</td></tr><tr><td>2</td><td>Barcelona</td><td>Juan</td></tr><tr><td>3</td><td>Barcelona</td><td>Maria</td></tr><tr><td>4</td><td>Madrid</td><td>Juan</td></tr><tr><td>5</td><td>Madrid</td><td>Maria</td></tr><tr><td>6</td><td>Londres</td><td>Juan</td></tr><tr><td>7</td><td>Paris</td><td>Juan</td></tr></table>		A	B	1	Provincia	Vendedor	2	Barcelona	Juan	3	Barcelona	Maria	4	Madrid	Juan	5	Madrid	Maria	6	Londres	Juan	7	Paris	Juan	<table><tr><th></th><th>A</th><th>B</th></tr><tr><td>1</td><td>Provincia</td><td>Vendedor</td></tr><tr><td>2</td><td>Barcelona</td><td>Juan</td></tr><tr><td>3</td><td>Barcelona</td><td>Maria</td></tr><tr><td>4</td><td>Madrid</td><td>Juan</td></tr><tr><td>5</td><td>Madrid</td><td>Maria</td></tr><tr><td>6</td><td>Londres</td><td>Juan</td></tr><tr><td>7</td><td>Paris</td><td>Juan</td></tr></table>		A	B	1	Provincia	Vendedor	2	Barcelona	Juan	3	Barcelona	Maria	4	Madrid	Juan	5	Madrid	Maria	6	Londres	Juan	7	Paris	Juan	<div>Inmediato</div> <div>\$B\$3:\$B\$4</div>
	A	B																																																
1	Provincia	Vendedor																																																
2	Barcelona	Juan																																																
3	Barcelona	Maria																																																
4	Madrid	Juan																																																
5	Madrid	Maria																																																
6	Londres	Juan																																																
7	Paris	Juan																																																
	A	B																																																
1	Provincia	Vendedor																																																
2	Barcelona	Juan																																																
3	Barcelona	Maria																																																
4	Madrid	Juan																																																
5	Madrid	Maria																																																
6	Londres	Juan																																																
7	Paris	Juan																																																
Inicio	Fin																																																	

2.4.1.57 ORIENTATION

Define o consulta el valor que indica la orientación del texto.

Código	
<pre>Sub prueba_Orientation()   Debug.Print Range("A1").Orientation   Debug.Print Range("A2").Orientation   Debug.Print Range("A3").Orientation End Sub</pre>	
Resultado	
	
Inicio	Fin

2.4.1.58 OUTLINELEVEL

Define o consulta el nivel de esquema de la fila o columna indicada.

Código
<pre>Sub prueba_OutlineLevel()   Debug.Print Range("A1:A12").Rows(1).OutlineLevel   Debug.Print Range("A1:A12").Rows(2).OutlineLevel End Sub</pre>

Resultado																																									
<table border="1"> <tr> <th></th><th>A</th><th>B</th></tr> <tr> <td>1</td><td><b>Provincia</b></td><td><b>Importe</b></td></tr> <tr> <td>2</td><td>Barcelona</td><td>80</td></tr> <tr> <td>3</td><td>Barcelona</td><td>150</td></tr> <tr> <td>4</td><td>Madrid</td><td>220</td></tr> <tr> <td>5</td><td>Madrid</td><td>230</td></tr> <tr> <td>6</td><td><b>España</b></td><td>680</td></tr> <tr> <td>7</td><td>Londres</td><td>320</td></tr> <tr> <td>8</td><td>Manchester</td><td>560</td></tr> <tr> <td>9</td><td><b>Reino Unido</b></td><td>880</td></tr> <tr> <td>10</td><td>Lyon</td><td>158</td></tr> <tr> <td>11</td><td>Paris</td><td>740</td></tr> <tr> <td>12</td><td><b>Francia</b></td><td>898</td></tr> </table>		A	B	1	<b>Provincia</b>	<b>Importe</b>	2	Barcelona	80	3	Barcelona	150	4	Madrid	220	5	Madrid	230	6	<b>España</b>	680	7	Londres	320	8	Manchester	560	9	<b>Reino Unido</b>	880	10	Lyon	158	11	Paris	740	12	<b>Francia</b>	898	<div>Inmediato</div> <div>1</div> <div>2</div>	
	A	B																																							
1	<b>Provincia</b>	<b>Importe</b>																																							
2	Barcelona	80																																							
3	Barcelona	150																																							
4	Madrid	220																																							
5	Madrid	230																																							
6	<b>España</b>	680																																							
7	Londres	320																																							
8	Manchester	560																																							
9	<b>Reino Unido</b>	880																																							
10	Lyon	158																																							
11	Paris	740																																							
12	<b>Francia</b>	898																																							
<b>Inicio</b>	<b>Fin</b>																																								

### 2.4.1.59 PAGEBREAK

Define o consulta la ubicación de un salto de página. Los valores pueden ser:

**xlPageBreakAutomatic**

**xlPageBreakManual**

**xlPageBreakNone**

Código
<pre>Sub prueba_PageBreak()     Range("A1:M3").Columns(3).PageBreak = xlPageBreakManual     Range("A1:M3").Rows(2).PageBreak = xlPageBreakManual End Sub</pre>

Ra-Ma<sup>®</sup>

Resultado															
Inicio		<div></div>	A	B	C	D	E	F	G	H	I	J	K	L	M
	1	A	1	a	9	A	1	a	9	A	1	a	9	A	
	2	B	2	b	8	B	2	b	8	B	2	b	8	B	
	3	C	3	c	7	C	3	c	7	C	3	c	7	C	
Fin		<div></div>	A	B	C	D	E	F	G	H	I	J	K	L	M
	1	A	1	a	9	A	1	a	9	A	1	a	9	A	
	2	B	2	b	8	B	2	b	8	B	2	b	8	B	
	3	C	3	c	7	C	3	c	7	C	3	c	7	C	

### 2.4.1.60 PARENT

Proporciona el objeto primario del rango indicado.

Código
<pre>Sub prueba_Parent()     Debug.Print Range("A1:A12").Parent.Name End Sub</pre>
Resultado
<div> <div>Inmediato</div> <div>MiHoja</div> </div>

### 2.4.1.61 PHONETIC

Proporciona un objeto de tipo Phonetic con información sobre una cadena de texto fonética de una celda.

### 2.4.1.62 PHONETICS


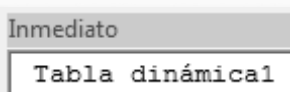
Proporciona una colección de objetos de tipo Phonetics del rango indicado.

Código
<pre>Sub prueba_Phonetics()     Dim objPhon As Phonetics     Dim objPhonItem As Phonetic     Set objPhon = Range("a1:a3").Phonetics     With objPhon         For Each objPhonItem In objPhon             MsgBox "Phonetic object: " &amp; .Text         Next     End With End Sub</pre>



### 2.4.1.63 PIVOTCELL

Proporciona un objeto de tipo PivotCell el cual representa una celda de una tabla dinámica.

Código	
<pre>Sub prueba_PivotCell()     Debug.Print Range("E4").PivotCell.Parent.Name End Sub</pre>	
Resultado	
	
<b>Inicio</b>	<b>Fin</b>

### 2.4.1.64 PIVOTFIELD

Proporciona un objeto de tipo PivotField que representa el campo de la tabla dinámica en la que se halla la celda de la esquina superior izquierda del rango indicado.

Código
<pre>Sub prueba_PivotField()     Debug.Print Range("E3").PivotField     Debug.Print Range("E4").PivotField     Debug.Print Range("F5").PivotField End Sub</pre>

Resultado																											
<table><tr><th>E</th><th>F</th></tr><tr><td colspan="2">Etiquetas de fila <input checked="" type="checkbox"/> Suma de Importe</td></tr><tr><td>Juan</td><td>1360</td></tr><tr><td>Barcelona</td><td>80</td></tr><tr><td>Londres</td><td>320</td></tr><tr><td>Madrid</td><td>220</td></tr><tr><td>Paris</td><td>740</td></tr><tr><td>Maria</td><td>1770</td></tr><tr><td>Barcelona</td><td>150</td></tr><tr><td>Londres</td><td>740</td></tr><tr><td>Madrid</td><td>230</td></tr><tr><td>Paris</td><td>650</td></tr><tr><td>Total general</td><td>3130</td></tr></table>	E	F	Etiquetas de fila <input checked="" type="checkbox"/> Suma de Importe		Juan	1360	Barcelona	80	Londres	320	Madrid	220	Paris	740	Maria	1770	Barcelona	150	Londres	740	Madrid	230	Paris	650	Total general	3130	<div>Inmediato</div> <div>Vendedor</div> <div>Provincia</div> <div>Suma de Importe</div>
E	F																										
Etiquetas de fila <input checked="" type="checkbox"/> Suma de Importe																											
Juan	1360																										
Barcelona	80																										
Londres	320																										
Madrid	220																										
Paris	740																										
Maria	1770																										
Barcelona	150																										
Londres	740																										
Madrid	230																										
Paris	650																										
Total general	3130																										
Inicio	Fin																										

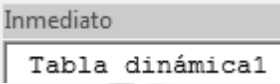
2.4.1.65 PIVOTITEM

Proporciona un objeto de tipo PivotItem que representa el elemento de la tabla dinámica en la que se halla la celda de la esquina superior izquierda del rango indicado.

Código																											
<pre>Sub prueba_PivotItem()     Debug.Print Range("E3").PivotItem     Debug.Print Range("E4").PivotItem End Sub</pre>																											
Resultado																											
<table><tr><th>E</th><th>F</th></tr><tr><td colspan="2">Etiquetas de fila <input checked="" type="checkbox"/> Suma de Importe</td></tr><tr><td>Juan</td><td>1360</td></tr><tr><td>Barcelona</td><td>80</td></tr><tr><td>Londres</td><td>320</td></tr><tr><td>Madrid</td><td>220</td></tr><tr><td>Paris</td><td>740</td></tr><tr><td>Maria</td><td>1770</td></tr><tr><td>Barcelona</td><td>150</td></tr><tr><td>Londres</td><td>740</td></tr><tr><td>Madrid</td><td>230</td></tr><tr><td>Paris</td><td>650</td></tr><tr><td>Total general</td><td>3130</td></tr></table>	E	F	Etiquetas de fila <input checked="" type="checkbox"/> Suma de Importe		Juan	1360	Barcelona	80	Londres	320	Madrid	220	Paris	740	Maria	1770	Barcelona	150	Londres	740	Madrid	230	Paris	650	Total general	3130	<div>Inmediato</div> <div>Juan</div> <div>Barcelona®</div>
E	F																										
Etiquetas de fila <input checked="" type="checkbox"/> Suma de Importe																											
Juan	1360																										
Barcelona	80																										
Londres	320																										
Madrid	220																										
Paris	740																										
Maria	1770																										
Barcelona	150																										
Londres	740																										
Madrid	230																										
Paris	650																										
Total general	3130																										
Inicio	Fin																										

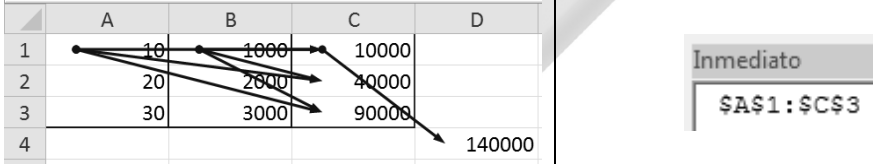
### 2.4.1.66 PIVOTTABLE

Devuelve un objeto de tipo PivotTable que representa la tabla dinámica en la que se halla la celda de la esquina superior izquierda del rango indicado.

Código
<pre>Sub prueba_PivotTable()     Debug.Print Range("E3").PivotTable End Sub</pre>
Resultado


### 2.4.1.67 PRECEDENTS

Proporciona un objeto de tipo Range con todas las celdas que preceden a una determinada celda.

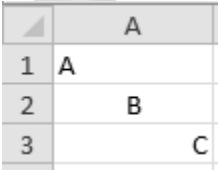

Código
<pre>Sub prueba_Precedents()     Debug.Print Range("D4").Precedents.Address End Sub</pre>
Resultado


En este ejemplo D4 posee la suma del rango C1:C3 que a su vez, contiene el resultado de cada una de las multiplicaciones de los valores de A1 por B1, A2 por B2 y A3 por B3 respectivamente.

2.4.1.68 PREFIXCHARACTER

Proporciona el carácter de prefijo de la celda que puede ser alguno de los siguientes si la propiedad TransitionNavigKeys está definida como True:

- ' rótulo justificado a la izquierda
- " rótulo justificado a la derecha
- ^ rótulo centrado
- \ rótulo repetido o un espacio en blanco.

Código	
<pre>Sub prueba_PrefixCharacter()   Application.TransitionNavigKeys = True   Debug.Print Range("A1").PrefixCharacter   Debug.Print Range("A2").PrefixCharacter   Debug.Print Range("A3").PrefixCharacter End Sub</pre>	
Resultado	
	
Inicio	Fin

2.4.1.69 PREVIOUS

Proporciona un objeto de tipo Range que representa la celda anterior al rango indicado.

Código
<pre>Sub prueba_Previous()   Debug.Print Range("B1").Previous End Sub</pre>

Resultado													
<table><tr><th></th><th>A</th><th>B</th></tr><tr><th>1</th><td>A</td><td>1</td></tr><tr><th>2</th><td>B</td><td>2</td></tr><tr><th>3</th><td>C</td><td>3</td></tr></table>		A	B	1	A	1	2	B	2	3	C	3	<div>Inmediato</div> <div>A</div>
	A	B											
1	A	1											
2	B	2											
3	C	3											
Inicio	Fin												

2.4.1.70 QUERYTABLE

Proporciona un objeto de tipo QueryTable que representa la tabla de consulta que forma una intersección con el rango indicado.

Código
<pre>Sub prueba_QueryTable()     Debug.Print Range("A1").QueryTables(1).Count     Debug.Print Range("A1").QueryTables(1).Name End Sub</pre>
Resultado
<div>Inmediato</div> <div>1 DatosExternos_1</div>

2.4.1.71 RANGE

Proporciona un objeto que representa un rango.

Código
<pre>Sub prueba_Range()     Dim t As Double     Dim r As Range     Set r = Range("A1:B3")     For Each r In Range("A1:B3")         Debug.Print "Celda : " &amp; r.Address &amp; " Valor : " &amp; r.Value         t = t + r.Value     Next     Debug.Print "Total : " &amp; t End Sub</pre>

Resultado																						
<table><tr><td></td><td>A</td><td>B</td></tr><tr><td>1</td><td>10</td><td></td></tr><tr><td>2</td><td>20</td><td>30</td></tr><tr><td>3</td><td></td><td>40</td></tr></table>			A	B	1	10		2	20	30	3		40	<table><tr><th>Inmediato</th></tr><tr><td>Celda : \$A\$1 Valor : 10</td></tr><tr><td>Celda : \$B\$1 Valor :</td></tr><tr><td>Celda : \$A\$2 Valor : 20</td></tr><tr><td>Celda : \$B\$2 Valor : 30</td></tr><tr><td>Celda : \$A\$3 Valor :</td></tr><tr><td>Celda : \$B\$3 Valor : 40</td></tr><tr><td>Total : 100</td></tr></table>	Inmediato	Celda : \$A\$1 Valor : 10	Celda : \$B\$1 Valor :	Celda : \$A\$2 Valor : 20	Celda : \$B\$2 Valor : 30	Celda : \$A\$3 Valor :	Celda : \$B\$3 Valor : 40	Total : 100
	A	B																				
1	10																					
2	20	30																				
3		40																				
Inmediato																						
Celda : \$A\$1 Valor : 10																						
Celda : \$B\$1 Valor :																						
Celda : \$A\$2 Valor : 20																						
Celda : \$B\$2 Valor : 30																						
Celda : \$A\$3 Valor :																						
Celda : \$B\$3 Valor : 40																						
Total : 100																						
Inicio		Fin																				

### 2.4.1.72 READINGORDER

Devuelve o establece el orden de lectura del objeto especificado. Puede ser una de las constantes siguientes: xlRTL (de derecha a izquierda), xlLTR (de izquierda a derecha) o xlContext.Long de lectura y escritura.

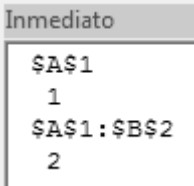
Define o consulta el orden de lectura de un control si el idioma con el que estamos trabajando lo permite.

Código
<pre>Sub prueba_ReadingOrder()     Dim r As Range     Set r = Range("A1:B3")     r.ReadingOrder = xlRTL End Sub</pre>

### 2.4.1.73 RESIZE

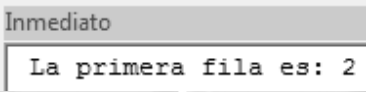
Proporciona un rango partiendo de un rabngo indicado al que se le ha modificado el número de columnas y/o filas.

Código
<pre>Sub prueba_Resize()     Dim r, f As Range     Set r = Range("A1")     r.Select     Debug.Print r.Address     Debug.Print r.Columns.Count     Set f = r.Resize(2, 2)     f.Select     Debug.Print f.Address     Debug.Print f.Columns.Count End Sub</pre>

Resultado


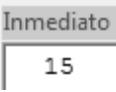
### 2.4.1.74 ROW

Indica el número de fila correspondiente a la primera columna del rango indicado.

Código
<pre>Sub prueba_Row()     Dim r As Range     Set r = Range("B2:C3")     Debug.Print "La primera fila es: " &amp; r.Row End Sub</pre>
Resultado


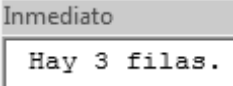
### 2.4.1.75 ROWHEIGHT

Proporciona el alto correspondiente a la suma de las filas del rango indicado.

Código
<pre>Sub prueba_RowHeight()     Debug.Print Range("A1:A3").RowHeight End Sub</pre>
Resultado


### 2.4.1.76 ROWS

Proporciona un objeto de tipo Range con las filas correspondientes al rango indicado.

Código
<pre>Sub prueba_Rows()     Debug.Print "Hay " &amp; Range("A1:A3").Rows.Count &amp; " filas." End Sub</pre>
Resultado


### 2.4.1.77 SERVERACTIONS

Indica las acciones que pueden realizarse en el servidor SharePoint con un rango.

### 2.4.1.78 SHOWDETAIL

Si True, se expande el rango asociado a la fila o columna resumen de un esquema.

Código	
<pre>Sub prueba_ShowDetail()     Range("A6").Rows.ShowDetail = True End Sub</pre>	
Resultado	
	
Inicio	Fin



### 2.4.1.79 SHRINKTOFIT

Define o consulta si un rango debe reducir el tamaño del texto para ajustarse al ancho de columna aplicado.

Código																
<pre>Sub prueba_ShrinkToFit()     Range("A1").ShrinkToFit = True End Sub</pre>																
Resultado																
<table><tr><th></th><th>A</th></tr><tr><td>1</td><td>PRIMERA FRASE</td></tr><tr><td>2</td><td>SEGUNDA FRASE</td></tr></table>		A	1	PRIMERA FRASE	2	SEGUNDA FRASE	<table><tr><th></th><th>A</th><th>B</th></tr><tr><td>1</td><td>PRIMERA FRASE</td><td></td></tr><tr><td>2</td><td>SEGUNDA FRASE</td><td></td></tr></table>		A	B	1	PRIMERA FRASE		2	SEGUNDA FRASE	
	A															
1	PRIMERA FRASE															
2	SEGUNDA FRASE															
	A	B														
1	PRIMERA FRASE															
2	SEGUNDA FRASE															
Inicio	Fin															

### 2.4.1.80 SOUNDNOTE

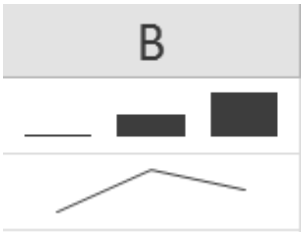

No se utiliza.

### 2.4.1.81 SPARKLINEGROUPS

Proporciona un objeto de tipo SparklineGroups con los diferentes minigráficos existentes en el rango indicado.

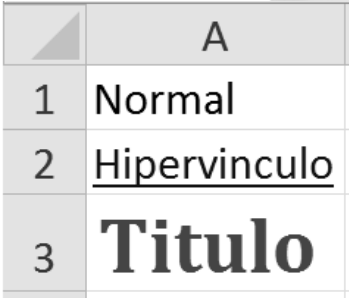
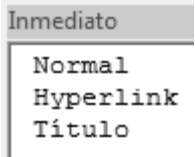
Código
<pre>Sub prueba_SparklineGroups()     Dim s As SparklineGroup     For Each s In Range("A1:B3").SparklineGroups         Debug.Print "Tipo : " &amp; s.Type     Next End Sub</pre>

Ra-Ma®

Resultado	
	
Inicio	Fin

2.4.1.82 STYLE

Devuelve o establece un valor de tipo Variant que contiene un objeto Style que representa el estilo del rango especificado.

Código	
<pre>Sub prueba_Style()     Debug.Print Range("A1").Style     Debug.Print Range("A2").Style     Debug.Print Range("A3").Style End Sub</pre>	
Resultado	
	
Inicio	Fin

2.4.1.83 SUMMARY

Define o consulta la descripción asociada al texto alternativo de la tabla indicada.

Código

```
Sub prueba_Summary()  
    Debug.Print Range("A1:B4").ListObject.Summary  
End Sub
```

Resultado

	A	B	C	D
1	Provincia	Importe		
2	Barcelona	1000		
3	Tarragona	2000		
4	Madrid	1500		
5				
6				
7				
8				
9				

Texto alternativo

Título

Ventas

Descripción

Descripción de pruebas de la tabla

Los títulos y las descripciones proporcionan otras representaciones basadas en texto de la información incluida en las tablas. Esta información es útil para personas con dificultades visuales o cognitivas que quizá no pueden ver o entender una tabla.  
Al leerles el título del contenido, pueden decidir si desean oír el resto de la descripción.

Aceptar

Cancelar

Inmediato

Descripción de pruebas

Inicio

Fin

2.4.1.84 TEXT

Proporciona el texto asociado al objeto indicado.

Código

```
Sub prueba_Text()  
    Debug.Print Range("A1").Text  
    Debug.Print Range("A1").Value  
End Sub
```

Resultado

	A	
1	=2+2	

Inmediato

=2+2  
4

Inicio

Fin

2.4.1.85 TOP

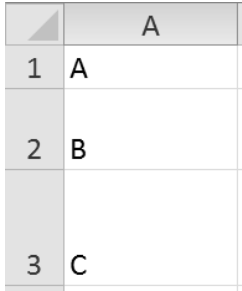
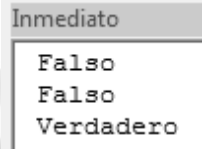
Proporciona la distancia en puntos existente entre el borde superior de la fila 1 hasta el borde superior del rango indicado.

Código													
<pre>Sub prueba_Top()   Debug.Print Range("A2").Top   Debug.Print Range("A3").Top   Debug.Print Range("A4").Top End Sub</pre>													
Resultado													
<table><tr><td></td><td>A</td></tr><tr><td>1</td><td>A</td></tr><tr><td>2</td><td>B</td></tr><tr><td>3</td><td>C</td></tr></table>		A	1	A	2	B	3	C	<table><tr><td>Inmediato</td></tr><tr><td>15</td></tr><tr><td>42,75</td></tr><tr><td>82,5</td></tr></table>	Inmediato	15	42,75	82,5
	A												
1	A												
2	B												
3	C												
Inmediato													
15													
42,75													
82,5													
Inicio	Fin												

2.4.1.86 USESTANDARDHEIGHT

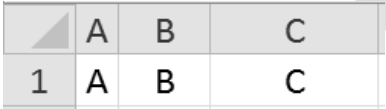
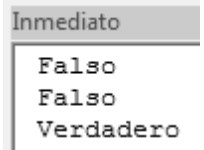
Indica mediante un Boolean si el alto del rango indicado es idéntico al alto estándar de la hoja. Si el rango tiene varias filas de diferente tamaño, devuelve null.

Código
<pre>Sub prueba_UseStandardHeight()   Debug.Print Range("A2").UseStandardHeight   Debug.Print Range("A3").UseStandardHeight   Debug.Print Range("A4").UseStandardHeight End Sub</pre>

Resultado	
	
<b>Inicio</b>	<b>Fin</b>

#### 2.4.1.87 USESTANDARDWIDTH

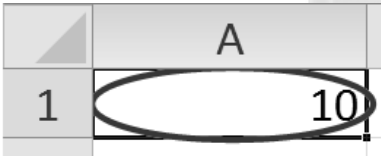
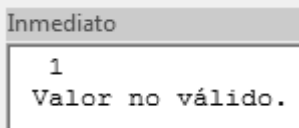
Indica mediante un Boolean si el ancho del rango indicado es idéntico al ancho estándar de la hoja. Si el rango tiene varias filas de diferente tamaño, devuelve null.

Código	
<pre>Sub prueba_UseStandardWidth()     Debug.Print Range("A1").UseStandardWidth     Debug.Print Range("B1").UseStandardWidth     Debug.Print Range("C1").UseStandardWidth End Sub</pre>	
Resultado	
	
<b>Inicio</b>	<b>Fin</b>

#### 2.4.1.88 VALIDATION

Proporciona un objeto de tipo Validation mediante el cual podemos averiguar que tipo de validación se halla aplicado al rango indicado. Los tipos de validación vienen representados por la constante XIDVType que puede tener alguno de los siguientes valores:

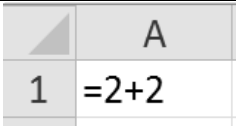
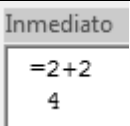
- xlValidateInputOnly 0
- xlValidateWholeNumber 1
- xlValidateDecimal 2
- xlValidateList 3
- xlValidateDate 4
- xlValidateTime 5
- xlValidateTextLength 6
- xlValidateCustom 7

Código	
<pre>Sub prueba_Validation()     Debug.Print Range("A1").Validation.Type     Debug.Print Range("A1").Validation.ErrorMessage End Sub</pre>	
Resultado	
	
Inicio	Fin

2.4.1.89 VALUE

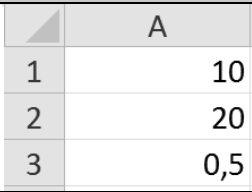
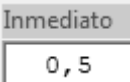
Proporciona o asigna el valor asociado al objeto indicado.

Código
<pre>Sub prueba_Value()     Debug.Print Range("A1").Text     Debug.Print Range("A1").Value End Sub</pre>

Resultado	
	
<b>Inicio</b>	<b>Fin</b>

#### 2.4.1.90 VALUE2

Permite obtener o asignar el valor de la celda indicada. La diferencia con respecto a Value es que Value2 no utiliza los tipos de datos Currency y Date

Código	
<pre>Sub prueba_Value2()     Range("A3").Value2 = Range("A1").Value2 / Range("A2").Value2     Debug.Print Range("A3").Value2 End Sub</pre>	
Resultado	
	
<b>Inicio</b>	<b>Fin</b>

#### 2.4.1.91 VERTICALALIGNMENT

Define o consulta el tipo de alineación vertical del rango indicado. El valor devuelto está representado por la conntante xlVAlign y puede ser alguno de los siguientes:

xlVAlignBottom -4107

xlVAlignCenter -4108

xlVAlignDistributed -4117

xlVAlignJustify -4130

xlVAlignTop -4160

Código																					
<pre>Sub prueba_VerticalAlignment()   Debug.Print Range("A1:A3").VerticalAlignment   Debug.Print Range("B1:B3").VerticalAlignment   Debug.Print Range("C1:C3").VerticalAlignment End Sub</pre>																					
Resultado																					
<table><tr><th></th><th>A</th><th>B</th><th>C</th></tr><tr><td>1</td><td>10</td><td>UNO</td><td>40</td></tr><tr><td>2</td><td>20</td><td>DOS</td><td>50</td></tr><tr><td>3</td><td>30</td><td>TRES</td><td>60</td></tr></table>		A	B	C	1	10	UNO	40	2	20	DOS	50	3	30	TRES	60	<table><tr><th>Inmediato</th></tr><tr><td>-4160</td></tr><tr><td>-4108</td></tr><tr><td>-4107</td></tr></table>	Inmediato	-4160	-4108	-4107
	A	B	C																		
1	10	UNO	40																		
2	20	DOS	50																		
3	30	TRES	60																		
Inmediato																					
-4160																					
-4108																					
-4107																					
Inicio	Fin																				

2.4.1.92 WIDTH

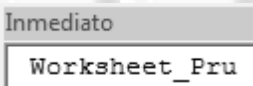
Devuelve un valor de tipo Variant que representa el ancho, en unidades, del rango.

Código													
<pre>Sub prueba_Width()     Debug.Print Range("A1").Width     Debug.Print Range("B1").Width     Debug.Print Range("C1").Width End Sub</pre>													
Resultado													
<table><tr><th></th><th>A</th><th>B</th><th>C</th></tr><tr><td>1</td><td>10</td><td>20</td><td>30</td></tr></table>		A	B	C	1	10	20	30	<table><tr><th>Inmediato</th></tr><tr><td>19,5</td></tr><tr><td>35,25</td></tr><tr><td>51</td></tr></table>	Inmediato	19,5	35,25	51
	A	B	C										
1	10	20	30										
Inmediato													
19,5													
35,25													
51													
Inicio	Fin												



### 2.4.1.93 WORKSHEET

Proporciona un objeto de tipo Worksheet que representa la hoja en la que se halla el rango indicado.

Código
<pre>Sub prueba_Worksheet()     Debug.Print Range("A1").Worksheet.Name End Sub</pre>
Resultado


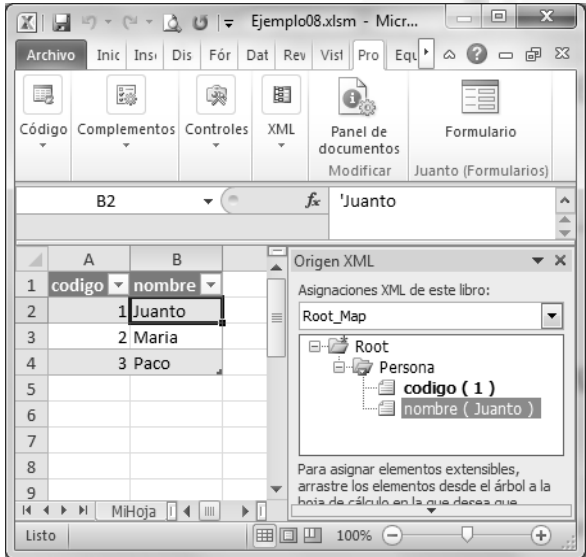
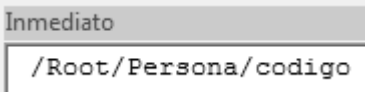
### 2.4.1.94 WRAPTEXT

Permite asignar o consultar el valor de tipo Boolean que indica si el texto debe ajustarse automáticamente según el ancho de la celda o no.

Código										
<pre>Sub prueba_WrapText()     Debug.Print Range("A1").WrapText     Debug.Print Range("A2").WrapText End Sub</pre>										
Resultado										
<table><tr><th></th><th>A</th><th>B</th></tr><tr><td>1</td><td>NO AJUSTA EL TEXTO</td><td></td></tr><tr><td>2</td><td>AJUSTA EL TEXTO</td><td></td></tr></table>		A	B	1	NO AJUSTA EL TEXTO		2	AJUSTA EL TEXTO		<div>Inmediato</div> <div>Falso</div> <div>Verdadero</div>
	A	B								
1	NO AJUSTA EL TEXTO									
2	AJUSTA EL TEXTO									
Inicio	Fin									

### 2.4.1.95 XPATH

Proporciona un objeto de tipo XPah que representa el XPath asociado al elemento indicado.

Código	
<pre>Sub prueba_XPath()     Debug.Print Range("A1").XPath.Value End Sub</pre>	
Resultado	
	
Inicio	Fin

2.4.2 Métodos

2.4.2.1 ACTIVATE

2.4.2.1.1 Descripción

Permite activar un rango determinado.

2.4.2.1.2 Sintaxis:

Activate

2.4.2.1.3 Argumentos

No tiene.

### 2.4.2.1.4 Ejemplo

Código																				
<pre>Sub prueba_Activate()     Range("B2:C3").Activate End Sub</pre>																				
Resultado																				
<table><tr><td></td><td>A</td><td>B</td><td>C</td></tr><tr><td>1</td><td>10</td><td>40</td><td>70</td></tr><tr><td>2</td><td>20</td><td>50</td><td>80</td></tr><tr><td>3</td><td>30</td><td>60</td><td>90</td></tr></table>						A	B	C	1	10	40	70	2	20	50	80	3	30	60	90
	A	B	C																	
1	10	40	70																	
2	20	50	80																	
3	30	60	90																	
<table><tr><td></td><td>A</td><td>B</td><td>C</td></tr><tr><td>1</td><td>10</td><td>40</td><td>70</td></tr><tr><td>2</td><td>20</td><td>50</td><td>80</td></tr><tr><td>3</td><td>30</td><td>60</td><td>90</td></tr></table>						A	B	C	1	10	40	70	2	20	50	80	3	30	60	90
	A	B	C																	
1	10	40	70																	
2	20	50	80																	
3	30	60	90																	
Inicio			Fin																	

## 2.4.2.2 ADDCOMMENT

### 2.4.2.2.1 Descripción

Agrega un comentario al rango indicado.

### 2.4.2.2.2 Sintaxis:

`AddComment (Text)`

### 2.4.2.2.3 Argumentos

Argumento	Comentario
Text	Texto a añadir como comentario.

### 2.4.2.2.4 Ejemplo

Código	
<pre>Sub prueba_AddComment()     Range("A1").AddComment "Mi comentario" End Sub</pre>	

Resultado					
	A	B	C	D	
1	10	Mi comentario			
2	20				
3	30				
4					

### 2.4.2.3 ADVANCEDFILTER

#### 2.4.2.3.1 Descripción

Filtra o copia datos de una lista en función de un rango de criterios. Si la selección inicial es una sola celda, se usará su región actual.

Permite realizar un filtro avanzado en base a un rango de criterios pudiendo dejar el resultado del filtro en un determinado lugar. Equivale a utilizar la opción de Avanzadas del grupo de funciones Ordenar y filtrar de la pestaña de Datos:

	A	B	C	D	E	F
1	Provincia	Importe		Importe		
2	Barcelona	80		>200		
3	Barcelona	150				
4	Madrid	230				
5	Londres	320				
6	Manchester	560				
7	Lyon	158				
8						

#### 2.4.2.3.2 Sintaxis:

**AdvancedFilter**(Action, CriteriaRange, CopyToRange, Unique)

Ra-Ma<sup>®</sup>

### 1.1.1.3.5 Argumentos

Argumento	Comentario
<b>Action</b>	Indica si los datos han de copiarse a una nueva ubicación o han de dejarse en el lugar donde están. Los valores pueden ser:  <b>xlFilterCopy</b> 2 Copia los datos filtrados en el lugar indicado (CopyToRange). <b>xlFilterInPlace</b> 1 Dejar los datos en el lugar donde están.
<b>CriteriaRange</b>	Rango que contiene los criterios a utilizar en el filtrado.
<b>CopyToRange</b>	Rango donde se dejarán los datos copiados si <b>Action</b> es <b>xlFilterCopy</b> .
<b>Unique</b>	Si indicamos True, se filtran sólo los registros únicos mostrando solo una ocurrencia en caso de que un mismo valor pueda aparecer varias veces.

### 2.4.2.3.3 Ejemplo

Código																																												
<pre>Sub prueba_AdvancedFilter()   Range("A1:B7").AdvancedFilter _     Action:=xlFilterCopy, _     CriteriaRange:=Range("D1:D2"), _     CopyToRange:=Range("A10") End Sub</pre>																																												
Resultado																																												
<table><tr><td></td><td>A</td><td>B</td><td>C</td><td>D</td></tr><tr><td>1</td><td>Provincia</td><td>Importe</td><td></td><td>Importe</td></tr><tr><td>2</td><td>Barcelona</td><td>80</td><td></td><td>&gt;200</td></tr><tr><td>3</td><td>Barcelona</td><td>150</td><td></td><td></td></tr><tr><td>4</td><td>Madrid</td><td>230</td><td></td><td></td></tr><tr><td>5</td><td>Londres</td><td>320</td><td></td><td></td></tr><tr><td>6</td><td>Manchester</td><td>560</td><td></td><td></td></tr><tr><td>7</td><td>Lyon</td><td>158</td><td></td><td></td></tr></table>						A	B	C	D	1	Provincia	Importe		Importe	2	Barcelona	80		>200	3	Barcelona	150			4	Madrid	230			5	Londres	320			6	Manchester	560			7	Lyon	158		
	A	B	C	D																																								
1	Provincia	Importe		Importe																																								
2	Barcelona	80		>200																																								
3	Barcelona	150																																										
4	Madrid	230																																										
5	Londres	320																																										
6	Manchester	560																																										
7	Lyon	158																																										
<table><tr><td></td><td>A</td><td>B</td></tr><tr><td>9</td><td></td><td></td></tr><tr><td>10</td><td>Provincia</td><td>Importe</td></tr><tr><td>11</td><td>Madrid</td><td>230</td></tr><tr><td>12</td><td>Londres</td><td>320</td></tr><tr><td>13</td><td>Manchester</td><td>560</td></tr></table>						A	B	9			10	Provincia	Importe	11	Madrid	230	12	Londres	320	13	Manchester	560																						
	A	B																																										
9																																												
10	Provincia	Importe																																										
11	Madrid	230																																										
12	Londres	320																																										
13	Manchester	560																																										
<table><tr><td>Inicio</td><td>Fin</td></tr></table>					Inicio	Fin																																						
Inicio	Fin																																											

En este ejemplo se muestra el resultado producido tras la ejecución del procedimiento.

## 2.4.2.4 ALLOCATECHANGES

### 2.4.2.4.1 Descripción

Permite realizar una escritura diferida de las celdas editadas de un rango basado en un origen de datos OLAP.

### 2.4.2.4.2 Sintaxis:

**AllocateChanges**

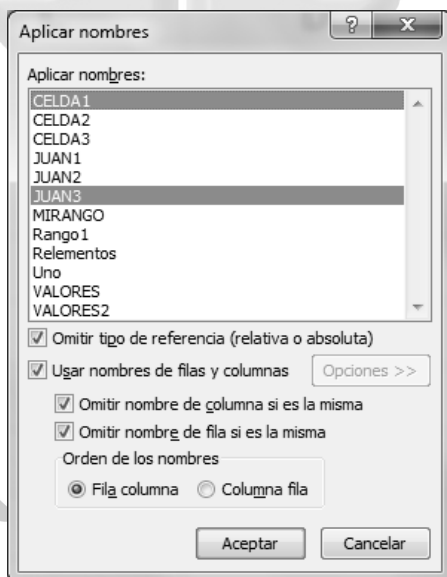
### 2.4.2.4.3 Argumentos

No tiene.

## 2.4.2.5 APPLYNAMES

### 2.4.2.5.1 Descripción

Permite aplicar los nombres indicados a las celdas contenidas en el rango que se indique. Equivale a utilizar la opción **Aplicar nombres...** que se halla en las opciones asociadas a **Asignar nombre** del grupo de controles **Nombres definidos** existente en la pestaña **Fórmulas**:



### 2.4.2.5.2 Sintaxis:

**ApplyNames** (**Names**, **IgnoreRelativeAbsolute**,  
**UseRowColumnNames**, **OmitColumn**, **OmitRow**, **Order**,  
**AppendLast**)

### 2.4.2.5.3 Argumentos

Argumento	Comentario
<b>Names</b>	Array con los nombres a aplicar. Si no se especifica, se aplicarán todos los nombres de la hoja.
<b>IgnoreRelativeAbsolute</b>	True indica que se han de sustituir los nombres sin tener en cuenta los tipos de referencia. False si tiene en cuenta los tipos de referencia.
<b>UseRowColumnNames</b>	Si se indica True, se utilizan los nombres de los rangos de filas y columnas que contienen el rango indicado si no se encuentran los nombres del rango. Si se indica False, se omiten los argumentos <b>OmitColumn</b> y <b>OmitRow</b> . Por defecto es True.
<b>OmitColumn</b>	Si True reemplaza la referencia completa por el nombre con orientación de fila. El nombre con orientación de columna sólo puede omitirse si la celda referenciada está en la misma columna que la fórmula y se encuentra dentro de un rango con nombre orientado a fila. Por defecto es True.
<b>OmitRow</b>	Si True reemplaza la referencia completa por el nombre con orientación de columna. El nombre con orientación de fila sólo puede omitirse si la celda referenciada está en la misma fila que la fórmula y se encuentra dentro de un rango con nombre orientado a columna. Por defecto es True.

Ra-Ma®

<b>Order</b>	<p>Permite determinar qué nombre de rango se ha de enumerar en primer lugar cuando una referencia es reemplazada por un nombre de rango con orientación de fila y de columna. El valor de la constante XlApplyNamesOrder puede tener alguno de los siguientes valores:</p> <p>xlColumnThenRow 2 Las columnas se enumeran en primer lugar.</p> <p>xlRowThenColumn 1 Las filas se enumeran en primer lugar.</p>
<b>AppendLast</b>	<p>Si True, reemplaza las definiciones de nombres de Names y las definiciones de los últimos nombres que se definieron. Si False, sólo se reemplazan las definiciones de Names.</p>

2.4.2.5.4 Ejemplo

Código																									
<pre>Sub prueba_ApplyNames()     Dim s As Variant     s = Array("CELDA1", "CELDA3")     Range("A5").ApplyNames (s) End Sub</pre>																									
Resultado																									
<table><tr><th></th><th>A</th></tr><tr><td>1</td><td>10</td></tr><tr><td>2</td><td>20</td></tr><tr><td>3</td><td>30</td></tr><tr><td>4</td><td></td></tr><tr><td>5</td><td>=A1+A2+A3</td></tr></table>		A	1	10	2	20	3	30	4		5	=A1+A2+A3	<table><tr><th></th><th>A</th></tr><tr><td>1</td><td>10</td></tr><tr><td>2</td><td>20</td></tr><tr><td>3</td><td>30</td></tr><tr><td>4</td><td></td></tr><tr><td>5</td><td>=CELDA1+A2+CELDA3</td></tr></table>		A	1	10	2	20	3	30	4		5	=CELDA1+A2+CELDA3
	A																								
1	10																								
2	20																								
3	30																								
4																									
5	=A1+A2+A3																								
	A																								
1	10																								
2	20																								
3	30																								
4																									
5	=CELDA1+A2+CELDA3																								
Inicio	Fin																								



En este ejemplo, tenemos que la celda A1 tiene el nombre CELDA1, la celda A2 se llama CELDA2 y la celda A3 se llama CELDA3. Tras la ejecución del procedimiento, vemos que la fórmula que se halla en la celda A5 ha sustituido las referencias a A1 y A3 por sus respectivos nombres CELDA1 y CELDA3.

## 2.4.2.6 APPLYOUTLINESTYLES

### 2.4.2.6.1 Descripción

Permite aplicar estilos de esquema al rango indicado.



### 2.4.2.6.2 Sintaxis:

ApplyOutlineStyles

### 2.4.2.6.3 Argumentos

No tiene

### 2.4.2.6.4 Ejemplo

Código		
<pre>Sub prueba_ApplyOutlineStyles()   ' Muestra el nivel 1   ActiveSheet.Outline.ShowLevels RowLevels:=1   Range("A1:B12").CurrentRegion.Select   ' Aplica estilos   Selection.ApplyOutlineStyles End Sub</pre>		
Resultado		
		
<b>Inicio</b>	<b>Fin</b>	

Podemos observar que tras la ejecución del procedimiento, el esquema se contrae para mostrar el nivel 1 y además, los subtotales por país aparecen en negrita.

2.4.2.7 AUTOCOMPLETE

2.4.2.7.1 Descripción

Proporciona una coincidencia a partir del texto indicado. Si no se halla ninguna coincidencia o se hallan varias, se devuelve un string vacío.

2.4.2.7.2 Sintaxis:

AutoComplete (cadena)

2.4.2.7.3 Argumentos

Argumento	Comentario
cadena	Cadena de texto sobre la que se desea buscar una coincidencia

2.4.2.7.4 Ejemplo

Código																																																																																			
<pre>Sub prueba_AutoComplete() Dim res As Variant ' Busca coincidencias en la columna A res = ActiveSheet.Range("A1").AutoComplete(Range("B8").Value) Range("A9").Value = res ' Busca coincidencias en la columna B res = ActiveSheet.Range("C1").AutoComplete(Range("B8").Value) Range("C9").Value = res End Sub</pre>																																																																																			
Resultado																																																																																			
<table><tr><th></th><th>A</th><th>B</th><th>C</th></tr><tr><td>1</td><td>Lunes</td><td></td><td>Uno</td></tr><tr><td>2</td><td>Martes</td><td></td><td>Dos</td></tr><tr><td>3</td><td>Miércoles</td><td></td><td>Tres</td></tr><tr><td>4</td><td>Jueves</td><td></td><td>Cuatro</td></tr><tr><td>5</td><td>Viernes</td><td></td><td>Cinco</td></tr><tr><td>6</td><td>Sábado</td><td></td><td>Seis</td></tr><tr><td>7</td><td>Domingo</td><td></td><td>Siete</td></tr><tr><td>8</td><td></td><td>Do</td><td></td></tr><tr><td>9</td><td></td><td></td><td></td></tr></table>			A	B	C	1	Lunes		Uno	2	Martes		Dos	3	Miércoles		Tres	4	Jueves		Cuatro	5	Viernes		Cinco	6	Sábado		Seis	7	Domingo		Siete	8		Do		9				<table><tr><th></th><th>A</th><th>B</th><th>C</th></tr><tr><td>1</td><td>Lunes</td><td></td><td>Uno</td></tr><tr><td>2</td><td>Martes</td><td></td><td>Dos</td></tr><tr><td>3</td><td>Miércoles</td><td></td><td>Tres</td></tr><tr><td>4</td><td>Jueves</td><td></td><td>Cuatro</td></tr><tr><td>5</td><td>Viernes</td><td></td><td>Cinco</td></tr><tr><td>6</td><td>Sábado</td><td></td><td>Seis</td></tr><tr><td>7</td><td>Domingo</td><td></td><td>Siete</td></tr><tr><td>8</td><td></td><td>Do</td><td></td></tr><tr><td>9</td><td>Domingo</td><td></td><td>Dos</td></tr></table>			A	B	C	1	Lunes		Uno	2	Martes		Dos	3	Miércoles		Tres	4	Jueves		Cuatro	5	Viernes		Cinco	6	Sábado		Seis	7	Domingo		Siete	8		Do		9	Domingo		Dos
	A	B	C																																																																																
1	Lunes		Uno																																																																																
2	Martes		Dos																																																																																
3	Miércoles		Tres																																																																																
4	Jueves		Cuatro																																																																																
5	Viernes		Cinco																																																																																
6	Sábado		Seis																																																																																
7	Domingo		Siete																																																																																
8		Do																																																																																	
9																																																																																			
	A	B	C																																																																																
1	Lunes		Uno																																																																																
2	Martes		Dos																																																																																
3	Miércoles		Tres																																																																																
4	Jueves		Cuatro																																																																																
5	Viernes		Cinco																																																																																
6	Sábado		Seis																																																																																
7	Domingo		Siete																																																																																
8		Do																																																																																	
9	Domingo		Dos																																																																																
Inicio		Fin																																																																																	

Si introducimos un texto en la celda B8 (en el ejemplo ‘Do’) y ejecutamos el procedimiento, observaremos que en la celda A9 se colocará el texto ‘Domingo’ porque Autocompletar ha buscado en la columna A, y en la celda C9 se colocará el texto ‘Dos’ porque Autocompletar ha buscado en la columna C.

## 2.4.2.8 AUTOFILL

### 2.4.2.8.1 Descripción

Rellena automáticamente las celdas del rango especificado.

### 2.4.2.8.2 Sintaxis:

AutoFill (Destino,Tipo)

### 2.4.2.8.3 Argumentos

Argumento	Comentario		
<b>Destino</b>	Variable de tipo rango que indica las celdas que han de rellenarse. Este rango debe contener el rango que contiene los valores sobre los que se basará el relleno automático.		
<b>Tipo</b>	Puede ser alguno de los valores indicados por la constante <b>XlAutoFillType</b> :		
	<b>Valor</b>	<b>Nombre</b>	<b>Comentario</b>
	<b>0</b>	<b>xlFillDefault</b>	Se utilizan los valores y formatos por defecto para rellenar el rango de destino.
	<b>1</b>	<b>xlFillCopy</b>	Copiar valores y formatos desde origen hacia destino, repitiéndolos hasta completar el destino.
	<b>2</b>	<b>xlFillSeries</b>	Rellena destino extendiendo la serie indicada en origen.
	<b>3</b>	<b>xlFillFormats</b>	Copiar sólo formatos del origen al destino.
	<b>4</b>	<b>xlFillValues</b>	Copiar sólo los valores del origen en el destino.

	<b>5</b>	<b>xlFillDays</b>	Rellena destino con los nombres de los días de la semana.
	<b>6</b>	<b>xlFillWeekdays</b>	Rellena destino con los nombres de los días laborables de la semana.
	<b>7</b>	<b>xlFillMonths</b>	Rellena destino con los nombres de los meses.
	<b>8</b>	<b>xlFillYears</b>	Rellena destino extendiendo los años del rango de origen.
	<b>9</b>	<b>xlLinearTrend</b>	Rellena destino suponiendo que cada valor es el resultado de sumar el valor anterior más un valor constante.
	<b>10</b>	<b>xlGrowthTrend</b>	Rellena destino suponiendo que cada valor es el resultado de multiplicar el valor anterior por un valor determinado.

#### 2.4.2.8.4 Ejemplo

Código
<pre> Sub prueba_AutoFill()     Dim origen, destino As Range     Set origen = ActiveSheet.Range("A1:A3")     Set destino = ActiveSheet.Range("A1:A9")     origen.AutoFill Destination:=destino     Set origen = ActiveSheet.Range("B1:B1")     Set destino = ActiveSheet.Range("B1:B7")     origen.AutoFill Destination:=destino, Type:=xlFillDays     Set origen = ActiveSheet.Range("C1:C2")     Set destino = ActiveSheet.Range("C1:C9")     origen.AutoFill Destination:=destino, Type:=xlFillDefault     Set origen = ActiveSheet.Range("D1:D1")     Set destino = ActiveSheet.Range("D1:D9")     origen.AutoFill Destination:=destino, Type:=xlFillMonths     Set origen = ActiveSheet.Range("E1:E2")     Set destino = ActiveSheet.Range("E1:E9")     origen.AutoFill Destination:=destino, Type:=xlGrowthTrend     Set origen = ActiveSheet.Range("F1:F2")     Set destino = ActiveSheet.Range("F1:F9")     origen.AutoFill Destination:=destino, Type:=xlLinearTrend End Sub </pre>



<b>Operator</b>	Uno de		
	<b>Valor</b>	<b>Nombre</b>	<b>Comentario</b>
	1	xlAnd	Realiza un AND entre el criterio1 y el criterio2.
	2	xlOr	Realiza un OR entre el criterio1 y el criterio2.
	3	xlTop10Items	Muestra el número de elementos de mayor valor. Dicho número será el indicado en el criterio1.
	4	xlBottom10Items	Muestra el número de elementos de menor valor. Dicho número será el indicado en el criterio1.
	5	xlTop10Percent	Se muestran los elementos de mayor valor (el porcentaje especificado en Criterio 1).  Muestra el número de elementos que superan el % indicado en el criterio1.
	6	xlBottom10Percent	Muestra el número de elementos que NO superan el % indicado en el criterio1.
	7	xlFilterValues	Valores de filtro.
	8	xlFilterCellColor	Color de la celda.
	9	xlFilterFontColor	Color de la fuente.
	10	xlFilterIcon	Icono de filtro.
	11	xlFilterDynamic	Filtro dinámico.
<b>Criteria2</b>	Segundo criterio a utilizar (si procede).		
<b>VisibleDropDown</b>	Si True, se muestra la flecha que indica que hay un filtro aplicados. Es el valor por defecto. False no muestra dicha flecha.		

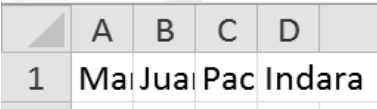
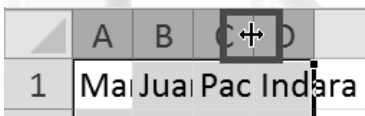
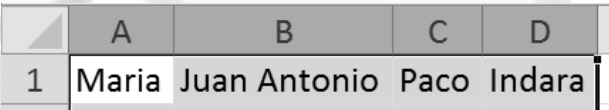
### 2.4.2.9.4 Ejemplo

Código																																																																														
Ejemplo 1	<pre>Sub prueba_AutoFilter() Dim r As Range Set r = ActiveSheet.Range("A1") r.AutoFilter field:=2, _ Criteria1:="J??? A*", _ VisibleDropDown:=True End Sub</pre>																																																																													
Ejemplo 2	<pre>Sub prueba_AutoFilter2() Dim r As Range Set r = ActiveSheet.Range("A1") r.AutoFilter field:=4, _ Criteria1:="&gt;20", _ Operator:=xlAnd, _ Criteria2:="&lt;60", _ VisibleDropDown:=True End Sub</pre>																																																																													
Resultado																																																																														
Original	<table><tr><th></th><th>A</th><th>B</th><th>C</th><th>D</th><th>E</th><th>F</th></tr><tr><th>1</th><th>Código</th><th>Nombre</th><th>Código</th><th>Valor</th><th>Código</th><th>Valor</th></tr><tr><td>2</td><td>1</td><td>Ana</td><td>1</td><td>10</td><td>1</td><td>10</td></tr><tr><td>3</td><td>2</td><td>Maria</td><td>2</td><td>20</td><td>2</td><td>20</td></tr><tr><td>4</td><td>3</td><td>Luis</td><td>3</td><td>30</td><td>3</td><td>30</td></tr><tr><td>5</td><td>4</td><td>Juan Antonio</td><td>4</td><td>40</td><td>4</td><td>40</td></tr><tr><td>6</td><td>5</td><td>Jose Luis</td><td>5</td><td>50</td><td>5</td><td>50</td></tr><tr><td>7</td><td>6</td><td>Pedro</td><td>6</td><td>60</td><td>6</td><td>60</td></tr><tr><td>8</td><td>7</td><td>Jose Antonio</td><td>7</td><td>70</td><td>7</td><td>70</td></tr><tr><td>9</td><td>8</td><td>Jordi</td><td>8</td><td>80</td><td>8</td><td>80</td></tr><tr><td>10</td><td>9</td><td>Xavi</td><td>9</td><td>90</td><td>9</td><td>90</td></tr></table>		A	B	C	D	E	F	1	Código	Nombre	Código	Valor	Código	Valor	2	1	Ana	1	10	1	10	3	2	Maria	2	20	2	20	4	3	Luis	3	30	3	30	5	4	Juan Antonio	4	40	4	40	6	5	Jose Luis	5	50	5	50	7	6	Pedro	6	60	6	60	8	7	Jose Antonio	7	70	7	70	9	8	Jordi	8	80	8	80	10	9	Xavi	9	90	9	90
		A	B	C	D	E	F																																																																							
	1	Código	Nombre	Código	Valor	Código	Valor																																																																							
	2	1	Ana	1	10	1	10																																																																							
	3	2	Maria	2	20	2	20																																																																							
	4	3	Luis	3	30	3	30																																																																							
	5	4	Juan Antonio	4	40	4	40																																																																							
	6	5	Jose Luis	5	50	5	50																																																																							
	7	6	Pedro	6	60	6	60																																																																							
	8	7	Jose Antonio	7	70	7	70																																																																							
	9	8	Jordi	8	80	8	80																																																																							
10	9	Xavi	9	90	9	90																																																																								
Ejemplo 1	<table><tr><th></th><th>A</th><th>B</th><th>C</th><th>D</th><th>E</th><th>F</th></tr><tr><th>1</th><th>Códig</th><th>Nombre</th><th>Códig</th><th>Valo</th><th>Códig</th><th>Valo</th></tr><tr><td>5</td><td>4</td><td>Juan Antonio</td><td>4</td><td>40</td><td>4</td><td>40</td></tr><tr><td>8</td><td>7</td><td>Jose Antonio</td><td>7</td><td>70</td><td>7</td><td>70</td></tr></table>		A	B	C	D	E	F	1	Códig	Nombre	Códig	Valo	Códig	Valo	5	4	Juan Antonio	4	40	4	40	8	7	Jose Antonio	7	70	7	70																																																	
	A	B	C	D	E	F																																																																								
1	Códig	Nombre	Códig	Valo	Códig	Valo																																																																								
5	4	Juan Antonio	4	40	4	40																																																																								
8	7	Jose Antonio	7	70	7	70																																																																								
Ejemplo 2	<table><tr><th></th><th>A</th><th>B</th><th>C</th><th>D</th><th>E</th><th>F</th></tr><tr><th>1</th><th>Códig</th><th>Nombre</th><th>Códig</th><th>Valo</th><th>Códig</th><th>Valo</th></tr><tr><td>4</td><td>3</td><td>Luis</td><td>3</td><td>30</td><td>3</td><td>30</td></tr><tr><td>5</td><td>4</td><td>Juan Antonio</td><td>4</td><td>40</td><td>4</td><td>40</td></tr><tr><td>6</td><td>5</td><td>Jose Luis</td><td>5</td><td>50</td><td>5</td><td>50</td></tr></table>		A	B	C	D	E	F	1	Códig	Nombre	Códig	Valo	Códig	Valo	4	3	Luis	3	30	3	30	5	4	Juan Antonio	4	40	4	40	6	5	Jose Luis	5	50	5	50																																										
	A	B	C	D	E	F																																																																								
1	Códig	Nombre	Códig	Valo	Códig	Valo																																																																								
4	3	Luis	3	30	3	30																																																																								
5	4	Juan Antonio	4	40	4	40																																																																								
6	5	Jose Luis	5	50	5	50																																																																								

## 2.4.2.10 AUTOFIT

### 2.4.2.10.1 Descripción

Permite ajustar el tamaño de las filas o columnas automáticamente. Equivale a seleccionar el rango que se desea ajustar manualmente y hacer doble clic con el puntero del mouse en la separación entre columnas o filas.

Columnas a autoajustar.	
Selección de las columnas y doble clic entre separación.	
Resultado del autoajuste.	

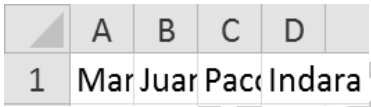
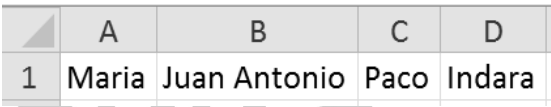
### 2.4.2.10.2 Sintaxis:

AutoFit (par)

### 2.4.2.10.3 Argumentos

No tiene

### 2.4.2.10.4 Ejemplo

Código	
<pre>Sub prueba_AutoFit()     ActiveSheet.Columns("A:D").AutoFit End Sub</pre>	
Resultado	
	
<b>Inicio</b>	<b>Fin</b>



## 2.4.2.11 AUTOOUTLINE

### 2.4.2.11.1 Descripción

Permite crear automáticamente un esquema para el rango especificado reemplazando el rango que ya pudiera existir.

### 2.4.2.11.2 Sintaxis:

AutoOutline

### 2.4.2.11.3 Argumentos

No tiene.

### 2.4.2.11.4 Ejemplo

Código																																									
<pre>Sub prueba_AutoOutline()     Dim r As Range     Set r = Range("A1:B12")     r.AutoOutline End Sub</pre>																																									
Resultado																																									
<table border="1"> <thead> <tr> <th></th><th>A</th><th>B</th></tr> </thead> <tbody> <tr><td>1</td><td>Provincia</td><td>Importe</td></tr> <tr><td>2</td><td>Barcelona</td><td>80</td></tr> <tr><td>3</td><td>Barcelona</td><td>150</td></tr> <tr><td>4</td><td>Madrid</td><td>220</td></tr> <tr><td>5</td><td>Madrid</td><td>230</td></tr> <tr><td>6</td><td>España</td><td>680</td></tr> <tr><td>7</td><td>Londres</td><td>320</td></tr> <tr><td>8</td><td>Manchester</td><td>560</td></tr> <tr><td>9</td><td>Inglaterra</td><td>880</td></tr> <tr><td>10</td><td>Lyon</td><td>158</td></tr> <tr><td>11</td><td>Paris</td><td>740</td></tr> <tr><td>12</td><td>Francia</td><td>898</td></tr> </tbody> </table>				A	B	1	Provincia	Importe	2	Barcelona	80	3	Barcelona	150	4	Madrid	220	5	Madrid	230	6	España	680	7	Londres	320	8	Manchester	560	9	Inglaterra	880	10	Lyon	158	11	Paris	740	12	Francia	898
	A	B																																							
1	Provincia	Importe																																							
2	Barcelona	80																																							
3	Barcelona	150																																							
4	Madrid	220																																							
5	Madrid	230																																							
6	España	680																																							
7	Londres	320																																							
8	Manchester	560																																							
9	Inglaterra	880																																							
10	Lyon	158																																							
11	Paris	740																																							
12	Francia	898																																							
<table border="1"> <thead> <tr> <th></th><th>A</th><th>B</th></tr> </thead> <tbody> <tr><td>1</td><td>Provincia</td><td>Importe</td></tr> <tr><td>2</td><td>Barcelona</td><td>80</td></tr> <tr><td>3</td><td>Barcelona</td><td>150</td></tr> <tr><td>4</td><td>Madrid</td><td>220</td></tr> <tr><td>5</td><td>Madrid</td><td>230</td></tr> <tr><td>6</td><td>España</td><td>680</td></tr> <tr><td>7</td><td>Londres</td><td>320</td></tr> <tr><td>8</td><td>Manchester</td><td>560</td></tr> <tr><td>9</td><td>Inglaterra</td><td>880</td></tr> <tr><td>10</td><td>Lyon</td><td>158</td></tr> <tr><td>11</td><td>Paris</td><td>740</td></tr> <tr><td>12</td><td>Francia</td><td>898</td></tr> </tbody> </table>				A	B	1	Provincia	Importe	2	Barcelona	80	3	Barcelona	150	4	Madrid	220	5	Madrid	230	6	España	680	7	Londres	320	8	Manchester	560	9	Inglaterra	880	10	Lyon	158	11	Paris	740	12	Francia	898
	A	B																																							
1	Provincia	Importe																																							
2	Barcelona	80																																							
3	Barcelona	150																																							
4	Madrid	220																																							
5	Madrid	230																																							
6	España	680																																							
7	Londres	320																																							
8	Manchester	560																																							
9	Inglaterra	880																																							
10	Lyon	158																																							
11	Paris	740																																							
12	Francia	898																																							
Inicio		Fin																																							

## 2.4.2.12 BORDERAROUND

### 2.4.2.12.1 Descripción

Permite incluir un borde a un rango definiendo características de color, estilo de línea y grosor del mismo.

### 2.4.2.12.2 Sintaxis:

**BorderAround**(LineStyle, Weight, ColorIndex, Color, ThemeColor)

### 2.4.2.12.3 Argumentos

Argumento	Comentario		
<b>LineStyle</b>	Estilo de la línea para el borde. Puede tener alguno de los valores de la constante <b>XlLineStyle</b> :		
	<b>Valor</b>	<b>Nombre</b>	<b>Comentario</b>
	1	<b>xlContinuous</b>	Línea continua.
	4	<b>xlDashDot</b>	Alterna guiones y puntos.
	5	<b>xlDashDotDot</b>	Guión seguido de dos puntos.
	13	<b>xlSlantDashDot</b>	Guiones inclinados.
	-4115	<b>xlDash</b>	Línea con guiones.
	-4118	<b>xlDot</b>	Línea de puntos.
	-4119	<b>xlDouble</b>	Línea doble.
	-4142	<b>xlLineStyleNone</b>	Sin línea.

Ra-Ma®

<b>Weight</b>	<p>Grosor del borde que puede ser alguno de los valores de la constante <b>XlBorderWeight</b>:</p> <table><tr><th>Valor</th><th>Nombre</th><th>Comentario</th></tr><tr><td>1</td><td><b>xlHairline</b></td><td>Fino.</td></tr><tr><td>2</td><td><b>xlThin</b></td><td>Delgado.</td></tr><tr><td>4</td><td><b>xlThick</b></td><td>Grueso.</td></tr><tr><td>-4138</td><td><b>xlMedium</b></td><td>Medio.</td></tr></table>	Valor	Nombre	Comentario	1	<b>xlHairline</b>	Fino.	2	<b>xlThin</b>	Delgado.	4	<b>xlThick</b>	Grueso.	-4138	<b>xlMedium</b>	Medio.											
Valor	Nombre	Comentario																									
1	<b>xlHairline</b>	Fino.																									
2	<b>xlThin</b>	Delgado.																									
4	<b>xlThick</b>	Grueso.																									
-4138	<b>xlMedium</b>	Medio.																									
<b>ColorIndex</b>	<p>Valor que indica el color del borde indicado como un valor de la paleta de colores o como un valor de la constante <b>XlColorIndex</b>.</p> <table><tr><th>Valor</th><th>Nombre</th><th>Comentario</th></tr><tr><td>-4105</td><td><b>xlColorIndexAutomatic</b></td><td>Color automático.</td></tr><tr><td>-4142</td><td><b>xlColorIndexNone</b></td><td>Sin color.</td></tr></table>	Valor	Nombre	Comentario	-4105	<b>xlColorIndexAutomatic</b>	Color automático.	-4142	<b>xlColorIndexNone</b>	Sin color.																	
Valor	Nombre	Comentario																									
-4105	<b>xlColorIndexAutomatic</b>	Color automático.																									
-4142	<b>xlColorIndexNone</b>	Sin color.																									
<b>Color</b>	Color del borde especificado como RGB.																										
<b>ThemeColor</b>	<p>Color del tema que puede tener alguno de os valores de la constante <b>XlThemeColor</b>:</p> <table><tr><th>Valor</th><th>Nombre</th></tr><tr><td>1</td><td><b>xlThemeColorDark1</b></td></tr><tr><td>2</td><td><b>xlThemeColorLight1</b></td></tr><tr><td>3</td><td><b>xlThemeColorDark2</b></td></tr><tr><td>4</td><td><b>xlThemeColorLight2</b></td></tr><tr><td>5</td><td><b>xlThemeColorAccent1</b></td></tr><tr><td>6</td><td><b>xlThemeColorAccent2</b></td></tr><tr><td>7</td><td><b>xlThemeColorAccent3</b></td></tr><tr><td>8</td><td><b>xlThemeColorAccent4</b></td></tr><tr><td>9</td><td><b>xlThemeColorAccent5</b></td></tr><tr><td>10</td><td><b>xlThemeColorAccent6</b></td></tr><tr><td>11</td><td><b>xlThemeColorHyperlink</b></td></tr><tr><td>12</td><td><b>xlThemeColorFollowedHyperlink</b></td></tr></table>	Valor	Nombre	1	<b>xlThemeColorDark1</b>	2	<b>xlThemeColorLight1</b>	3	<b>xlThemeColorDark2</b>	4	<b>xlThemeColorLight2</b>	5	<b>xlThemeColorAccent1</b>	6	<b>xlThemeColorAccent2</b>	7	<b>xlThemeColorAccent3</b>	8	<b>xlThemeColorAccent4</b>	9	<b>xlThemeColorAccent5</b>	10	<b>xlThemeColorAccent6</b>	11	<b>xlThemeColorHyperlink</b>	12	<b>xlThemeColorFollowedHyperlink</b>
Valor	Nombre																										
1	<b>xlThemeColorDark1</b>																										
2	<b>xlThemeColorLight1</b>																										
3	<b>xlThemeColorDark2</b>																										
4	<b>xlThemeColorLight2</b>																										
5	<b>xlThemeColorAccent1</b>																										
6	<b>xlThemeColorAccent2</b>																										
7	<b>xlThemeColorAccent3</b>																										
8	<b>xlThemeColorAccent4</b>																										
9	<b>xlThemeColorAccent5</b>																										
10	<b>xlThemeColorAccent6</b>																										
11	<b>xlThemeColorHyperlink</b>																										
12	<b>xlThemeColorFollowedHyperlink</b>																										

2.4.2.12.4 Ejemplo

Código																									
<pre>Sub prueba_BorderAround()     ActiveSheet.Range("A1:B3").BorderAround _         LineStyle:=xlDouble, _         Weight:=xlThick, _         Color:=255, _         ThemeColor:=xlThemeColorDark2 End Sub</pre>																									
Resultado																									
<table><tr><th></th><th>A</th><th>B</th></tr><tr><th>1</th><td>10</td><td>40</td></tr><tr><th>2</th><td>20</td><td>50</td></tr><tr><th>3</th><td>30</td><td>60</td></tr></table>		A	B	1	10	40	2	20	50	3	30	60	<table><tr><th></th><th>A</th><th>B</th></tr><tr><th>1</th><td>10</td><td>40</td></tr><tr><th>2</th><td>20</td><td>50</td></tr><tr><th>3</th><td>30</td><td>60</td></tr></table>		A	B	1	10	40	2	20	50	3	30	60
	A	B																							
1	10	40																							
2	20	50																							
3	30	60																							
	A	B																							
1	10	40																							
2	20	50																							
3	30	60																							
Inicio	Fin																								

2.4.2.13 CALCULATE

2.4.2.13.1 Descripción

Permite calcular un rango, una hoja, un libro o todos los libros abiertos.

2.4.2.13.2 Sintaxis:

Calculate

2.4.2.13.3 Argumentos

No tiene.

2.4.2.13.4 Ejemplo

Código
<pre>Sub prueba_Calculate()     ActiveSheet.Calculate End Sub</pre>

Resultado																										
<table><tr><td></td><td>A</td></tr><tr><td>1</td><td>10</td></tr><tr><td>2</td><td>20</td></tr><tr><td>3</td><td>0</td></tr></table>		A	1	10	2	20	3	0	<table><tr><td></td><td>A</td></tr><tr><td>1</td><td>10</td></tr><tr><td>2</td><td>20</td></tr><tr><td>3</td><td>=A1+A2</td></tr></table>		A	1	10	2	20	3	=A1+A2	<table><tr><td></td><td>A</td></tr><tr><td>1</td><td>10</td></tr><tr><td>2</td><td>20</td></tr><tr><td>3</td><td>30</td></tr></table>		A	1	10	2	20	3	30
	A																									
1	10																									
2	20																									
3	0																									
	A																									
1	10																									
2	20																									
3	=A1+A2																									
	A																									
1	10																									
2	20																									
3	30																									
Inicio		Fin																								

En este ejemplo, se muestra como se realiza el recálculo tras la ejecución de este procedimiento.

#### 2.4.2.14 CALCULATE ROWMAJOR ORDER

##### 2.4.2.14.1 Descripción

Permite calcular un rango de celdas siguiendo un determinado orden que consiste en empezar por la celda que se halla en la esquina superior izquierda del rango en cuestión y continuar hasta la celda que se halla en la esquina inferior derecha del rango (última fila).

##### 2.4.2.14.2 Sintaxis:

CalculateRowMajorOrder

##### 2.4.2.14.3 Argumentos

No tiene.

##### 2.4.2.14.4 Ejemplo

Código
<pre>Sub prueba_CalculateRowMajorOrder()     Dim r As Range     Set r = Range("B1:B4")     r.CalculateRowMajorOrder End Sub</pre>

Resultado							
	A	B	C		A	B	C
1	10	40	70	1	10	40	70
2	20	50	80	2	20	50	80
3	30	60	90	3	30	60	90
4	0	0	0	4	0	150	0
Inicio				Fin			

### 2.4.2.15 CHECKSPELLING

#### 2.4.2.15.1 Descripción

Permite comprobar la ortografía del objeto indicado.

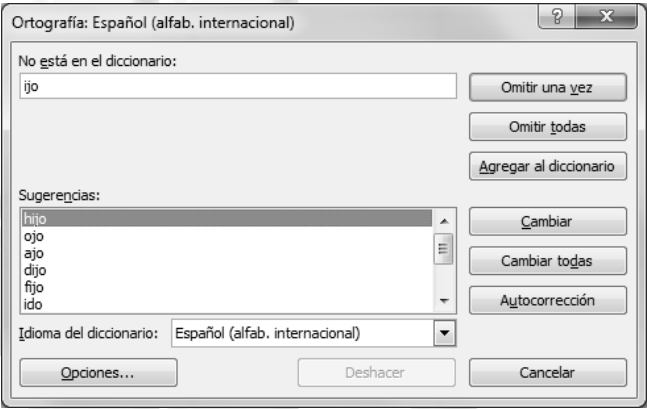
#### 2.4.2.15.2 Sintaxis:

CheckSpelling(CustomDictionary, IgnoreUppercase,  
AlwaysSuggest, SpellLang)

#### 2.4.2.15.3 Argumentos

Argumento	Comentario
<b>CustomDictionary</b>	Diccionario a utilizar si la palabra no se halla en el diccionario principal.
<b>IgnoreUppercase</b>	Si <b>TRUE</b> , no se verifican las palabras que contengan todas las letras en mayúsculas. Si <b>FALSE</b> , sólo se verifican las que tengan todas las letras en mayúsculas. Si no se indica nada, se usa la configuración actual.
<b>AlwaysSuggest</b>	Si <b>TRUE</b> , se muestra una lista de sugerencias.
<b>SpellLang</b>	Idioma del diccionario utilizado. Puede ser alguno de los valores determinados por la constante <b>MsoLanguageID</b> .

2.4.2.15.4 Ejemplo

Código							
<pre>Sub prueba_CheckSpelling()   Dim r As Range   Set r = Range("A1:A2")   r.CheckSpelling AlwaysSuggest:=True End Sub</pre>							
Resultado							
<table><tr><td></td><td>A</td></tr><tr><td>1</td><td>ijo</td></tr><tr><td>2</td><td>urro</td></tr></table>		A	1	ijo	2	urro	
	A						
1	ijo						
2	urro						
Inicio	Fin						

Al ejecutar este procedimiento, se dispara el cuadro de diálogo de Ortografía permitiendo corregir el posible error ortográfico.

2.4.2.16 CLEAR

2.4.2.16.1 Descripción

Borra todo el objeto.

Permite borrar el contenido del objeto indicado.

2.4.2.16.2 Sintaxis:

Clear

2.4.2.16.3 Argumentos

No tiene.

#### 2.4.2.16.4 Ejemplo

Código																	
<pre>Sub prueba_Clear()   Dim r As Range   Set r = Range("A1:A3")   r.Clear End Sub</pre>																	
Resultado																	
<table border="1"> <thead> <tr> <th></th><th>A</th></tr> </thead> <tbody> <tr> <td>1</td><td>10,00</td></tr> <tr> <td>2</td><td>2.000</td></tr> <tr> <td>3</td><td>01/01/2012</td></tr> </tbody> </table>		A	1	10,00	2	2.000	3	01/01/2012	<table border="1"> <thead> <tr> <th></th><th>A</th></tr> </thead> <tbody> <tr> <td>1</td><td></td></tr> <tr> <td>2</td><td></td></tr> <tr> <td>3</td><td></td></tr> </tbody> </table>		A	1		2		3	
	A																
1	10,00																
2	2.000																
3	01/01/2012																
	A																
1																	
2																	
3																	
<b>Inicio</b>	<b>Fin</b>																

### 2.4.2.17 CLEARCOMMENTS

#### 2.4.2.17.1 Descripción

Permite eliminar los comentarios del rango indicado.

#### 2.4.2.17.2 Sintaxis:

ClearComments

#### 2.4.2.17.3 Argumentos

No tiene.

#### 2.4.2.17.4 Ejemplo

Código
<pre>Sub prueba_ClearComments()   Dim r As Range   Set r = Range("A1:A3")   r.ClearComments End Sub</pre>



Resultado					
	A	B	C		A
1	10,00			1	10,00
2	2.000			2	2.000
3	01/01/2012			3	01/01/2012
4					
5					
6					
7					
Inicio				Fin	

## 2.4.2.18 CLEARCONTENTS

### 2.4.2.18.1 Descripción

Permite borrar las fórmulas existentes en un rango dejando intacto el formato del mismo.

### 2.4.2.18.2 Sintaxis:

ClearContents

### 2.4.2.18.3 Argumentos

No tiene.

### 2.4.2.18.4 Ejemplo

Código
<pre> Sub prueba_ClearContents()     Dim r As Range     Set r = Range("A1:A7")     r.ClearContents End Sub </pre>

Resultado											
		A				A				A	
1		10		1		10		1			
2		15		2		=A1+5		2			
3		20		3		=A2+5		3			
4		25		4		=A3+5		4			
5		30		5		=A4+5		5			
6		35		6		=A5+5		6			
7		40		7		=A6+5		7			
Inicio								Fin			

## 2.4.2.19 CLEARFORMATS

### 2.4.2.19.1 Descripción

Permite borrar el formato del objeto indicado.

### 2.4.2.19.2 Sintaxis:

ClearFormats

### 2.4.2.19.3 Argumentos

No tiene.

### 2.4.2.19.4 Ejemplo

Código
<pre>Sub prueba_ClearFormats()   Dim r As Range   Set r = Range("A1:A3")   r.ClearFormats End Sub</pre>

Resultado			
	A		A
1	10,00	1	10
2	2.000	2	2000
3	01/01/2012	3	40909
Inicio		Fin	

## 2.4.2.20 CLEARHYPERLINKS

### 2.4.2.20.1 Descripción

Elimina los hipervínculos existentes en el rango indicado.

### 2.4.2.20.2 Sintaxis:

ClearHyperlinks

### 2.4.2.20.3 Argumentos

No tiene.

### 2.4.2.20.4 Ejemplo

Código			
<pre>Sub prueba_ClearHyperlinks()     Dim r As Range     Set r = Range("A1:A3")     r.ClearHyperlinks End Sub</pre>			
Resultado			
	A		A
1	<a href="#">Mango</a>	1	<a href="#">Mango</a>
2	<a href="#">UOC</a>	2	<a href="#">UOC</a>
3	<a href="http://www.youtube.com/user/juanto2007">http://www.youtube.com/user/juanto2007</a>	3	<a href="http://www.youtube.com/user/juanto2007">http://www.youtube.com/user/juanto2007</a>
4			
5			

Tras ejecutar el procedimiento, los hipervínculos se eliminan y solo queda el texto asociado a los mismos.

### 2.4.2.21 CLEARNOTES

#### 2.4.2.21.1 Descripción

Elimina las notas escritas y las notas sonoras de todas las celdas del rango especificado.


#### 2.4.2.21.2 Sintaxis:

ClearNotes

#### 2.4.2.21.3 Argumentos

No tiene.

#### 2.4.2.21.4 Ejemplo

Código			
<pre>Sub prueba_ClearNotes()     Dim r As Range     Set r = Range("A1:A3")     r.ClearNotes End Sub</pre>			
Resultado			
			

### 2.4.2.22 CLEAROUTLINE

#### 2.4.2.22.1 Descripción

Elimina el esquema asociado al rango que se indique.

**2.4.2.22.2 Sintaxis:**

ClearOutline

**2.4.2.22.3 Argumentos**

No tiene.

**2.4.2.22.4 Ejemplo**

Código																																																							
<pre>Sub prueba_ClearOutline()     Dim r As Range     Set r = Range("A1:B12")     r.ClearOutline End Sub</pre>																																																							
Resultado																																																							
<table><tr><td>12</td><td></td><td>A</td><td>B</td></tr><tr><td></td><td>1</td><td>Provincia</td><td>Importe</td></tr><tr><td rowspan="5">·</td><td>2</td><td>Barcelona</td><td>80</td></tr><tr><td>3</td><td>Barcelona</td><td>150</td></tr><tr><td>4</td><td>Madrid</td><td>220</td></tr><tr><td>5</td><td>Madrid</td><td>230</td></tr><tr><td>6</td><td>España</td><td>680</td></tr><tr><td rowspan="2">-</td><td>7</td><td>Londres</td><td>320</td></tr><tr><td>8</td><td>Manchester</td><td>560</td></tr><tr><td>-</td><td>9</td><td>Inglaterra</td><td>880</td></tr><tr><td rowspan="2">·</td><td>10</td><td>Lyon</td><td>158</td></tr><tr><td>11</td><td>Paris</td><td>740</td></tr><tr><td>-</td><td>12</td><td>Francia</td><td>898</td></tr></table>				12		A	B		1	Provincia	Importe	·	2	Barcelona	80	3	Barcelona	150	4	Madrid	220	5	Madrid	230	6	España	680	-	7	Londres	320	8	Manchester	560	-	9	Inglaterra	880	·	10	Lyon	158	11	Paris	740	-	12	Francia	898						
12		A	B																																																				
	1	Provincia	Importe																																																				
·	2	Barcelona	80																																																				
	3	Barcelona	150																																																				
	4	Madrid	220																																																				
	5	Madrid	230																																																				
	6	España	680																																																				
-	7	Londres	320																																																				
	8	Manchester	560																																																				
-	9	Inglaterra	880																																																				
·	10	Lyon	158																																																				
	11	Paris	740																																																				
-	12	Francia	898																																																				
<table><tr><td></td><td></td><td>A</td><td>B</td></tr><tr><td></td><td>1</td><td>Provincia</td><td>Importe</td></tr><tr><td></td><td>2</td><td>Barcelona</td><td>80</td></tr><tr><td></td><td>3</td><td>Barcelona</td><td>150</td></tr><tr><td></td><td>4</td><td>Madrid</td><td>220</td></tr><tr><td></td><td>5</td><td>Madrid</td><td>230</td></tr><tr><td></td><td>6</td><td>España</td><td>680</td></tr><tr><td></td><td>7</td><td>Londres</td><td>320</td></tr><tr><td></td><td>8</td><td>Manchester</td><td>560</td></tr><tr><td></td><td>9</td><td>Inglaterra</td><td>880</td></tr><tr><td></td><td>10</td><td>Lyon</td><td>158</td></tr><tr><td></td><td>11</td><td>Paris</td><td>740</td></tr><tr><td></td><td>12</td><td>Francia</td><td>898</td></tr></table>						A	B		1	Provincia	Importe		2	Barcelona	80		3	Barcelona	150		4	Madrid	220		5	Madrid	230		6	España	680		7	Londres	320		8	Manchester	560		9	Inglaterra	880		10	Lyon	158		11	Paris	740		12	Francia	898
		A	B																																																				
	1	Provincia	Importe																																																				
	2	Barcelona	80																																																				
	3	Barcelona	150																																																				
	4	Madrid	220																																																				
	5	Madrid	230																																																				
	6	España	680																																																				
	7	Londres	320																																																				
	8	Manchester	560																																																				
	9	Inglaterra	880																																																				
	10	Lyon	158																																																				
	11	Paris	740																																																				
	12	Francia	898																																																				

Ra-Ma®

2.4.2.23 COLUMNDIFFERENCES

2.4.2.23.1 Descripción

Proporciona un rango con las celdas de una columna que son diferentes a una celda indicada como referencia.

2.4.2.23.2 Sintaxis:

ColumnDifferences(Comparison)

2.4.2.23.3 Argumentos

Argumento	Comentario
Comparison	Celda a comparar con el rango indicado.

2.4.2.23.4 Ejemplo

Código																	
<pre>Sub prueba_ColumnDifferences()     Dim d As Range     Dim difs As Range     Set difs = ActiveSheet.Columns("A").ColumnDifferences(ActiveSheet.Range("A2"))     For Each d In difs         Debug.Print d.Address &amp; " - " &amp; d.Value     Next End Sub</pre>																	
Resultado																	
<table><tr><td></td><td>A</td></tr><tr><td>1</td><td>10</td></tr><tr><td>2</td><td>20</td></tr><tr><td>3</td><td>30</td></tr><tr><td>4</td><td>20</td></tr><tr><td>5</td><td>40</td></tr><tr><td>6</td><td>50</td></tr><tr><td>7</td><td>20</td></tr></table>		A	1	10	2	20	3	30	4	20	5	40	6	50	7	20	<div>Inmediato</div> <div>\$A\$1 - 10</div> <div>\$A\$3 - 30</div> <div>\$A\$5 - 40</div> <div>\$A\$6 - 50</div>
	A																
1	10																
2	20																
3	30																
4	20																
5	40																
6	50																
7	20																

En este ejemplo, se imprimen en la ventana de Inmediato aquellas celdas cuyo valor es diferente al valor de la celda A2 (o sea 20).

## 2.4.2.24 CONSOLIDATE

### 2.4.2.24.1 Descripción

Permite consolidar en un solo rango, datos existentes en varios rangos de varias hojas.

### 2.4.2.24.2 Sintaxis:

Consolidate(Sources, Function, TopRow, LeftColumn, CreateLinks)

### 2.4.2.24.3 Argumentos

Argumento	Comentario		
<b>Sources</b>	Matriz con los diferentes orígenes a consolidar. Las referencias a los rangos de dichos orígenes deben estar en formato F1C1 (o R1C1 dependiendo del idioma).		
<b>Function</b>	Indica la función a utilizar que puede ser alguno de los valores de la enumeración <b>XlConsolidationFunction</b> :		
	<b>Valor</b>	<b>Nombre</b>	<b>Comentario</b>
	1000	<b>xlUnknown</b>	Sin función de subtotal.
	-4106	<b>xlAverage</b>	Calcula el promedio.
	-4112	<b>xlCount</b>	Cuenta valores.
	-4113	<b>xlCountNums</b>	Cuenta sólo valores numéricos.
	-4136	<b>xlMax</b>	Halla máximo.
	-4139	<b>xlMin</b>	Halla Mínimo.
	-4149	<b>xlProduct</b>	Multiplica valores.
	-4155	<b>xlStDev</b>	Desviación estándar, basada en una muestra.
	-4156	<b>xlStDevP</b>	Desviación estándar, basada en toda la población.
	-4157	<b>xlSum</b>	Calcula la suma.
	-4164	<b>xlVar</b>	Variación, basada en una muestra.
	-4165	<b>xlVarP</b>	Variación, basada en toda la población.

<b>TopRow</b>	Si TRUE, los datos se consolidan a partir de los encabezados de columna de la fila superior de los rangos que se están consolidando. Si FALSE, los datos se consolidan según su posición. Por defecto vale FALSE.
<b>LeftColumn</b>	Si TRUE, los datos se consolidan a partir de los encabezados de fila de la columna izquierda de los rangos que se están consolidando. Si FALSE, los datos se consolidan según su posición. Por defecto vale FALSE.
<b>CreateLinks</b>	Si TRUE, la consolidación utiliza vínculos para obtener los datos. Si FALSE, los datos son copiados. Por defecto vale FALSE.

#### 2.4.2.24.4 Ejemplo

Código																																											
<pre>Sub prueba_Consolidate()   Dim r As Range   Set r = Range("A2")   r.Consolidate Sources:=Array("C1!R2C1:R4C1", "C2!R2C1:R4C1"), _     Function:=xlSum End Sub</pre>																																											
Resultado																																											
<table><tr><th></th><th>A</th></tr><tr><td>1</td><td>VALORES1</td></tr><tr><td>2</td><td>10</td></tr><tr><td>3</td><td>11</td></tr><tr><td>4</td><td>12</td></tr></table>		A	1	VALORES1	2	10	3	11	4	12	<table><tr><th></th><th>A</th></tr><tr><td>1</td><td>VALORES2</td></tr><tr><td>2</td><td>20</td></tr><tr><td>3</td><td>21</td></tr><tr><td>4</td><td>22</td></tr></table>		A	1	VALORES2	2	20	3	21	4	22	<table><tr><th></th><th>A</th></tr><tr><td>1</td><td>TOTAL</td></tr><tr><td>2</td><td></td></tr><tr><td>3</td><td></td></tr><tr><td>4</td><td></td></tr></table>		A	1	TOTAL	2		3		4		<table><tr><th></th><th>A</th></tr><tr><td>1</td><td>TOTAL</td></tr><tr><td>2</td><td>30</td></tr><tr><td>3</td><td>32</td></tr><tr><td>4</td><td>34</td></tr></table>		A	1	TOTAL	2	30	3	32	4	34
	A																																										
1	VALORES1																																										
2	10																																										
3	11																																										
4	12																																										
	A																																										
1	VALORES2																																										
2	20																																										
3	21																																										
4	22																																										
	A																																										
1	TOTAL																																										
2																																											
3																																											
4																																											
	A																																										
1	TOTAL																																										
2	30																																										
3	32																																										
4	34																																										
C1	C2	RES-Inicio	RES-Fin																																								

Tras la ejecución del procedimiento vemos que efectivamente se suman los valores de las hojas C1 y C2 en la hoja RES la cual inicialmente contienen vacías las celdas A2, A3 y A4.



### 2.4.2.25 Copy

#### 2.4.2.25.1 Descripción

Permite copiar un determinado rango sobre otro o en el portapapeles.

#### 2.4.2.25.2 Sintaxis:

Copy(Destination)

#### 2.4.2.25.3 Argumentos

Argumento	Comentario
<b>Destination</b>	Indica el rango sobre el que se copiará el rango copiado. Si no se indica nada, dicho rango se copiará en el portapapeles.

#### 2.4.2.25.4 Ejemplo

Código																			
<pre> Sub prueba_Copy()     ActiveSheet.Range("A1:A3").Copy _         Destination:=ActiveSheet.Range("B1:B3")     Dim origen As Range     Set origen = ActiveSheet.Range("A1:A3")     Dim destino As Range     Set destino = ActiveSheet.Range("C1:C3")     origen.Copy ' Copia en portapapeles     destino.PasteSpecial ' Recupera de portapapeles End Sub </pre>																			
Resultado																			
<table border="1"> <thead> <tr> <th></th><th>A</th><th>B</th><th>C</th></tr> </thead> <tbody> <tr> <td>1</td><td>10</td><td></td><td></td></tr> <tr> <td>2</td><td>20</td><td></td><td></td></tr> <tr> <td>3</td><td>30</td><td></td><td></td></tr> </tbody> </table>					A	B	C	1	10			2	20			3	30		
	A	B	C																
1	10																		
2	20																		
3	30																		
<table border="1"> <thead> <tr> <th></th><th>A</th><th>B</th><th>C</th></tr> </thead> <tbody> <tr> <td>1</td><td>10</td><td>10</td><td>10</td></tr> <tr> <td>2</td><td>20</td><td>20</td><td>20</td></tr> <tr> <td>3</td><td>30</td><td>30</td><td>30</td></tr> </tbody> </table>					A	B	C	1	10	10	10	2	20	20	20	3	30	30	30
	A	B	C																
1	10	10	10																
2	20	20	20																
3	30	30	30																

Ra-Ma®

## 2.4.2.26 COPYFROMRECORDSET

### 2.4.2.26.1 Descripción

Copia el contenido de un objeto Recordset ADO o DAO en una hoja de cálculo, comenzando en la esquina superior izquierda del rango especificado. Si el objeto Recordset contiene campos con objetos OLE, este método no se completará correctamente.

Permite copiar en la hoja, el contenido de un recordset ADO o DAO. La copia se realiza a partir de la esquina superior izquierda del rango indicado.

### 2.4.2.26.2 Sintaxis:

CopyFromRecordset(Data, MaxRows, MaxColumns)

### 1.1.1.3.6 Argumentos

Argumento	Comentario
<b>Data</b>	Recordset a copiar en el rango.
<b>MaxRows</b>	Máximo número de registros a copiar en la hoja. Por defecto se copiarán todos.
<b>MaxColumns</b>	Máximo número de campos a copiar en la hoja. Por defecto se copiarán todos.

### 2.4.2.26.3 Ejemplo

Código
<pre> Sub prueba_CopyFromRecordset()     Dim sPath, consulta, conexString As String     Dim conex As ADODB.Connection     Dim recordSet As ADODB.recordSet     sPath = "C:\Cursos\Curso_Macros\Temp\bdccurso.mdb"     conexString = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=" &amp; _         sPath &amp; ";Persist Security Info=False;"     Set conex = New ADODB.Connection     conex.Open conexString     Set recordSet = New ADODB.recordSet     With recordSet         .CursorLocation = adUseClient         .CursorType = adOpenStatic         .LockType = adLockOptimistic     End With End Sub </pre>

```

End With
consulta = "select * from Clientes"
recordSet.Open consulta, conex
Range("A1").CopyFromRecordset recordSet
recordSet.Close
conex.Close
Set recordSet = Nothing
Set conex = Nothing
End Sub

```

### Resultado

	A	B	C
1		1 Cliente 1	931111111
2		2 Cliente 2	912222222
3		3 Cliente 3	963333333
4		4 Cliente 4	972444444
5		5 Cliente 5	956666666
6		6 Cliente 6	979595959

## 2.4.2.27 COPYPICTURE

### 2.4.2.27.1 Descripción

Permite copiar el rango en el portatapeles en formato imagen.

### 2.4.2.27.2 Sintaxis:

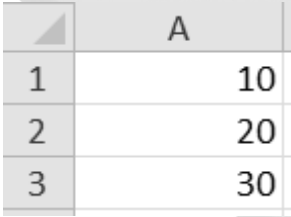
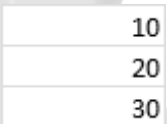
CopyPicture(Appearance, Format)

### 2.4.2.27.3 Argumentos

Argumento	Comentario		
<b>Appearance</b>	Indica cómo se copiará la imagen. Puede tener alguno de los valores de la constante XlPictureAppearance:		
	<b>Valor</b>	<b>Nombre</b>	<b>Comentario</b>
	1	<b>xlScreen</b>	La imagen se copia de forma parecida a como se ve en pantalla.
	2	<b>xlPrinter</b>	La imagen se copia de forma parecida a como se imprime.

<b>Format</b>	Indica el formato que ha de tener la imagen y puede ser alguno de los valores de la constante <b>XlCopyPictureFormat</b> : <table><tr><th>Valor</th><th>Nombre</th><th>Comentario</th></tr><tr><td>2</td><td>xlBitmap</td><td>Mapa de bits (.bmp, .jpg, .gif).</td></tr><tr><td>-4147</td><td>xlPicture</td><td>Imagen dibujada (.png, .wmf, .mix).</td></tr></table>		Valor	Nombre	Comentario	2	xlBitmap	Mapa de bits (.bmp, .jpg, .gif).	-4147	xlPicture	Imagen dibujada (.png, .wmf, .mix).
Valor	Nombre	Comentario									
2	xlBitmap	Mapa de bits (.bmp, .jpg, .gif).									
-4147	xlPicture	Imagen dibujada (.png, .wmf, .mix).									

2.4.2.27.4 Ejemplo

Código		
<pre>Sub prueba_CopyPicture()   Dim r As Range   Set r = Range("A1:A3")   r.CopyPicture Appearance:=xlPrinter, Format:=xlPicture End Sub</pre>		
Resultado		
		<div>10</div> <div>20</div> <div>30</div>
	Appearance:=xlScreen, Format:=xlBitmap	Appearance:=xlPrinter, Format:=xlPicture

2.4.2.28 CREATENAMES

2.4.2.28.1 Descripción

Crea nombres de rangos utilizando los rótulos de columnas o filas del rango indicado.

2.4.2.28.2 Sintaxis:

CreateNames(Top, Left, Bottom, Right)

### 2.4.2.28.3 Argumentos

Argumento	Comentario
<b>Top</b>	Si True, crea los nombres usando los rótulos de la fila superior.
<b>Left</b>	Si True, crea los nombres usando los rótulos de la columna izquierda.
<b>Bottom</b>	Si True, crea los nombres usando los rótulos de la fila inferior.
<b>Right</b>	Si True, crea los nombres usando los rótulos de la columna derecha.

### 2.4.2.28.4 Ejemplo

## Código

```
Sub prueba_CreateNames()  
    Dim r As Range  
    Set r = Range("A1:D4")  
    r.CreateNames Top:=True  
End Sub
```

## Resultado

	A	B	C	D
1	<b>Articulo</b>	<b>Unidades</b>	<b>Precio</b>	<b>Total</b>
2	1	10	100	1000
3	2	20	200	4000
4	3	30	300	9000

Tras ejecutar el procedimiento, podemos comprobar que los nombres de rangos han sido creados utilizando los rótulos de las columnas.

### 2.4.2.29 CUT

#### 2.4.2.29.1 Descripción

Corta el objeto y lo pega en el Portapapeles o en un destino especificado.

#### 2.4.2.29.2 Sintaxis:

Cut(Destination)

#### 2.4.2.29.3 Argumentos

Argumento	Comentario
<b>Destination</b>	Lugar en el que se pegará el rabgo cortado. Por defecto será el portapapeles.

#### 2.4.2.29.4 Ejemplo

Código																															
<pre>Sub prueba_Cut()   Dim r As Range   Set r = Range("A1:A3")   r.Cut Destination:=Range("B2:B4") End Sub</pre>																															
Resultado																															
<table><tr><th></th><th>A</th><th>B</th></tr><tr><td>1</td><td>10</td><td></td></tr><tr><td>2</td><td>20</td><td></td></tr><tr><td>3</td><td>30</td><td></td></tr><tr><td>4</td><td></td><td></td></tr></table>		A	B	1	10		2	20		3	30		4			<table><tr><th></th><th>A</th><th>B</th></tr><tr><td>1</td><td></td><td></td></tr><tr><td>2</td><td></td><td>10</td></tr><tr><td>3</td><td></td><td>20</td></tr><tr><td>4</td><td></td><td>30</td></tr></table>		A	B	1			2		10	3		20	4		30
	A	B																													
1	10																														
2	20																														
3	30																														
4																															
	A	B																													
1																															
2		10																													
3		20																													
4		30																													
Inicio	Fin																														

### 2.4.2.30 DATASERIES

#### 2.4.2.30.1 Descripción

Permite rellenar un rango basándose en una serie de datos.

**2.4.2.30.2 Sintaxis:**

**DataSeries**(Rowcol, Type, Date, Step, Stop, Trend)

**2.4.2.30.3 Argumentos**

Argumento	Comentario		
<b>Rowcol</b>	Constante que indica la dirección a utilizar para rellenar las celdas. Puede ser xlRows o xlColumns. Por defecto se utilizará el tamaño y forma del rango indicado.		
<b>Type</b>	Indica el tipo de la serie y su valor puede ser alguno de los valores de la constante XlDataType:		
	<b>Valor</b>	<b>Nombre</b>	<b>Comentario</b>
	2	xlGrowth	Rellena extendiendo los valores en progresión multiplicativa (p.e. 1,2,4,8).
	3	xlChronological	Rellenar con fechas.
	4	xlAutoFill	Rellenar según Autorrellenar.
	-4132	xlDataSeriesLinear	Rellena extendiendo los valores en forma de suma (p.e. 1,2,3,4).
<b>Date</b>	Indica incremento de tiempo si Type es xlChronological. Puede tener alguno de los valores de la constante XlDateSeriesDate:		
	<b>Valor</b>	<b>Nombre</b>	<b>Comentario</b>
	1	xlDay	Día
	2	xlWeekday	Días de la semana
	3	xlMonth	Mes
	4	xlYear	Año

<b>Step</b>	Incremento a utilizar. Por defecto vale 1.
<b>Stop</b>	Valor en el que se ha de detener la serie. Si no se indica, se rellenará todo el rango.
<b>Trend</b>	Si True, se crea una tendencia lineal o geométrica. Si False, se crea una serie estándar. Por defecto es False.

2.4.2.30.4 Ejemplo

Código																																																																																											
<pre>Sub prueba_DataSeries()   Dim r, s, t As Range   Set r = Range("A1:A10")   Set s = Range("B1:B10")   Set t = Range("C1:C10")   r.DataSeries Type:=xlDataSeriesLinear, step:=5, stop:=40   s.DataSeries Type:=xlChronological, Date:=xlMonth   t.DataSeries Type:=xlGrowth, step:=2 End Sub</pre>																																																																																											
Resultado																																																																																											
<table><tr><th></th><th>A</th><th>B</th><th>C</th></tr><tr><td>1</td><td>5</td><td>01-ene-12</td><td>2</td></tr><tr><td>2</td><td></td><td></td><td></td></tr><tr><td>3</td><td></td><td></td><td></td></tr><tr><td>4</td><td></td><td></td><td></td></tr><tr><td>5</td><td></td><td></td><td></td></tr><tr><td>6</td><td></td><td></td><td></td></tr><tr><td>7</td><td></td><td></td><td></td></tr><tr><td>8</td><td></td><td></td><td></td></tr><tr><td>9</td><td></td><td></td><td></td></tr><tr><td>10</td><td></td><td></td><td></td></tr></table>			A	B	C	1	5	01-ene-12	2	2				3				4				5				6				7				8				9				10				<table><tr><th></th><th>A</th><th>B</th><th>C</th></tr><tr><td>1</td><td>5</td><td>01-ene-12</td><td>2</td></tr><tr><td>2</td><td>10</td><td>01-feb-12</td><td>4</td></tr><tr><td>3</td><td>15</td><td>01-mar-12</td><td>8</td></tr><tr><td>4</td><td>20</td><td>01-abr-12</td><td>16</td></tr><tr><td>5</td><td>25</td><td>01-may-12</td><td>32</td></tr><tr><td>6</td><td>30</td><td>01-jun-12</td><td>64</td></tr><tr><td>7</td><td>35</td><td>01-jul-12</td><td>128</td></tr><tr><td>8</td><td>40</td><td>01-ago-12</td><td>256</td></tr><tr><td>9</td><td></td><td>01-sep-12</td><td>512</td></tr><tr><td>10</td><td></td><td>01-oct-12</td><td>1024</td></tr></table>			A	B	C	1	5	01-ene-12	2	2	10	01-feb-12	4	3	15	01-mar-12	8	4	20	01-abr-12	16	5	25	01-may-12	32	6	30	01-jun-12	64	7	35	01-jul-12	128	8	40	01-ago-12	256	9		01-sep-12	512	10		01-oct-12	1024
	A	B	C																																																																																								
1	5	01-ene-12	2																																																																																								
2																																																																																											
3																																																																																											
4																																																																																											
5																																																																																											
6																																																																																											
7																																																																																											
8																																																																																											
9																																																																																											
10																																																																																											
	A	B	C																																																																																								
1	5	01-ene-12	2																																																																																								
2	10	01-feb-12	4																																																																																								
3	15	01-mar-12	8																																																																																								
4	20	01-abr-12	16																																																																																								
5	25	01-may-12	32																																																																																								
6	30	01-jun-12	64																																																																																								
7	35	01-jul-12	128																																																																																								
8	40	01-ago-12	256																																																																																								
9		01-sep-12	512																																																																																								
10		01-oct-12	1024																																																																																								
Inicio		Fin																																																																																									



## 2.4.2.31 DELETE

### 2.4.2.31.1 Descripción

Permite eliminar el rango indicado desplazando las celdas contiguas en un determinado sentido.

### 2.4.2.31.2 Sintaxis:

Delete(Shift)

### 2.4.2.31.3 Argumentos

Argumento	Comentario		
<b>Shift</b>	Indica el sentido en el que deben desplazarse las celdas contiguas a las que se están eliminando. Puede tener alguno de los valores de la constante <b>XlDeleteShiftDirection</b> :		
	<b>Valor</b>	<b>Nombre</b>	<b>Comentario</b>
	-4159	<b>xlShiftToLeft</b>	Las celdas se desplazarán hacia la izquierda.
	-4162	<b>xlShiftUp</b>	Las celdas se desplazarán hacia arriba.

### 2.4.2.31.4 Ejemplo

Código
<pre>Sub prueba_Delete()     Dim r As Range     Set r = Range("B2:B3")     r.Delete (xlShiftToLeft) End Sub</pre>

Ra-Ma®

Resultado							
	A	B	C		A	B	C
1	10	60	110	1	10	60	110
2	20	70	120	2	20	120	
3	30	80	130	3	30	130	
4	40	90	140	4	40	90	140
5	50	100	150	5	50	100	150
Inicio				Fin			

### 2.4.2.32 DIALOGBOX

#### 2.4.2.32.1 Descripción

Muestra un cuadro de diálogo definido por una tabla de definición de cuadro de diálogo de una hoja de macros de Microsoft Excel 4.0. Devuelve el número del control elegido o devuelve False si el usuario hace clic en el botón Cancelar.

#### 2.4.2.32.2 Sintaxis:

DialogBox

#### 2.4.2.32.3 Argumentos

No tiene.

### 2.4.2.33 DIRTY

#### 2.4.2.33.1 Descripción

Método que indica a Excel que un rango o libro se ha de actualizar.

#### 2.4.2.33.2 Sintaxis:

Dirty

### 2.4.2.33.3 Argumentos

No tiene.

### 2.4.2.33.4 Ejemplo

Código	
<pre>Sub prueba_Dirty()     Dim r As Range     Set r = Range("A3:A3")     ' r.Dirty     Debug.Print _         Application.ThisWorkbook.Saved End Sub</pre>	<pre>Sub prueba_Dirty()     Dim r As Range     Set r = Range("A3:A3")     r.Dirty     Debug.Print _         Application.ThisWorkbook.Saved End Sub</pre>
Resultado	
<div>Inmediato</div> <div>Verdadero</div>	<div>Inmediato</div> <div>Falso</div>

En el primer ejemplo (donde la instrucción `r.Dirty` está comentada), podemos observar que si el libro está salvado previamente a la ejecución del procedimiento, la propiedad `Saved` del libro en curso indica que efectivamente está guardado y no tiene ningún cambio pendiente de actualizar.

En el segundo ejemplo, al descomentar la instrucción `r.Dirty` para que ésta se ejecute, observaremos que, aunque salvemos el libro previamente, la instrucción `Dirty` siempre forzará a que el libro tenga cambios pendientes de actualizar.

### 2.4.2.34 DISCARDCHANGES

#### 2.4.2.34.1 Descripción

Descarta todos los cambios realizados en las celdas editadas de un rango basado en un origen de datos OLAP. Al ejecutar este método, se ejecuta una operación de actualización que recupera los valores mas recientes del origen de datos. En el servidor OLAP, se ejecuta una instrucción **ROLLBACK TRANSACTION**.

#### 2.4.2.34.2 Sintaxis:

`DiscardChanges`

### 2.4.2.34.3 Argumentos

No tiene.

Código
<pre>Sub prueba_DiscardChanges()     Dim r As Range     Set r = Range("A1")     r.Value = 10     r.DiscardChanges End Sub</pre>

### 2.4.2.35 EXPORTASFIXEDFORMAT

#### 2.4.2.35.1 Descripción

Permite exportar el contenido de un rango a un archivo con un formato determinado.


#### 2.4.2.35.2 Sintaxis:

ExportAsFixedFormat(Type, Filename, Quality, IncludeDocProperties, IgnorePrintAreas, From, To, OpenAfterPublish)

#### 2.4.2.35.3 Argumentos

Argumento	Comentario
<b>Type</b>	Tipo a utilizar para la exportación. Puede ser: <b>xlTypePDF</b> o <b>xlTypeXPS</b> .
<b>Filename</b>	Nombre del archivo con el que se creará la exportación.
<b>Quality</b>	Calidad con la que se creará el documento. Puede ser: <b>xlQualityStandard</b> o <b>xlQualityMinimum</b> .
<b>IncludeDocProperties</b>	Si se indica True, se incluyen las propiedades en los documentos.
<b>IgnorePrintAreas</b>	Si se indica True se omiten las áreas de impresión.
<b>From</b>	Indica el número de la primera página a publicar.
<b>To</b>	Indica el número de la última página a publicar.
<b>OpenAfterPublish</b>	Si se indica True, el documento resultante se abre una vez exportado.

### 2.4.2.35.4 Ejemplo

Código													
<pre>Sub prueba_ExportAsFixedFormat()     Dim r As Range     Set r = Range("A1:B3")     ' Se exporta a pdf y abre documento.     r.ExportAsFixedFormat Type:=xlTypePDF, _         Filename:="c:\temp\Juanto.pdf", _         Quality:=xlQualityStandard, _         OpenAfterPublish:=True End Sub</pre>													
Resultado													
<table><tr><th></th><th>A</th><th>B</th></tr><tr><td>1</td><td>Dato 1</td><td>10</td></tr><tr><td>2</td><td>Dato 2</td><td>20</td></tr><tr><td>3</td><td>Dato 3</td><td>30</td></tr></table>		A	B	1	Dato 1	10	2	Dato 2	20	3	Dato 3	30	
	A	B											
1	Dato 1	10											
2	Dato 2	20											
3	Dato 3	30											
Inicio	Fin												

### 2.4.2.36 FILLDOWN

#### 2.4.2.36.1 Descripción

Permite copiar el contenido de una o varias celdas ubicadas en la fila superior de un rango hacia las filas de abajo del rango indicado.

#### 2.4.2.36.2 Sintaxis:

FillDown

### 2.4.2.36.3 Argumentos

No tiene.

### 2.4.2.36.4 Ejemplo

Código			
<pre>Sub prueba_FillDown()   Dim r As Range   Set r = Range("A1:C5")   r.FillDown End Sub</pre>			
Resultado			
	A	B	C
1	10	20	30
2			
3			
4			
5			
Inicio		Fin	

### 2.4.2.37 FILLLEFT

#### 2.4.2.37.1 Descripción

Permite copiar el contenido de una o varias celdas ubicadas en la columna derecha de un rango hacia las columnas de la izquierda del rango indicado.

#### 2.4.2.37.2 Sintaxis:

FillLeft

#### 2.4.2.37.3 Argumentos

No tiene.

### 2.4.2.37.4 Ejemplo

Código																				
<pre>Sub prueba_FillLeft()   Dim r As Range   Set r = Range("A1:C3")   r.FillLeft End Sub</pre>																				
Resultado																				
<table><tr><td></td><td>A</td><td>B</td><td>C</td></tr><tr><td>1</td><td></td><td></td><td>10</td></tr><tr><td>2</td><td></td><td></td><td>20</td></tr><tr><td>3</td><td></td><td></td><td>30</td></tr></table>						A	B	C	1			10	2			20	3			30
	A	B	C																	
1			10																	
2			20																	
3			30																	
<table><tr><td></td><td>A</td><td>B</td><td>C</td></tr><tr><td>1</td><td>10</td><td>10</td><td>10</td></tr><tr><td>2</td><td>20</td><td>20</td><td>20</td></tr><tr><td>3</td><td>30</td><td>30</td><td>30</td></tr></table>						A	B	C	1	10	10	10	2	20	20	20	3	30	30	30
	A	B	C																	
1	10	10	10																	
2	20	20	20																	
3	30	30	30																	
Inicio			Fin																	

### 2.4.2.38 FILLRIGHT

#### 2.4.2.38.1 Descripción

Permite copiar el contenido de una o varias celdas ubicadas en la columna izquierda de un rango hacia las columnas de la derecha del rango indicado.

#### 2.4.2.38.2 Sintaxis:

FillRight

#### 2.4.2.38.3 Argumentos

No tiene.

#### 2.4.2.38.4 Ejemplo

Código
<pre>Sub prueba_FillRight()   Dim r As Range   Set r = Range("A1:C3")   r.FillRight End Sub</pre>

Resultado							
	A	B	C		A	B	C
1	10			1	10	10	10
2	20			2	20	20	20
3	30			3	30	30	30
Inicio				Fin			

### 2.4.2.39 FILLUP

#### 2.4.2.39.1 Descripción

Permite copiar el contenido de una o varias celdas ubicadas en la fila inferior de un rango hacia las filas de arriba del rango indicado.

#### 2.4.2.39.2 Sintaxis:

FillUp

#### 2.4.2.39.3 Argumentos

No tiene.

#### 2.4.2.39.4 Ejemplo

Código
<pre>Sub prueba_FillUp()   Dim r As Range   Set r = Range("A1:C5")   r.FillUp End Sub</pre>

Ra-Ma®



Resultado							
	A	B	C		A	B	C
1				1	10	20	30
2				2	10	20	30
3				3	10	20	30
4				4	10	20	30
5	10	20	30	5	10	20	30
Inicio				Fin			

## 2.4.2.40 FIND

### 2.4.2.40.1 Descripción

Permite buscar una información en un determinado rango.

### 2.4.2.40.2 Sintaxis:

Find(What, After, LookIn, LookAt, SearchOrder, SearchDirection, MatchCase, MatchByte, SearchFormat)

### 2.4.2.40.3 Argumentos

Argumento	Comentario		
What	Texto que se está buscando.		
After	Celda a partir de la cual se iniciará la búsqueda.		
LookIn	Tipo de información a buscar. Puede ser alguno de los valores de la constante <b>XlFindLookIn</b> :		
	xlComments	-4144	Comentarios.
	xlFormulas	-4123	Fórmulas.
	xlValues	-4163	Valores.

<b>LookAt</b>	Parte que se está buscando y puede ser alguno de los valores de la constante <b>XILookAt</b> : <table><tr><td><b>xlWhole</b></td><td>1</td><td>Completa.</td></tr><tr><td><b>xlPart</b></td><td>2</td><td>Parcial.</td></tr></table>	<b>xlWhole</b>	1	Completa.	<b>xlPart</b>	2	Parcial.
<b>xlWhole</b>	1	Completa.					
<b>xlPart</b>	2	Parcial.					
<b>SearchOrder</b>	Indica el orden de la búsqueda y puede ser alguno de los valores de la constante <b>XISearchOrder</b> : <table><tr><td><b>xlByRows</b></td><td>1</td><td>Por filas.</td></tr><tr><td><b>xlByColumns</b></td><td>2</td><td>Por columnas.</td></tr></table>	<b>xlByRows</b>	1	Por filas.	<b>xlByColumns</b>	2	Por columnas.
<b>xlByRows</b>	1	Por filas.					
<b>xlByColumns</b>	2	Por columnas.					
<b>SearchDirection</b>	Indica la dirección de la búsqueda y puede ser alguno de los valores de la constante <b>XISearchDirection</b> : <table><tr><td><b>xlNext</b></td><td>1</td><td>Buscar la siguiente coincidencia.</td></tr><tr><td><b>xlPrevious</b></td><td>2</td><td>Buscar la coincidencia anterior.</td></tr></table>	<b>xlNext</b>	1	Buscar la siguiente coincidencia.	<b>xlPrevious</b>	2	Buscar la coincidencia anterior.
<b>xlNext</b>	1	Buscar la siguiente coincidencia.					
<b>xlPrevious</b>	2	Buscar la coincidencia anterior.					
<b>MatchCase</b>	Si <b>True</b> , se distingue entre mayúsculas y minúsculas. Por defecto vale <b>False</b> .						
<b>MatchByte</b>	Si <b>True</b> , se buscan coincidencias entre caracteres de 2 bytes.						
<b>SearchFormat</b>	Formato de la búsqueda.						

#### 2.4.2.40.4 Ejemplo

Código
<pre> Sub prueba_Find()     Dim r As Range     Set r = Range("A1:G4")     Dim s As Range     Dim a As String     Dim partícula As String     partícula = "an" ' Partícula a buscar     Set s = r.Find(What:=partícula, LookIn:=xlValues)     If Not s Is Nothing Then         Debug.Print "En celda (" &amp; s.Address &amp; ") hallada palabra ("     &amp; s.Value &amp; ")"     End If End Sub </pre>

Resultado							
	A	B	C	D	E	F	G
1	En	un	lugar	de	la	Mancha,	de
2	cuyo	nombre	no	quiero	acordarme,	no	ha
3	mucho	tiempo	que	vivía	un	hidalgo	de
4	los	de	lanza	en	astillero,	adarga	antigua,

Inmediato
En celda (\$F\$1) hallada palabra (Mancha,)

Tras ejecutar el procedimiento, podemos observar como se muestra la primera coincidencia hallada.

## 2.4.2.41 FINDNEXT

### 2.4.2.41.1 Descripción

Permite volver a realizar la última búsqueda ejecutada con el método Find pero para localizar la coincidencia siguiente.

### 2.4.2.41.2 Sintaxis:

FindNext(After)

### 2.4.2.41.3 Argumentos

Argumento	Comentario
After	Celda a partir de la cual se iniciará la búsqueda. La búsqueda se empezará a partir de esta celda sin tenerla en cuenta.

Ra-Ma®

2.4.2.41.4 Ejemplo

Código

```
Sub prueba_FindNext()  
  Dim r As Range  
  Set r = Range("A1:G4")  
  Dim s As Range  
  Dim a As String  
  Dim particula As String  
  particula = "an" ' Partícula a buscar  
  Set s = r.Find(What:=particula, LookIn:=xlValues)  
  If Not s Is Nothing Then  
    a = s.Address  
    Do  
      Debug.Print "En celda (" & s.Address & ") hallada  
palabra (" & s.Value & ")"  
      Set s = r.FindNext(s)  
    Loop While Not s Is Nothing And s.Address <> a  
  End If  
End Sub
```

Resultado

	A	B	C	D	E	F	G
1	En	un	lugar	de	la	Mancha,	de
2	cuyo	nombre	no	quiero	acordarme,	no	ha
3	mucho	tiempo	que	vivía	un	hidalgo	de
4	los	de	lanza	en	astillero,	adarga	antigua,

Inmediato

```
En celda ($F$1) hallada palabra (Mancha,)  
En celda ($C$4) hallada palabra (lanza)  
En celda ($G$4) hallada palabra (antigua,)
```

Tras ejecutar el procedimiento, podemos observar como se muestran las celdas y palabras que contienen la partícula buscada “an”.

## 2.4.2.42 FINDPREVIOUS

### 2.4.2.42.1 Descripción

Permite volver a realizar la última búsqueda ejecutada con el método Find pero para localizar la coincidencia anterior.

### 2.4.2.42.2 Sintaxis:

FindPrevious(After)

### 2.4.2.42.3 Argumentos

Argumento	Comentario
After	Celda anterior a la cual se iniciará la búsqueda. La búsqueda se empezará a partir de esta celda sin tenerla en cuenta.

### 2.4.2.42.4 Ejemplo

Código
<pre> Sub prueba_FindPrevious()     Dim r As Range     Set r = Range("A1:G4")     Dim s As Range     Dim a As String     Dim particula As String     particula = "an" ' Partícula a buscar     Set s = r.Find(What:=particula, LookIn:=xlValues)     If Not s Is Nothing Then         a = s.Address         Do             Debug.Print "En celda (" &amp; s.Address &amp; ") hallada palabra (" &amp; s.Value &amp; ")"             Set s = r.FindPrevious(s)         Loop While Not s Is Nothing And s.Address &lt;&gt; a     End If End Sub </pre>

Ra-Ma®

Resultado							
	A	B	C	D	E	F	G
1	En	un	lugar	de	la	Mancha,	de
2	cuyo	nombre	no	quiero	acordarme,	no	ha
3	mucho	tiempo	que	vivía	un	hidalgo	de
4	los	de	lanza	en	astillero,	adarga	antigua,

Inmediato							
En celda (\$F\$1) hallada palabra (Mancha,) En celda (\$G\$4) hallada palabra (antigua,) En celda (\$C\$4) hallada palabra (lanza)							

### 2.4.2.43 FUNCTIONWIZARD

#### 2.4.2.43.1 Descripción

Invoca el asistente de introducción de funciones para insertar una función sobre la celda indicada.

#### 2.4.2.43.2 Sintaxis:

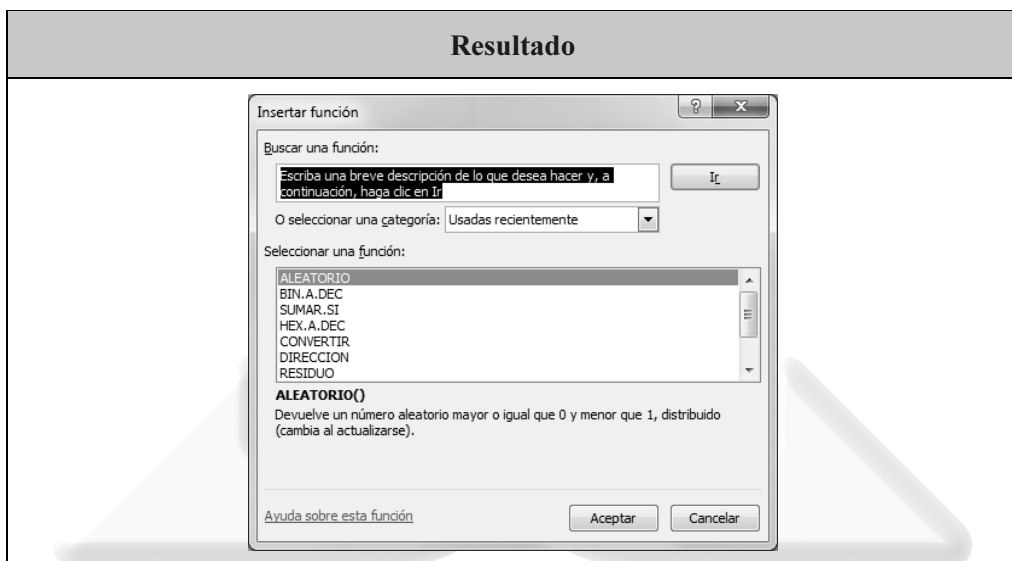
FunctionWizard

#### 2.4.2.43.3 Argumentos

No tiene.

#### 2.4.2.43.4 Ejemplo

Código
<pre>Sub prueba_FunctionWizard()   Dim r As Range   Set r = Range("B2")   r.FunctionWizard End Sub</pre>

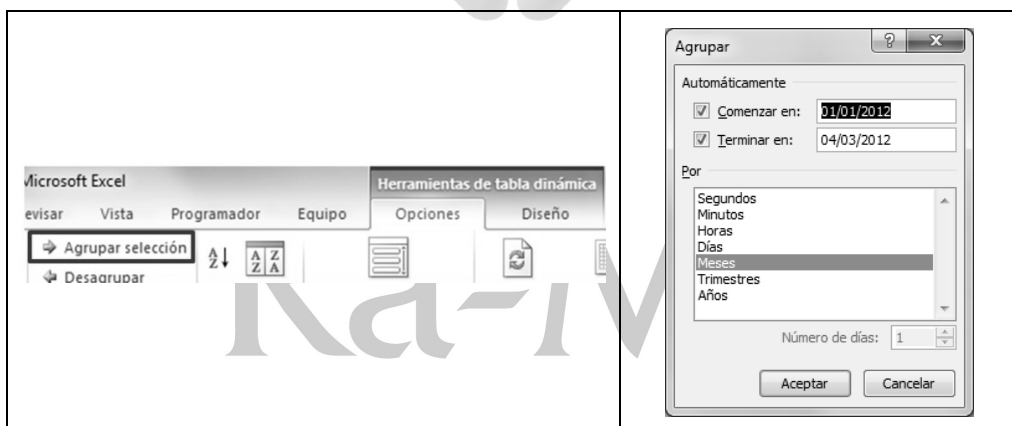


## 2.4.2.44 GROUP

### 2.4.2.44.1 Descripción

Cuando el objeto Range representa una sola celda en un rango de datos de un campo de tabla dinámica, el método Group realizará una agrupación numérica o basada en datos en dicho campo.

Equivale a realizar una agrupación sobre una tabla dinámica invocando la funcionalidad **Agrupar Selección** del grupo de controles **Agrupar** de la pestaña de **Herramientas de tabla dinámica**:



### 2.4.2.44.2 Sintaxis:

Group(Start, End, By, Periods)

### 2.4.2.44.3 Argumentos

Argumento	Comentario
<b>Start</b>	Si True, se utiliza el primer valor del campo.
<b>End</b>	Si True, se utiliza el último valor del campo. Si el campo es una fecha, el argumento especificará días si el cuarto elemento es True y el resto False. Si no se especifica nada, Excel elige un tamaño de grupo predeterminado.
<b>By</b>	Indica el tamaño de cada grupo.
<b>Periods</b>	Matriz que indica el período del grupo a partir de valores de tipo booleanos. Este argumento solo sirve para campos de tipo fecha.

### 2.4.2.44.4 Ejemplo

Código																																							
<pre>Sub prueba_Group()     Dim r As Range     Set r = Range("E2")     r.Group Start:=True, _         End:=True, _         periods:=Array(False, False, False, False, True, False, False) End Sub</pre>																																							
Resultado																																							
<table border="1"> <thead> <tr> <th>E</th><th>F</th></tr> </thead> <tbody> <tr> <td><b>Etiquetas de fila</b></td><td><b>Suma de Importe</b></td></tr> <tr><td>01/01/2012</td><td>100</td></tr> <tr><td>02/01/2012</td><td>200</td></tr> <tr><td>03/01/2012</td><td>300</td></tr> <tr><td>01/02/2012</td><td>400</td></tr> <tr><td>02/02/2012</td><td>500</td></tr> <tr><td>03/02/2012</td><td>600</td></tr> <tr><td>04/02/2012</td><td>700</td></tr> <tr><td>01/03/2012</td><td>800</td></tr> <tr><td>02/03/2012</td><td>900</td></tr> <tr><td>03/03/2012</td><td>1000</td></tr> <tr> <td><b>Total general</b></td><td><b>5500</b></td></tr> </tbody> </table>	E	F	<b>Etiquetas de fila</b>	<b>Suma de Importe</b>	01/01/2012	100	02/01/2012	200	03/01/2012	300	01/02/2012	400	02/02/2012	500	03/02/2012	600	04/02/2012	700	01/03/2012	800	02/03/2012	900	03/03/2012	1000	<b>Total general</b>	<b>5500</b>	<table border="1"> <thead> <tr> <th>E</th><th>F</th></tr> </thead> <tbody> <tr> <td><b>Etiquetas de fila</b></td><td><b>Suma de Importe</b></td></tr> <tr><td>ene</td><td>600</td></tr> <tr><td>feb</td><td>2200</td></tr> <tr><td>mar</td><td>2700</td></tr> <tr> <td><b>Total general</b></td><td><b>5500</b></td></tr> </tbody> </table>	E	F	<b>Etiquetas de fila</b>	<b>Suma de Importe</b>	ene	600	feb	2200	mar	2700	<b>Total general</b>	<b>5500</b>
E	F																																						
<b>Etiquetas de fila</b>	<b>Suma de Importe</b>																																						
01/01/2012	100																																						
02/01/2012	200																																						
03/01/2012	300																																						
01/02/2012	400																																						
02/02/2012	500																																						
03/02/2012	600																																						
04/02/2012	700																																						
01/03/2012	800																																						
02/03/2012	900																																						
03/03/2012	1000																																						
<b>Total general</b>	<b>5500</b>																																						
E	F																																						
<b>Etiquetas de fila</b>	<b>Suma de Importe</b>																																						
ene	600																																						
feb	2200																																						
mar	2700																																						
<b>Total general</b>	<b>5500</b>																																						
<b>Inicio</b>	<b>Fin</b>																																						



## 2.4.2.45 INSERT

### 2.4.2.45.1 Descripción

Permite insertar un rango en una hoja desplazando las celdas contiguas al punto en el que se inserta.

### 2.4.2.45.2 Sintaxis:

Insert(Shift, CopyOrigin)

### 2.4.2.45.3 Argumentos

Argumento	Comentario		
<b>Shift</b>	Indica hacia donde se ha de producir el desplazamiento y puede tener alguno de los valores de la constante <code>XlInsertShiftDirection</code> :		
	<b>xlShiftDown</b>	-4121	Desplazar las celdas hacia abajo.
	<b>xlShiftToRight</b>	-4161	Desplazar las celdas hacia la derecha.
<b>CopyOrigin</b>	Indica el origen de la copia.		

### 2.4.2.45.4 Ejemplo

Código
<pre> Sub prueba_Insert()     Dim r As Range     Set r = Range("B2")     r.Insert shift:=xlShiftToRight, _     CopyOrigin:=Range("A6:A7").Copy End Sub </pre>

Ra-Ma®

Resultado																																																																												
<table><tr><th></th><th>A</th><th>B</th><th>C</th></tr><tr><td>1</td><td>10</td><td>50</td><td>90</td></tr><tr><td>2</td><td>20</td><td>60</td><td>100</td></tr><tr><td>3</td><td>30</td><td>70</td><td>110</td></tr><tr><td>4</td><td>40</td><td>80</td><td>120</td></tr><tr><td>5</td><td></td><td></td><td></td></tr><tr><td>6</td><td>A</td><td></td><td></td></tr><tr><td>7</td><td>B</td><td></td><td></td></tr></table>					A	B	C	1	10	50	90	2	20	60	100	3	30	70	110	4	40	80	120	5				6	A			7	B			<table><tr><th></th><th>A</th><th>B</th><th>C</th><th>D</th></tr><tr><td>1</td><td>10</td><td>50</td><td>90</td><td></td></tr><tr><td>2</td><td>20</td><td>A</td><td>60</td><td>100</td></tr><tr><td>3</td><td>30</td><td>B</td><td>70</td><td>110</td></tr><tr><td>4</td><td>40</td><td>80</td><td>120</td><td></td></tr><tr><td>5</td><td></td><td></td><td></td><td></td></tr><tr><td>6</td><td>A</td><td></td><td></td><td></td></tr><tr><td>7</td><td>B</td><td></td><td></td><td></td></tr></table>		A	B	C	D	1	10	50	90		2	20	A	60	100	3	30	B	70	110	4	40	80	120		5					6	A				7	B			
	A	B	C																																																																									
1	10	50	90																																																																									
2	20	60	100																																																																									
3	30	70	110																																																																									
4	40	80	120																																																																									
5																																																																												
6	A																																																																											
7	B																																																																											
	A	B	C	D																																																																								
1	10	50	90																																																																									
2	20	A	60	100																																																																								
3	30	B	70	110																																																																								
4	40	80	120																																																																									
5																																																																												
6	A																																																																											
7	B																																																																											
Inicio				Fin																																																																								

2.4.2.46 INSERTINDENT

2.4.2.46.1 Descripción

Agrega una sangría al rango especificado.

2.4.2.46.2 Sintaxis:

InsertIndent(InsertAmount)

2.4.2.46.3 Argumentos

Argumento	Comentario
InsertAmount	Cantidad a añadir a la sangría actual.

2.4.2.46.4 Ejemplo

Código
<pre>Sub prueba_InsertIndent()     Range("A2:A3").InsertIndent(1)     Range("A4:A5").InsertIndent(2) End Sub</pre>

Resultado																													
<table border="1"> <thead> <tr> <th></th><th>A</th></tr> </thead> <tbody> <tr><td>1</td><td>10</td></tr> <tr><td>2</td><td>20</td></tr> <tr><td>3</td><td>30</td></tr> <tr><td>4</td><td>40</td></tr> <tr><td>5</td><td>50</td></tr> <tr><td>6</td><td>60</td></tr> </tbody> </table>		A	1	10	2	20	3	30	4	40	5	50	6	60	<table border="1"> <thead> <tr> <th></th><th>A</th></tr> </thead> <tbody> <tr><td>1</td><td>10</td></tr> <tr><td>2</td><td>20</td></tr> <tr><td>3</td><td>30</td></tr> <tr><td>4</td><td>40</td></tr> <tr><td>5</td><td>50</td></tr> <tr><td>6</td><td>60</td></tr> </tbody> </table>		A	1	10	2	20	3	30	4	40	5	50	6	60
	A																												
1	10																												
2	20																												
3	30																												
4	40																												
5	50																												
6	60																												
	A																												
1	10																												
2	20																												
3	30																												
4	40																												
5	50																												
6	60																												
<b>Inicio</b>	<b>Fin</b>																												

## 2.4.2.47 JUSTIFY

### 2.4.2.47.1 Descripción

Este método permite reorganizar un texto dentro de un rango de forma que, si el texto sobrepasa la longitud del mismo, el texto se trunca y el resto pasa a la celda que tiene justo debajo.

### 2.4.2.47.2 Sintaxis:

Justify

### 2.4.2.47.3 Argumentos

No tiene.

### 2.4.2.47.4 Ejemplo

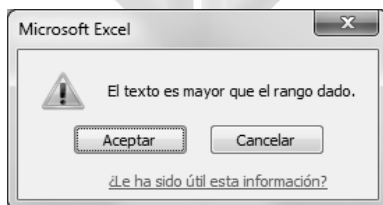
Código 1	Código 2
<pre>Sub prueba_Justify()     Range("A1:A1").Justify End Sub</pre>	<pre>Sub prueba_Justify()     Range("A1:B1").Justify End Sub</pre>

Original					
A1		'Es una prueba de texto muy largo para forzar la justificación.			
	A	B	C	D	E
1	Es una prueba de texto muy largo para forzar la justificación.				

Resultado 1		Resultado 2	
	A		B
1	Es una	1	Es una prueba de texto
2	prueba de	2	muy largo para forzar la
3	texto muy	3	justificación.
4	largo para		
5	forzar la		
6	justificación.		

Inicio	Fin
--------	-----

En ambos casos aparece el siguiente cuadro de diálogo indicando que la longitud del texto supera el rango indicado y que se cortará.



## 2.4.2.48 LISTNAMES

### 2.4.2.48.1 Descripción

Proporciona una lista de nombres existentes en la hoja y que no estén ocultos.

**2.4.2.48.2 Sintaxis:**

ListNames

**2.4.2.48.3 Argumentos**

No tiene.

**2.4.2.48.4 Ejemplo**

Código		
<pre>Sub prueba_ListNames()     Range("A1").ListNames End Sub</pre>		
Resultado		
	A	B
1	Articulo	=CreateNames!\$A\$2:\$A\$4
2	CANTIDAD	=ListNames!#jREF!
3	CELDA1	=ApplyNames!\$A\$1
4	CELDA2	=ApplyNames!\$A\$2
5	CELDA3	=ApplyNames!\$A\$3
6	COLUM2	=ApplyNames!\$A:\$A
7	HojaJuanto	=Sheet1!\$A\$1
8	JUAN1	=ApplyNames2!\$A\$1
9	JUAN2	=ApplyNames2!\$C\$3
10	JUAN3	=ApplyNames2!\$A\$3
11	MIRANGO	=Name!\$A\$1:\$C\$1

**2.4.2.49 MERGE****2.4.2.49.1 Descripción**

Crea una celda combinada a partir del objeto Range especificado.

**2.4.2.49.2 Sintaxis:**

Merge(Across)

2.4.2.49.3 Argumentos

Argumento	Comentario
Across	Si True, las celdas de cada fila del rango especificado se combinan como celdas combinadas distintas. Por defecto vale False.

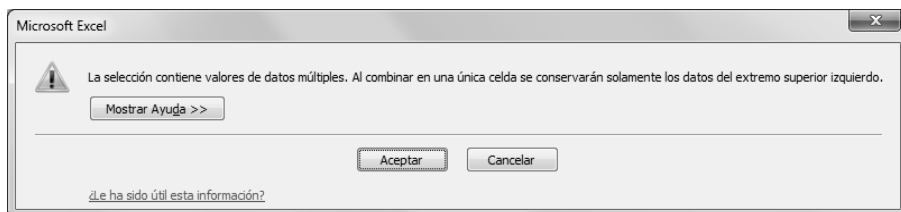
2.4.2.49.4 Ejemplo

Original				
	A	B	C	D
1	uno			
2	dos			
3	tres			
4	cuatro			

Código 1		Código 2		
<pre>Sub prueba_Merge()   Range("A1:B1").Merge   Range("A2:C2").Merge   Range("A3:D4").Merge _     Across:=True End Sub</pre>		<pre>Sub prueba_Merge()   Range("A1:B1").Merge   Range("A2:C2").Merge   Range("A3:D4").Merge _     Across:=False End Sub</pre>		
Resultado 1		Resultado 2		
	A	B	C	D
1	uno			
2	dos			
3	tres			
4	cuatro			

	A	B	C	D
1	uno			
2	dos			
3				
4	tres			

En el segundo ejemplo, aparece el siguiente cuadro de diálogo advirtiendo que la combinación conservará solo los datos del extremo superior izquierdo del rango indicado:



## 2.4.2.50 NAVIGATEARROW

### 2.4.2.50.1 Descripción

Mueve una flecha de seguimiento del rango especificado a la celda o celdas precedentes, dependientes o causantes de error. Selecciona las celdas precedentes, dependientes o causantes de error y devuelve un objeto Range que representa la nueva selección. Este método provoca un error si se aplica a una celda sin flechas de seguimiento visibles.

Permite seleccionar la celda o celdas precedentes o dependientes a una celda que forma parte de una fórmula o función.

### 2.4.2.50.2 Sintaxis:

NavigateArrow(TowardPrecedent, ArrowNumber, LinkNumber)

### 2.4.2.50.3 Argumentos

Argumento	Comentario
<b>TowardPrecedent</b>	Si True, el movimiento de la flecha es hacia las precedentes. Si false será hacia las dependientes.
<b>ArrowNumber</b>	Indica cuantas flechas se ha de mover el cursor.
<b>LinkNumber</b>	En caso de tratarse de una referencia externa, el argumento indica cual será la referencia externa a seguir.

Ra-Ma®

### 2.4.2.50.4 Ejemplo

Supongamos la siguiente hoja que contiene una tabla simple de cantidades, precios y totales:

	A	B	C
1	Cantidad	Precio	Valor
2	2	10	20
3	4	20	80
4	6	30	180
5	12		280

En la misma, puede observarse que la celda C2, depende de A2 y B2 y a su vez, precede a C5. A continuación, veremos 2 ejemplos en los que podemos seleccionar la celda precedente y dependiente respectivamente.

	Código	Resultado																								
True, 1	<pre>Sub prueba_NavigateArrow()   Dim r As Range   Set r = Range("C2")   r.NavigateArrow True, 1 End Sub</pre>	<table><tr><th></th><th>A</th><th>B</th><th>C</th></tr><tr><th>1</th><th>Cantidad</th><th>Precio</th><th>Valor</th></tr><tr><td>2</td><td>2</td><td>10</td><td>20</td></tr><tr><td>3</td><td>4</td><td>20</td><td>80</td></tr><tr><td>4</td><td>6</td><td>30</td><td>180</td></tr><tr><td>5</td><td>12</td><td></td><td>280</td></tr></table>		A	B	C	1	Cantidad	Precio	Valor	2	2	10	20	3	4	20	80	4	6	30	180	5	12		280
	A	B	C																							
1	Cantidad	Precio	Valor																							
2	2	10	20																							
3	4	20	80																							
4	6	30	180																							
5	12		280																							
True, 2	<pre>Sub prueba_NavigateArrow()   Dim r As Range   Set r = Range("C2")   r.NavigateArrow True, 2 End Sub</pre>	<table><tr><th></th><th>A</th><th>B</th><th>C</th></tr><tr><th>1</th><th>Cantidad</th><th>Precio</th><th>Valor</th></tr><tr><td>2</td><td>2</td><td>10</td><td>20</td></tr><tr><td>3</td><td>4</td><td>20</td><td>80</td></tr><tr><td>4</td><td>6</td><td>30</td><td>180</td></tr><tr><td>5</td><td>12</td><td></td><td>280</td></tr></table>		A	B	C	1	Cantidad	Precio	Valor	2	2	10	20	3	4	20	80	4	6	30	180	5	12		280
	A	B	C																							
1	Cantidad	Precio	Valor																							
2	2	10	20																							
3	4	20	80																							
4	6	30	180																							
5	12		280																							
False, 1	<pre>Sub prueba_NavigateArrow()   Dim r As Range   Set r = Range("C2")   r.NavigateArrow False, 1 End Sub</pre>	<table><tr><th></th><th>A</th><th>B</th><th>C</th></tr><tr><th>1</th><th>Cantidad</th><th>Precio</th><th>Valor</th></tr><tr><td>2</td><td>2</td><td>10</td><td>20</td></tr><tr><td>3</td><td>4</td><td>20</td><td>80</td></tr><tr><td>4</td><td>6</td><td>30</td><td>180</td></tr><tr><td>5</td><td>12</td><td></td><td>280</td></tr></table>		A	B	C	1	Cantidad	Precio	Valor	2	2	10	20	3	4	20	80	4	6	30	180	5	12		280
	A	B	C																							
1	Cantidad	Precio	Valor																							
2	2	10	20																							
3	4	20	80																							
4	6	30	180																							
5	12		280																							



2.4.2.51 NOTETEXT

2.4.2.51.1 Descripción

Devuelve o establece la nota asociada con la celda que se encuentra en la esquina superior izquierda del rango. String de lectura y escritura. Las notas de celda han sido reemplazadas por comentarios de rango. Para obtener más información, vea el objeto Comment.

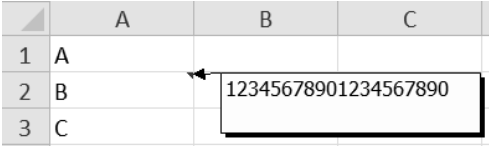
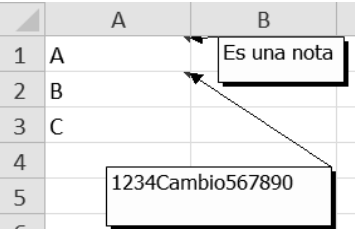
2.4.2.51.2 Sintaxis:

NoteText(Text, Start, Length)

2.4.2.51.3 Argumentos

Argumento	Comentario
Text	Texto a incluir en la nota.
Start	Posición a partir de la cual se reemplazará el texto indicado.
Length	Longitud del texto que se cambiará.

2.4.2.51.4 Ejemplo

Código	
<pre>Sub prueba_NoteText()     Range("A1").NoteText ("Es una nota")     Range("A2").NoteText Text:="Cambio", Start:=5, Length:=10 End Sub</pre>	
Resultado	
	

En este ejemplo, puede apreciarse cómo se ha insertado una nota en la celda A1 y cómo se ha modificado la nota existente en la celda A2 sustituyendo los 10 caracteres que habían a partir de la posición 5 por el texto “Cambio”.

## 2.4.2.52 PARSE

### 2.4.2.52.1 Descripción

Este método permite trocear una cadena de texto basándose en un determinado patrón y dejar el resultado en un rango determinado. Dicho patrón es el que indicamos en el parámetro ParseLine y básicamente se trata de una cadena de texto compuesto por corchetes que indican el número de caracteres que componen cada parte.

### 2.4.2.52.2 Sintaxis:

Parse(ParseLine, Destination)

### 2.4.2.52.3 Argumentos

Argumento	Comentario
<b>ParseLine</b>	Cadena de texto compuesta por corchetes indicando la parte que compone cada uno de los trozos en los que se dividirá la cadena indicada.
<b>Destination</b>	Rango que indica la esquina superior izquierda en la que se colocará el resultado de la división realizada por Parse.

### 2.4.2.52.4 Ejemplo

Código
<pre>Sub prueba_Parse()   Dim r As Range   Set r = Range("A2:A4")   r.Parse parseLine:="[xxxx] [xx] [xx]", Destination:=Range("B2") End Sub</pre>

Resultado									
	A	B	C	D		A	B	C	D
1	F.Reves	Año	Mes	Día	1	F.Reves	Año	Mes	Día
2	20120701				2	20120701	2012	7	1
3	20000127				3	20000127	2000	1	27
4	20100530				4	20100530	2010	5	30

### 2.4.2.53 PASTESPECIAL

#### 2.4.2.53.1 Descripción

Pega un objeto Range del Portapapeles en el rango especificado.

#### 2.4.2.53.2 Sintaxis:

PasteSpecial(Paste, Operation, SkipBlanks, Transpose)

#### 2.4.2.53.3 Argumentos

Argumento	Comentario		
<b>Paste</b>	Indica qué es lo que se va apegar del rango. Puede ser alguno de los valores de la constante XlPasteType:		
	<b>Valor</b>	<b>Nombre</b>	<b>Comentario</b>
	-4163	xlPasteValues	Pega valores.
	-4144	xlPasteComments	Pega comentarios.
	-4123	xlPasteFormulas	Pega fórmulas.
	-4122	xlPasteFormats	Pega formato de origen.
	-4104	xlPasteAll	Pega todo.
	6	xlPasteValidation	Pega validaciones.

	7	<b>xlPasteAllExceptBorders</b>	Pega todo excepto los bordes.
	8	<b>xlPasteColumnWidths</b>	Pega ancho de la columna.
	11	<b>xlPasteFormulasAndNumberFormats</b>	Pega fórmulas y formatos de número.
	12	<b>xlPasteValuesAndNumberFormats</b>	Pega valores y los formatos de número.
	13	<b>xlPasteAllUsingSourceTheme</b>	Pega todo usando el tema de origen.
	14	<b>xlPasteAllMergingConditionalFormats</b>	Pega todo y combina formatos condicionales.
<b>Operation</b>	Operación a realizar durante el copiado. Puede ser alguno de los valores de la constante XlPasteSpecialOperation:		
	<b>Valor</b>	<b>Nombre</b>	<b>Comentario</b>
	-4142	<b>xlPasteSpecialOperationNone</b>	Pega sin realizar ninguna operación.
	2	<b>xlPasteSpecialOperationAdd</b>	Pega sumando los datos a los valores existentes en las celdas destino.
	3	<b>xlPasteSpecialOperationSubtract</b>	Pega restando los datos a los valores existentes en las celdas destino.

	4	<b>xlPasteSpecialOperationMultiply</b>	Pega multiplicando los datos por los valores existentes en las celdas destino.
	5	<b>xlPasteSpecialOperationDivide</b>	Pega dividiendo los datos por los valores existentes en las celdas destino.
<b>SkipBlanks</b>	Si True, las celdas en blanco en el origen no se pegan en el destino. Por defecto vale False.		
<b>Transpose</b>	Si True cambia filas por columnas. Por defecto vale False.		

#### 2.4.2.53.4 Ejemplo

Código																																		
<pre> Sub prueba_PasteSpecial()     Dim r As Range     Set r = Range("A1:A3")     r.Copy     Range("B1").PasteSpecial Paste:=xlPasteAll     Range("C1").PasteSpecial Paste:=xlPasteValues     Range("D1").PasteSpecial Paste:=xlPasteFormulas     Range("A5").PasteSpecial Transpose:=True End Sub </pre>																																		
Resultado																																		
<table border="1"> <thead> <tr> <th></th><th>A</th><th>B</th><th>C</th><th>D</th></tr> </thead> <tbody> <tr> <td>1</td><td>1.000,00</td><td></td><td></td><td></td></tr> <tr> <td>2</td><td>2.000,00</td><td></td><td></td><td></td></tr> <tr> <td>3</td><td>3.000,00</td><td></td><td></td><td></td></tr> </tbody> </table>						A	B	C	D	1	1.000,00				2	2.000,00				3	3.000,00													
	A	B	C	D																														
1	1.000,00																																	
2	2.000,00																																	
3	3.000,00																																	
<table border="1"> <thead> <tr> <th></th><th>A</th><th>B</th><th>C</th><th>D</th></tr> </thead> <tbody> <tr> <td>1</td><td>1.000,00</td><td>1.000,00</td><td>1000</td><td>1000</td></tr> <tr> <td>2</td><td>2.000,00</td><td>2.000,00</td><td>2000</td><td>2000</td></tr> <tr> <td>3</td><td>3.000,00</td><td>3.000,00</td><td>3000</td><td>3000</td></tr> <tr> <td>4</td><td></td><td></td><td></td><td></td></tr> <tr> <td>5</td><td>1.000,00</td><td>2.000,00</td><td>3.000,00</td><td></td></tr> </tbody> </table>						A	B	C	D	1	1.000,00	1.000,00	1000	1000	2	2.000,00	2.000,00	2000	2000	3	3.000,00	3.000,00	3000	3000	4					5	1.000,00	2.000,00	3.000,00	
	A	B	C	D																														
1	1.000,00	1.000,00	1000	1000																														
2	2.000,00	2.000,00	2000	2000																														
3	3.000,00	3.000,00	3000	3000																														
4																																		
5	1.000,00	2.000,00	3.000,00																															
<b>Inicio</b>			<b>Fin</b> ®																															

## 2.4.2.54 PRINTOUT

### 2.4.2.54.1 Descripción

Permite imprimir el rango indicado configurando ciertas características.

### 2.4.2.54.2 Sintaxis:

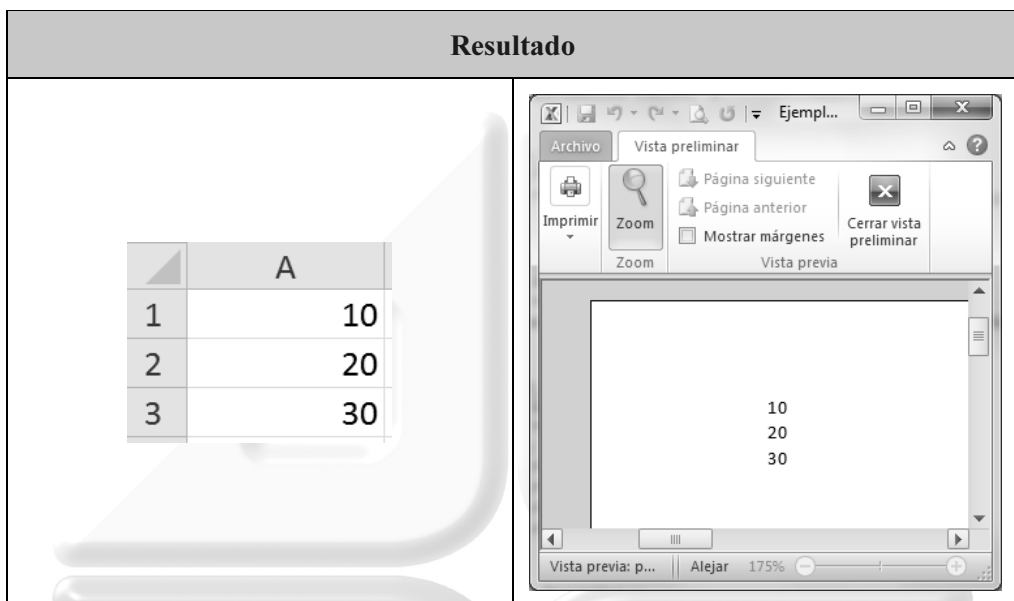
PrintOut(From, To, Copies, Preview, ActivePrinter, PrintToFile, Collate, PrToFileName)

### 2.4.2.54.3 Argumentos

Argumento	Comentario
<b>From</b>	Primera página a imprimir.
<b>To</b>	Ultima página a imprimir.
<b>Copies</b>	Número de copias.
<b>Preview</b>	Si se indica True se realiza una vista preliminar.
<b>ActivePrinter</b>	Nombre de la impresora a utilizar.
<b>PrintToFile</b>	Si se indica True se la impresión se envía a un archivo.
<b>Collate</b>	Si se indica True se pueden intercalar varias copias.
<b>PrToFileName</b>	Indica el nombre del fichero hacia el que se desea enviar la impresión.

### 2.4.2.54.4 Ejemplo

Código
<pre>Sub prueba_PrintOut()     Dim r As Range     Set r = Range("A1:A3")     r.PrintOut Preview:=True End Sub</pre>



## 2.4.2.55 PRINTPREVIEW

### 2.4.2.55.1 Descripción

Proporciona una vista previa de tal y como se imprimiría el rango.

### 2.4.2.55.2 Sintaxis:

PrintPreview(EnableChanges)

### 2.4.2.55.3 Argumentos

Argumento	Comentario
<b>EnableChanges</b>	Si se indica True, el usuario puede realizar cambios durante la vista preliminar en ciertos parámetros como son los márgenes, la orientación de la página, el encabezado y pie, etc... Se trata de un parámetro opcional.

2.4.2.55.4 Ejemplo

Código									
<pre>Sub prueba_PrintPreview()   Dim r As Range   Set r = Range("A1:A3")   r.PrintPreview EnableChanges:=True End Sub</pre>									
Resultado									
<table><tr><td></td><td>A</td></tr><tr><td>1</td><td>10</td></tr><tr><td>2</td><td>20</td></tr><tr><td>3</td><td>30</td></tr></table>		A	1	10	2	20	3	30	
	A								
1	10								
2	20								
3	30								

2.4.2.56 REMOVEDUPLICATES

2.4.2.56.1 Descripción

Permite eliminar los valores duplicados existentes en un rango.

2.4.2.56.2 Sintaxis:

RemoveDuplicates(Columns, Header)

2.4.2.56.3 Argumentos

Argumento	Comentario
Columns	Matriz que indica qué columnas tiene la información duplicada.
Header	Indica si la primera fila tiene información sobre encabezados. Puede ser alguno de los valores de la constante XlYesNoGuess:



	0	xlGuess	MS Excel intenta determinar qué encabezado se ha de utilizar.
	1	xlYes	El encabezado está en la primera fila.
	2	xlNo	No hay cabeceras.

#### 2.4.2.56.4 Ejemplo

Código																																																	
<pre>Sub prueba_RemoveDuplicates()   Dim r As Range   Set r = Range("A1:B8")   r.RemoveDuplicates Columns:=1, Header:=xlYes End Sub</pre>																																																	
Resultado																																																	
<table><tr><th></th><th>A</th><th>B</th></tr><tr><td>1</td><td>Valor</td><td>Nota</td></tr><tr><td>2</td><td>10</td><td>A</td></tr><tr><td>3</td><td>20</td><td>B</td></tr><tr><td>4</td><td>10</td><td>C</td></tr><tr><td>5</td><td>30</td><td>D</td></tr><tr><td>6</td><td>10</td><td>E</td></tr><tr><td>7</td><td>50</td><td>F</td></tr><tr><td>8</td><td>60</td><td>G</td></tr></table>		A	B	1	Valor	Nota	2	10	A	3	20	B	4	10	C	5	30	D	6	10	E	7	50	F	8	60	G	<table><tr><th></th><th>A</th><th>B</th></tr><tr><td>1</td><td>Valor</td><td>Nota</td></tr><tr><td>2</td><td>10</td><td>A</td></tr><tr><td>3</td><td>20</td><td>B</td></tr><tr><td>4</td><td>30</td><td>D</td></tr><tr><td>5</td><td>50</td><td>F</td></tr><tr><td>6</td><td>60</td><td>G</td></tr></table>		A	B	1	Valor	Nota	2	10	A	3	20	B	4	30	D	5	50	F	6	60	G
	A	B																																															
1	Valor	Nota																																															
2	10	A																																															
3	20	B																																															
4	10	C																																															
5	30	D																																															
6	10	E																																															
7	50	F																																															
8	60	G																																															
	A	B																																															
1	Valor	Nota																																															
2	10	A																																															
3	20	B																																															
4	30	D																																															
5	50	F																																															
6	60	G																																															
Inicio	Fin																																																

#### 2.4.2.57 REMOVESUBTOTAL

##### 2.4.2.57.1 Descripción

Permite eliminar los subtotales existentes en una lista. Equivale a hacer clic sobre el botón Quitar todos del cuadro de diálogo Subtotales invocado desde el

grupo de controles Esquema ubicado en la pestaña Datos. Supongamos la siguiente información y el subtotal aplicado mediante el cuadro de diálogo comentado:

<table><tr><th></th><th>A</th><th>B</th><th>C</th></tr><tr><td>1</td><td>Provincia</td><td>Vendedor</td><td>Importe</td></tr><tr><td>2</td><td>Barcelona</td><td>Juan</td><td>80</td></tr><tr><td>3</td><td>Barcelona</td><td>Maria</td><td>150</td></tr><tr><td>4</td><td>Madrid</td><td>Juan</td><td>220</td></tr><tr><td>5</td><td>Madrid</td><td>Maria</td><td>230</td></tr><tr><td>6</td><td>Londres</td><td>Juan</td><td>320</td></tr><tr><td>7</td><td>Paris</td><td>Juan</td><td>740</td></tr></table>					A	B	C	1	Provincia	Vendedor	Importe	2	Barcelona	Juan	80	3	Barcelona	Maria	150	4	Madrid	Juan	220	5	Madrid	Maria	230	6	Londres	Juan	320	7	Paris	Juan	740																				
	A	B	C																																																				
1	Provincia	Vendedor	Importe																																																				
2	Barcelona	Juan	80																																																				
3	Barcelona	Maria	150																																																				
4	Madrid	Juan	220																																																				
5	Madrid	Maria	230																																																				
6	Londres	Juan	320																																																				
7	Paris	Juan	740																																																				
<div>Subtotales</div> <div>Para cada cambio en: Provincia Usar función: Suma Agregar subtotal a: Provincia Vendedor Importe <input checked="" type="checkbox"/> Reemplazar subtotales actuales <input type="checkbox"/> Salto de página entre grupos <input checked="" type="checkbox"/> Resumen debajo de los datos Quitar todos    Aceptar    Cancelar</div>																																																							
<table><tr><th></th><th>A</th><th>B</th><th>C</th></tr><tr><td>1</td><td>Provincia</td><td>Vendedor</td><td>Importe</td></tr><tr><td>2</td><td>Barcelona</td><td>Juan</td><td>80</td></tr><tr><td>3</td><td>Barcelona</td><td>Maria</td><td>150</td></tr><tr><td>4</td><td>Total Barcelona</td><td></td><td>230</td></tr><tr><td>5</td><td>Madrid</td><td>Juan</td><td>220</td></tr><tr><td>6</td><td>Madrid</td><td>Maria</td><td>230</td></tr><tr><td>7</td><td>Total Madrid</td><td></td><td>450</td></tr><tr><td>8</td><td>Londres</td><td>Juan</td><td>320</td></tr><tr><td>9</td><td>Total Londres</td><td></td><td>320</td></tr><tr><td>10</td><td>Paris</td><td>Juan</td><td>740</td></tr><tr><td>11</td><td>Total Paris</td><td></td><td>740</td></tr><tr><td>12</td><td>Total general</td><td></td><td>1740</td></tr></table>					A	B	C	1	Provincia	Vendedor	Importe	2	Barcelona	Juan	80	3	Barcelona	Maria	150	4	Total Barcelona		230	5	Madrid	Juan	220	6	Madrid	Maria	230	7	Total Madrid		450	8	Londres	Juan	320	9	Total Londres		320	10	Paris	Juan	740	11	Total Paris		740	12	Total general		1740
	A	B	C																																																				
1	Provincia	Vendedor	Importe																																																				
2	Barcelona	Juan	80																																																				
3	Barcelona	Maria	150																																																				
4	Total Barcelona		230																																																				
5	Madrid	Juan	220																																																				
6	Madrid	Maria	230																																																				
7	Total Madrid		450																																																				
8	Londres	Juan	320																																																				
9	Total Londres		320																																																				
10	Paris	Juan	740																																																				
11	Total Paris		740																																																				
12	Total general		1740																																																				

2.4.2.57.2 Sintaxis:

RemoveSubtotal

2.4.2.57.3 Argumentos

No tiene.

2.4.2.57.4 Ejemplo

Código																																																																																					
<pre>Sub prueba_RemoveSubtotal()   Dim r As Range   Set r = Range("A1:C12")   r.RemoveSubtotal End Sub</pre>																																																																																					
Resultado																																																																																					
<table><tr><th></th><th>A</th><th>B</th><th>C</th></tr><tr><td>1</td><td>Provincia</td><td>Vendedor</td><td>Importe</td></tr><tr><td>2</td><td>Barcelona</td><td>Juan</td><td>80</td></tr><tr><td>3</td><td>Barcelona</td><td>Maria</td><td>150</td></tr><tr><td>4</td><td>Total Barcelona</td><td></td><td>230</td></tr><tr><td>5</td><td>Madrid</td><td>Juan</td><td>220</td></tr><tr><td>6</td><td>Madrid</td><td>Maria</td><td>230</td></tr><tr><td>7</td><td>Total Madrid</td><td></td><td>450</td></tr><tr><td>8</td><td>Londres</td><td>Juan</td><td>320</td></tr><tr><td>9</td><td>Total Londres</td><td></td><td>320</td></tr><tr><td>10</td><td>Paris</td><td>Juan</td><td>740</td></tr><tr><td>11</td><td>Total Paris</td><td></td><td>740</td></tr><tr><td>12</td><td>Total general</td><td></td><td>1740</td></tr></table>		A	B	C	1	Provincia	Vendedor	Importe	2	Barcelona	Juan	80	3	Barcelona	Maria	150	4	Total Barcelona		230	5	Madrid	Juan	220	6	Madrid	Maria	230	7	Total Madrid		450	8	Londres	Juan	320	9	Total Londres		320	10	Paris	Juan	740	11	Total Paris		740	12	Total general		1740	<table><tr><th></th><th>A</th><th>B</th><th>C</th></tr><tr><td>1</td><td>Provincia</td><td>Vendedor</td><td>Importe</td></tr><tr><td>2</td><td>Barcelona</td><td>Juan</td><td>80</td></tr><tr><td>3</td><td>Barcelona</td><td>Maria</td><td>150</td></tr><tr><td>4</td><td>Madrid</td><td>Juan</td><td>220</td></tr><tr><td>5</td><td>Madrid</td><td>Maria</td><td>230</td></tr><tr><td>6</td><td>Londres</td><td>Juan</td><td>320</td></tr><tr><td>7</td><td>Paris</td><td>Juan</td><td>740</td></tr></table>		A	B	C	1	Provincia	Vendedor	Importe	2	Barcelona	Juan	80	3	Barcelona	Maria	150	4	Madrid	Juan	220	5	Madrid	Maria	230	6	Londres	Juan	320	7	Paris	Juan	740
	A	B	C																																																																																		
1	Provincia	Vendedor	Importe																																																																																		
2	Barcelona	Juan	80																																																																																		
3	Barcelona	Maria	150																																																																																		
4	Total Barcelona		230																																																																																		
5	Madrid	Juan	220																																																																																		
6	Madrid	Maria	230																																																																																		
7	Total Madrid		450																																																																																		
8	Londres	Juan	320																																																																																		
9	Total Londres		320																																																																																		
10	Paris	Juan	740																																																																																		
11	Total Paris		740																																																																																		
12	Total general		1740																																																																																		
	A	B	C																																																																																		
1	Provincia	Vendedor	Importe																																																																																		
2	Barcelona	Juan	80																																																																																		
3	Barcelona	Maria	150																																																																																		
4	Madrid	Juan	220																																																																																		
5	Madrid	Maria	230																																																																																		
6	Londres	Juan	320																																																																																		
7	Paris	Juan	740																																																																																		
Inicio	Fin																																																																																				

## 2.4.2.58 REPLACE

### 2.4.2.58.1 Descripción

Devuelve un valor Boolean que indica los caracteres de las celdas del rango especificado. El uso de este método no cambia la selección ni la celda activa.

### 2.4.2.58.2 Sintaxis:

Replace(What, Replacement, LookAt, SearchOrder, MatchCase, MatchByte, SearchFormat, ReplaceFormat)

### 2.4.2.58.3 Argumentos

Argumento	Comentario								
What	Texto que se está buscando.								
Replacement	Texto a utilizar para el remplazamiento.								
LookAt	Parte que se está buscando y puede ser alguno de los valores de la constante <b>XlLookAt</b> : <table><tr><td>xlWhole</td><td>1</td><td>Completa.</td></tr><tr><td>xlPart</td><td>2</td><td>Parcial.</td></tr></table>			xlWhole	1	Completa.	xlPart	2	Parcial.
xlWhole	1	Completa.							
xlPart	2	Parcial.							
SearchOrder	Indica el orden de la búsqueda y puede ser alguno de los valores de la constante <b>XlSearchOrder</b> : <table><tr><td>xlByRows</td><td>1</td><td>Por filas.</td></tr><tr><td>xlByColumns</td><td>2</td><td>Por columnas.</td></tr></table>			xlByRows	1	Por filas.	xlByColumns	2	Por columnas.
xlByRows	1	Por filas.							
xlByColumns	2	Por columnas.							
MatchCase	Si <b>True</b> , se distingue entre mayúsculas y minúsculas. Por defecto vale <b>False</b> .								
MatchByte	Si <b>True</b> , se buscan coincidencias entre caracteres de 2 bytes.								
SearchFormat	Formato de la búsqueda.								
ReplaceFormat	Formato del reemplazo.								

2.4.2.58.4 Ejemplo

Código																					
<pre>Sub prueba_Replace()   Dim r As Range   Set r = Range("A1:A4")   Dim p_ini, p_fin As String   p_ini = "EFG" ' Partícula a sustituir   p_fin = "xyz" ' Particulo que reemplaza original   With Application.ReplaceFormat.Font     .FontStyle = "Negrita"   End With   r.Replace What:=p_ini, Replacement:=p_fin, LookAt:=xlPart, _     SearchOrder:=xlByRows, MatchCase:=False, SearchFormat:=False, _     ReplaceFormat:=True End Sub</pre>																					
Resultado																					
<table><tr><th></th><th>A</th></tr><tr><td>1</td><td>ABC123EFG456</td></tr><tr><td>2</td><td>123EFG456HIJ</td></tr><tr><td>3</td><td>EFG456HIJ789</td></tr><tr><td>4</td><td>456HIJ789ABC</td></tr></table>		A	1	ABC123EFG456	2	123EFG456HIJ	3	EFG456HIJ789	4	456HIJ789ABC	<table><tr><th></th><th>A</th></tr><tr><td>1</td><td><b>ABC123xyz456</b></td></tr><tr><td>2</td><td><b>123xyz456HIJ</b></td></tr><tr><td>3</td><td><b>xyz456HIJ789</b></td></tr><tr><td>4</td><td>456HIJ789ABC</td></tr></table>		A	1	<b>ABC123xyz456</b>	2	<b>123xyz456HIJ</b>	3	<b>xyz456HIJ789</b>	4	456HIJ789ABC
	A																				
1	ABC123EFG456																				
2	123EFG456HIJ																				
3	EFG456HIJ789																				
4	456HIJ789ABC																				
	A																				
1	<b>ABC123xyz456</b>																				
2	<b>123xyz456HIJ</b>																				
3	<b>xyz456HIJ789</b>																				
4	456HIJ789ABC																				
Inicio	Fin																				

2.4.2.59 ROWDIFFERENCES

2.4.2.59.1 Descripción

Proporciona un objeto con las celdas diferentes a una determinada celda indicada en la comparación.

2.4.2.59.2 Sintaxis:

RowDifferences(Comparison)  
Ra-Ma®

2.4.2.59.3 Argumentos

Argumento	Comentario
Comparison	Celda a utilizar commo modelo en la comparación.

1.1.1.3.7 Ejemplo

Código																											
<pre>Sub prueba_RowDifferences()   Dim r_analizar, r_res, c_modelo, c_dif As Range   Set r_analizar = Range("A1:G1")   Set c_modelo = Range("A1")   Set r_res = r_analizar.RowDifferences(Comparison:=c_modelo)   For Each c_dif In r_res     Debug.Print "Modelo " &amp; c_modelo.Address &amp; _       " valor (" &amp; c_modelo.Value &amp; _       ") dif (" &amp; c_dif.Address &amp; _       ") valor (" &amp; c_dif.Value &amp; ")"   Next End Sub</pre>																											
Resultado																											
<table><tr><td></td><td>A</td><td>B</td><td>C</td><td>D</td><td>E</td><td>F</td><td>G</td></tr><tr><td>1</td><td>10</td><td>20</td><td>10</td><td>30</td><td>10</td><td>50</td><td>60</td></tr></table>		A	B	C	D	E	F	G	1	10	20	10	30	10	50	60	<table><tr><th colspan="2">Inmediato</th></tr><tr><td>Modelo</td><td>\$A\$1 valor (10) dif (\$B\$1) valor (20)</td></tr><tr><td>Modelo</td><td>\$A\$1 valor (10) dif (\$D\$1) valor (30)</td></tr><tr><td>Modelo</td><td>\$A\$1 valor (10) dif (\$F\$1) valor (50)</td></tr><tr><td>Modelo</td><td>\$A\$1 valor (10) dif (\$G\$1) valor (60)</td></tr></table>	Inmediato		Modelo	\$A\$1 valor (10) dif (\$B\$1) valor (20)	Modelo	\$A\$1 valor (10) dif (\$D\$1) valor (30)	Modelo	\$A\$1 valor (10) dif (\$F\$1) valor (50)	Modelo	\$A\$1 valor (10) dif (\$G\$1) valor (60)
	A	B	C	D	E	F	G																				
1	10	20	10	30	10	50	60																				
Inmediato																											
Modelo	\$A\$1 valor (10) dif (\$B\$1) valor (20)																										
Modelo	\$A\$1 valor (10) dif (\$D\$1) valor (30)																										
Modelo	\$A\$1 valor (10) dif (\$F\$1) valor (50)																										
Modelo	\$A\$1 valor (10) dif (\$G\$1) valor (60)																										
Inicio	Fin																										

2.4.2.60 RUN

2.4.2.60.1 Descripción

Permite ejecutar una macro o función.

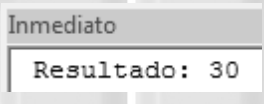
2.4.2.60.2 Sintaxis:

Run(Arg1, Arg2, Arg3, Arg4, Arg5, Arg6, Arg7, Arg8, Arg9, Arg10, Arg11, Arg12, Arg13, Arg14, Arg15, Arg16, Arg17, Arg18, Arg19, Arg20, Arg21, Arg22, Arg23, Arg24, Arg25, Arg26, Arg27, Arg28, Arg29, Arg30))

### 2.4.2.60.3 Argumentos

Argumento	Comentario
<b>Arg1</b> ... <b>Arg30</b>	Cada uno de los argumentos que se pasan a la función.

### 2.4.2.60.4 Ejemplo

Código
<pre>Sub prueba_Run()     Dim r As Variant     r = Run("suma", 10, 20)     Debug.Print "Resultado: " &amp; r End Sub  Function suma(ByVal op1 As Integer, ByVal op2 As Integer) As Integer     suma = op1 + op2 End Function</pre>
Resultado


## 2.4.2.61 SELECT

### 2.4.2.61.1 Descripción

Permite seleccionar un rango determinado.

### 2.4.2.61.2 Sintaxis:

Select

### 2.4.2.61.3 Argumentos

No tiene.

Ra-Ma®

### 2.4.2.61.4 Ejemplo

Código																					
<pre>Sub prueba_Select()   Dim r As Range   Set r = Range("A2:A3")   r.Select End Sub</pre>																					
Resultado																					
<table border="1"> <thead> <tr> <th></th><th>A</th></tr> </thead> <tbody> <tr> <td>1</td><td>10</td></tr> <tr> <td>2</td><td>20</td></tr> <tr> <td>3</td><td>30</td></tr> <tr> <td>4</td><td>40</td></tr> </tbody> </table>		A	1	10	2	20	3	30	4	40	<table border="1"> <thead> <tr> <th></th><th>A</th></tr> </thead> <tbody> <tr> <td>1</td><td>10</td></tr> <tr> <td>2</td><td>20</td></tr> <tr> <td>3</td><td>30</td></tr> <tr> <td>4</td><td>40</td></tr> </tbody> </table>		A	1	10	2	20	3	30	4	40
	A																				
1	10																				
2	20																				
3	30																				
4	40																				
	A																				
1	10																				
2	20																				
3	30																				
4	40																				
<b>Inicio</b>	<b>Fin</b>																				

### 2.4.2.62 SETPHONETIC

#### 2.4.2.62.1 Descripción

Permite crear objetos de tipo Phonetic para las celdas de un determinado rango.

#### 2.4.2.62.2 Sintaxis:

SetPhonetic

#### 2.4.2.62.3 Argumentos

No tiene.

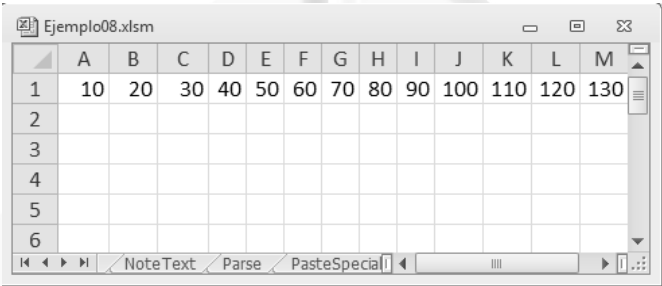
#### 2.4.2.62.4 Ejemplo

Código
<pre>Sub prueba_SetPhonetic()   Dim r As Range   Set r = Range("A1:A3")   r.SetPhonetic End Sub</pre>

2.4.2.63 SHOW

2.4.2.63.1 Descripción

Permite desplazar la ventana activa para visualizar el rango indicado. Para el ejemplo utilizado, vamos a suponer que disponemos de la siguiente hoja:



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	10	20	30	40	50	60	70	80	90	100	110	120	130
2													
3													
4													
5													
6													

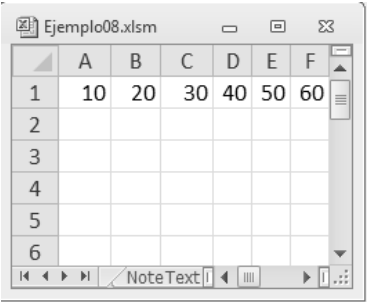
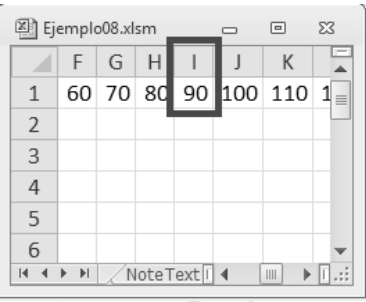
2.4.2.63.2 Sintaxis:

Show

2.4.2.63.3 Argumentos

No tiene.

2.4.2.63.4 Ejemplo

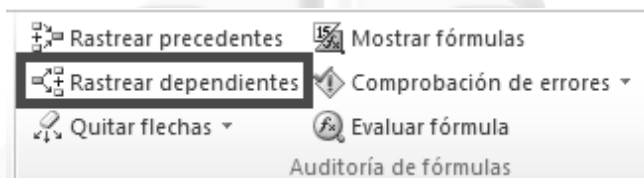
Código	
<pre>Sub prueba_Show()     Range("I1").Show End Sub</pre>	
Resultado	
	
Inicio	Fin



## 2.4.2.64 SHOWDEPENDENTS

### 2.4.2.64.1 Descripción

Permite visualizar las flechas que indican cuales son las celdas dependientes de la celda indicada. Equivale a utilizar la función **Rastrear Dependientes** del cuadro de controles de **Auditoría de fórmulas** de la pestaña **Fórmulas**:



Cada vez que se hace clic sobre esta opción, se van visualizando más niveles de dependencia (mas flechas). Para el ejemplo, vamos a suponer que disponemos de una hoja con la siguiente información:

	A	B	C
1	<b>Cantidad</b>	<b>Precio</b>	<b>Valor</b>
2	2	10	20
3	4	20	80
4	6	30	180
5	<b>12</b>		<b>280</b>

### 2.4.2.64.2 Sintaxis:

ShowDependents(Remove)

### 2.4.2.64.3 Argumentos

Argumento	Comentario
<b>Remove</b>	Si True, se van eliminando niveles de visualización de flechas.

### 2.4.2.64.4 Ejemplo

Código	
<pre> Sub prueba_ShowDependents()     Dim r As Range     Set r = Range("A2")     r.ShowDependents     MsgBox ("Nivel 1")     r.ShowDependents     MsgBox ("Nivel 2")     r.ShowDependents (True)     MsgBox ("Se elimina Nivel 2")     r.ShowDependents (True)     MsgBox ("Se elimina Nivel 1") End Sub </pre>	
Resultado	

## 2.4.2.65 SHOWERRORS

### 2.4.2.65.1 Descripción

Permite visualizar flechas que indican cuales son las celdas que están produciendo el error.

### 2.4.2.65.2 Sintaxis:

ShowErrors

### 2.4.2.65.3 Argumentos

No tiene.

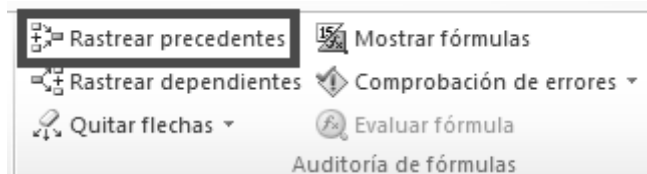
### 2.4.2.65.4 Ejemplo

Código																											
<pre>Sub prueba_ShowErrors()     Dim r As Range     Set r = Range("C5")     r.ShowErrors End Sub</pre>																											
Resultado																											
<table border="1"> <thead> <tr> <th></th><th>A</th><th>B</th><th>C</th></tr> </thead> <tbody> <tr> <td>1</td><td>Cantidad</td><td>Precio</td><td>Valor</td></tr> <tr> <td>2</td><td>2</td><td>10</td><td>=A2*B2</td></tr> <tr> <td>3</td><td>A</td><td>20</td><td>=A3*B3</td></tr> <tr> <td>4</td><td>6</td><td>30</td><td>=A4*B4</td></tr> <tr> <td>5</td><td>=SUMA(A2:A4)</td><td></td><td>=SUMA(C2:C4)</td></tr> </tbody> </table>					A	B	C	1	Cantidad	Precio	Valor	2	2	10	=A2*B2	3	A	20	=A3*B3	4	6	30	=A4*B4	5	=SUMA(A2:A4)		=SUMA(C2:C4)
	A	B	C																								
1	Cantidad	Precio	Valor																								
2	2	10	=A2*B2																								
3	A	20	=A3*B3																								
4	6	30	=A4*B4																								
5	=SUMA(A2:A4)		=SUMA(C2:C4)																								
<table border="1"> <thead> <tr> <th></th><th>A</th><th>B</th><th>C</th></tr> </thead> <tbody> <tr> <td>1</td><td>Cantidad</td><td>Precio</td><td>Valor</td></tr> <tr> <td>2</td><td>2</td><td>10</td><td>20</td></tr> <tr> <td>3</td><td>A</td><td>20</td><td>#¡VALOR!</td></tr> <tr> <td>4</td><td>6</td><td>30</td><td>180</td></tr> <tr> <td>5</td><td>8</td><td></td><td>#¡VALOR!</td></tr> </tbody> </table>					A	B	C	1	Cantidad	Precio	Valor	2	2	10	20	3	A	20	#¡VALOR!	4	6	30	180	5	8		#¡VALOR!
	A	B	C																								
1	Cantidad	Precio	Valor																								
2	2	10	20																								
3	A	20	#¡VALOR!																								
4	6	30	180																								
5	8		#¡VALOR!																								
<table border="1"> <thead> <tr> <th></th><th>A</th><th>B</th><th>C</th></tr> </thead> <tbody> <tr> <td>1</td><td>Cantidad</td><td>Precio</td><td>Valor</td></tr> <tr> <td>2</td><td>2</td><td>10</td><td>20</td></tr> <tr> <td>3</td><td>A</td><td>20</td><td>#¡VALOR!</td></tr> <tr> <td>4</td><td>6</td><td>30</td><td>180</td></tr> <tr> <td>5</td><td>8</td><td></td><td>#¡VALOR!</td></tr> </tbody> </table>					A	B	C	1	Cantidad	Precio	Valor	2	2	10	20	3	A	20	#¡VALOR!	4	6	30	180	5	8		#¡VALOR!
	A	B	C																								
1	Cantidad	Precio	Valor																								
2	2	10	20																								
3	A	20	#¡VALOR!																								
4	6	30	180																								
5	8		#¡VALOR!																								
Inicio		Fin																									

### 2.4.2.66 SHOWPRECEDENTS

#### 2.4.2.66.1 Descripción

Permite visualizar las flechas que indican cuales son las celdas precedentes a la celda indicada. Equivale a utilizar la función **Rastrear Precedentes** del cuadro de controles de **Auditoría de fórmulas** de la pestaña **Fórmulas**:



Cada vez que se hace clic sobre esta opción, se van visualizando más niveles de precedencia (mas flechas). Para el ejemplo, vamos a suponer que disponemos de una hoja con la siguiente información:

	A	B	C
1	Cantidad	Precio	Valor
2	2	10	20
3	4	20	80
4	6	30	180
5	12		280

2.4.2.66.2 Sintaxis:

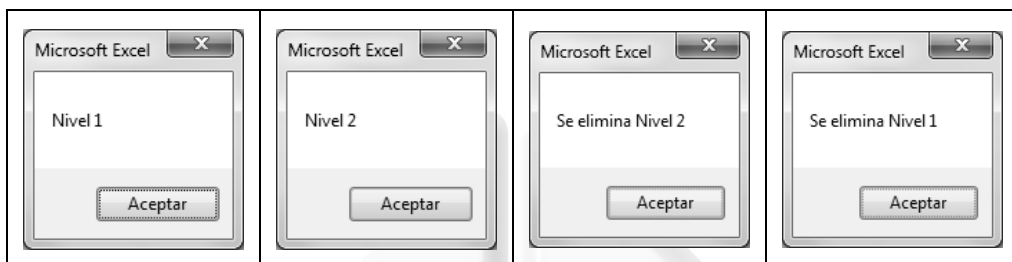
ShowPrecedents(Remove)

2.4.2.66.3 Argumentos

Argumento	Comentario
Remove	Si True, se van eliminando niveles de visualización de flechas.

2.4.2.66.4 Ejemplo

Código																																																																																																			
<pre>Sub prueba_ShowPrecedents()   Dim r As Range   Set r = Range("C5")   r.ShowPrecedents   MsgBox ("Nivel 1")   r.ShowPrecedents   MsgBox ("Nivel 2")   r.ShowPrecedents (True)   MsgBox ("Se elimina Nivel 2")   r.ShowPrecedents (True)   MsgBox ("Se elimina Nivel 1") End Sub</pre>																																																																																																			
Resultado																																																																																																			
<table><tr><th></th><th>A</th><th>B</th><th>C</th></tr><tr><td>1</td><td>Cantidad</td><td>Precio</td><td>Valor</td></tr><tr><td>2</td><td>2</td><td>10</td><td>20</td></tr><tr><td>3</td><td>4</td><td>20</td><td>80</td></tr><tr><td>4</td><td>6</td><td>30</td><td>180</td></tr><tr><td>5</td><td>12</td><td></td><td>280</td></tr></table>		A	B	C	1	Cantidad	Precio	Valor	2	2	10	20	3	4	20	80	4	6	30	180	5	12		280	<table><tr><th></th><th>A</th><th>B</th><th>C</th></tr><tr><td>1</td><td>Cantidad</td><td>Precio</td><td>Valor</td></tr><tr><td>2</td><td>← 2</td><td>← 10</td><td>← 20</td></tr><tr><td>3</td><td>← 4</td><td>← 20</td><td>← 80</td></tr><tr><td>4</td><td>← 6</td><td>← 30</td><td>← 180</td></tr><tr><td>5</td><td>12</td><td></td><td>280</td></tr></table>		A	B	C	1	Cantidad	Precio	Valor	2	← 2	← 10	← 20	3	← 4	← 20	← 80	4	← 6	← 30	← 180	5	12		280	<table><tr><th></th><th>A</th><th>B</th><th>C</th></tr><tr><td>1</td><td>Cantidad</td><td>Precio</td><td>Valor</td></tr><tr><td>2</td><td>2</td><td>10</td><td>20</td></tr><tr><td>3</td><td>4</td><td>20</td><td>80</td></tr><tr><td>4</td><td>6</td><td>30</td><td>180</td></tr><tr><td>5</td><td>12</td><td></td><td>280</td></tr></table>		A	B	C	1	Cantidad	Precio	Valor	2	2	10	20	3	4	20	80	4	6	30	180	5	12		280	<table><tr><th></th><th>A</th><th>B</th><th>C</th></tr><tr><td>1</td><td>Cantidad</td><td>Precio</td><td>Valor</td></tr><tr><td>2</td><td>2</td><td>10</td><td>20</td></tr><tr><td>3</td><td>4</td><td>20</td><td>80</td></tr><tr><td>4</td><td>6</td><td>30</td><td>180</td></tr><tr><td>5</td><td>12</td><td></td><td>280</td></tr></table>		A	B	C	1	Cantidad	Precio	Valor	2	2	10	20	3	4	20	80	4	6	30	180	5	12		280
	A	B	C																																																																																																
1	Cantidad	Precio	Valor																																																																																																
2	2	10	20																																																																																																
3	4	20	80																																																																																																
4	6	30	180																																																																																																
5	12		280																																																																																																
	A	B	C																																																																																																
1	Cantidad	Precio	Valor																																																																																																
2	← 2	← 10	← 20																																																																																																
3	← 4	← 20	← 80																																																																																																
4	← 6	← 30	← 180																																																																																																
5	12		280																																																																																																
	A	B	C																																																																																																
1	Cantidad	Precio	Valor																																																																																																
2	2	10	20																																																																																																
3	4	20	80																																																																																																
4	6	30	180																																																																																																
5	12		280																																																																																																
	A	B	C																																																																																																
1	Cantidad	Precio	Valor																																																																																																
2	2	10	20																																																																																																
3	4	20	80																																																																																																
4	6	30	180																																																																																																
5	12		280																																																																																																



## 2.4.2.67 SORT

### 2.4.2.67.1 Descripción

Permite ordenar un rango atendiendo a ciertos criterios y características.

### 2.4.2.67.2 Sintaxis:

Sort(Key1, Order1, Key2, Type, Order2, Key3, Order3, Header, OrderCustom, MatchCase, Orientation, SortMethod, DataOption1, DataOption2, DataOption3)

### 2.4.2.67.3 Argumentos

Argumento	Comentario		
<b>Key1</b>	Primer campo de ordenación.		
<b>Order1</b>	Indica el orden de ordenación y puede ser alguno de los valores de la constante <b>xlSortOrder</b> :		
	<b>xlAscending</b>	1	De menor a mayor.
	<b>xlDescending</b>	2	De mayor a menor.
<b>Key2</b>	Segundo campo de ordenación. No se puede utilizar en tablas dinámicas.		
<b>Type</b>	Indica qué elementos deben ordenarse.		
<b>Order2</b>	Criterio de ordenación para el segundo campo.		

<b>Key3</b>	Tercer campo de ordenación.	
<b>Order3</b>	Criterio de ordenación para el tercer campo.	
<b>Header</b>	Indica si la primera fila tiene información sobre encabezados. Puede ser alguno de los valores de la constante <b>XIYesNoGuess</b> :	
	0	<b>xlGuess</b> MS Excel intenta determinar qué encabezado se ha de utilizar.
	1	<b>xlYes</b> El encabezado está en la primera fila.
	2	<b>xlNo</b> No hay cabeceras.
<b>OrderCustom</b>	Posición en la lista de criterios de ordenación personalizados	
<b>MatchCase</b>	Si <b>True</b> , se distingue entre mayúsculas y minúsculas. Por defecto vale <b>False</b> .	
<b>Orientation</b>	Indica si la ordenación debe realizarse por filas o por columnas y puede ser alguno de los valores de la constante <b>XISortOrientation</b> :	
	1	<b>xlSortColumns</b> Por columnas.
	2	<b>xlSortRows</b> Por filas.
<b>SortMethod</b>	Indica el método de ordenación y puede ser alguno de los valores de la constante <b>XISortMethod</b> :	
	1	<b>xlPinYin</b> Criterio para la ordenación fonética de los caracteres en chino.
	2	<b>xlStroke</b> Ordenación por la cantidad de trazos de cada carácter.

<b>DataOption1</b>	Indica cómo se ha ordenar el texto en <b>key1</b> y puede ser alguno de los valores de la constante <b>XISortDataOption</b> :	
	1	<b>xlSortTextAsNumbers</b> Ordena los textos como datos numéricos.
	2	<b>xlSortNormal</b> Ordena números y textos por separado.
	No se puede utilizar en tablas dinámicas.	
<b>DataOption2</b>	Indica cómo se ha ordenar el texto en <b>key2</b> . No se puede utilizar en tablas dinámicas.	
<b>DataOption3</b>	Indica cómo se ha ordenar el texto en <b>key3</b> . No se puede utilizar en tablas dinámicas.	

#### 2.4.2.67.4 Ejemplo

Código																																			
<pre> Sub prueba_Sort()     Dim r As Range     Set r = Range("A1:B10")     r.Sort Key1:="NUMEROS", order1:=xlAscending, _         key2:="LETRAS", order2:=xlDescending, _         Header:=xlYes End Sub </pre>																																			
Resultado																																			
<table border="1"> <thead> <tr> <th></th><th>A</th><th>B</th></tr> </thead> <tbody> <tr><td>1</td><td>NUMEROS</td><td>LETRAS</td></tr> <tr><td>2</td><td>10</td><td>A</td></tr> <tr><td>3</td><td>20</td><td>B</td></tr> <tr><td>4</td><td>30</td><td>C</td></tr> <tr><td>5</td><td>10</td><td>D</td></tr> <tr><td>6</td><td>20</td><td>E</td></tr> <tr><td>7</td><td>31</td><td>A</td></tr> <tr><td>8</td><td>12</td><td>B</td></tr> <tr><td>9</td><td>20</td><td>C</td></tr> <tr><td>10</td><td>32</td><td>D</td></tr> </tbody> </table>				A	B	1	NUMEROS	LETRAS	2	10	A	3	20	B	4	30	C	5	10	D	6	20	E	7	31	A	8	12	B	9	20	C	10	32	D
	A	B																																	
1	NUMEROS	LETRAS																																	
2	10	A																																	
3	20	B																																	
4	30	C																																	
5	10	D																																	
6	20	E																																	
7	31	A																																	
8	12	B																																	
9	20	C																																	
10	32	D																																	
<b>Inicio</b>		<b>Fin</b>																																	

## 2.4.2.68 SORTSPECIAL

### 2.4.2.68.1 Descripción

Similar a **Sort** pero utilizando métodos de ordenación de idiomas del este asiático.

### 2.4.2.68.2 Sintaxis:

`SortSpecial(SortMethod, Key1, Order1, Type, Key2, Order2, Key3, Order3, Header, OrderCustom, MatchCase, Orientation, DataOption1, DataOption2, DataOption3)`

### 2.4.2.68.3 Argumentos

Argumento	Comentario								
Key1	Primer campo de ordenación.								
Order1	Indica el orden de ordenación y puede ser alguno de los valores de la constante <b>XISortOrder</b> : <table><tr><td>xlAscending</td><td>1</td><td>De menor a mayor.</td></tr><tr><td>xlDescending</td><td>2</td><td>De mayor a menor.</td></tr></table>			xlAscending	1	De menor a mayor.	xlDescending	2	De mayor a menor.
xlAscending	1	De menor a mayor.							
xlDescending	2	De mayor a menor.							
Key2	Segundo campo de ordenación. No se puede utilizar en tablas dinámicas.								
Type	Indica qué elementos deben ordenarse.								
Order2	Criterio de ordenación para el segundo campo.								
Key3	Tercer campo de ordenación.								
Order3	Criterio de ordenación para el tercer campo.								



<b>Header</b>	Indica si la primera fila tiene información sobre encabezados. Puede ser alguno de los valores de la constante <b>XIYesNoGuess</b> :		
	0	<b>xlGuess</b>	MS Excel intenta determinar qué encabezado se ha de utilizar.
	1	<b>xlYes</b>	El encabezado está en la primera fila.
	2	<b>xlNo</b>	No hay cabeceras.
<b>OrderCustom</b>	Posición en la lista de criterios de ordenación personalizados		
<b>MatchCase</b>	Si <b>True</b> , se distingue entre mayúsculas y minúsculas. Por defecto vale <b>False</b> .		
<b>Orientation</b>	Indica si la ordenación debe realizarse por filas o por columnas y puede ser alguno de los valores de la constante <b>XISortOrientation</b> :		
	1	<b>xlSortColumns</b>	Por columnas.
	2	<b>xlSortRows</b>	Por filas.
<b>SortMethod</b>	Indica el método de ordenación y puede ser alguno de los valores de la constante <b>XISortMethod</b> :		
	1	<b>xlPinYin</b>	Criterio para la ordenación fonética de los caracteres en chino.
	2	<b>xlStroke</b>	Ordenación por la cantidad de trazos de cada carácter.

Ra-Ma®

<b>DataOption1</b>	<p>Indica cómo se ha ordenar el texto en key1 y puede ser alguno de los gvalores de la constante XlSortDataOption:</p> <table><tr><td>1</td><td><b>xlSortTextAsNumbers</b></td><td>Ordena los textos como datos numéricos.</td></tr><tr><td>2</td><td><b>xlSortNormal</b></td><td>Ordena números y textos por separado.</td></tr></table> <p>No se puede utilizar en tablas dinámicas.</p>	1	<b>xlSortTextAsNumbers</b>	Ordena los textos como datos numéricos.	2	<b>xlSortNormal</b>	Ordena números y textos por separado.
1	<b>xlSortTextAsNumbers</b>	Ordena los textos como datos numéricos.					
2	<b>xlSortNormal</b>	Ordena números y textos por separado.					
<b>DataOption2</b>	<p>Indica cómo se ha ordenar el texto en key2. No se puede utilizar en tablas dinámicas.</p>						
<b>DataOption3</b>	<p>Indica cómo se ha ordenar el texto en key3. No se puede utilizar en tablas dinámicas.</p>						

#### 2.4.2.68.4 Ejemplo

Código																																																																				
<pre> Sub prueba_SortSpecial()     Dim r As Range     Set r = Range("A1:B10")     r.SortSpecial Key1:="NUMEROS", order1:=xlAscending, _     key2:="LETRAS", order2:=xlDescending, _     Header:=xlYes, SortMethod:=xlPinYin End Sub </pre>																																																																				
Resultado																																																																				
<table border="1"> <thead> <tr> <th></th><th>A</th><th>B</th></tr> </thead> <tbody> <tr><td>1</td><td>NUMEROS</td><td>LETRAS</td></tr> <tr><td>2</td><td>10</td><td>A</td></tr> <tr><td>3</td><td>20</td><td>B</td></tr> <tr><td>4</td><td>30</td><td>C</td></tr> <tr><td>5</td><td>10</td><td>D</td></tr> <tr><td>6</td><td>20</td><td>E</td></tr> <tr><td>7</td><td>31</td><td>A</td></tr> <tr><td>8</td><td>12</td><td>B</td></tr> <tr><td>9</td><td>20</td><td>C</td></tr> <tr><td>10</td><td>32</td><td>D</td></tr> </tbody> </table>		A	B	1	NUMEROS	LETRAS	2	10	A	3	20	B	4	30	C	5	10	D	6	20	E	7	31	A	8	12	B	9	20	C	10	32	D	<table border="1"> <thead> <tr> <th></th><th>A</th><th>B</th></tr> </thead> <tbody> <tr><td>1</td><td>NUMEROS</td><td>LETRAS</td></tr> <tr><td>2</td><td>10</td><td>D</td></tr> <tr><td>3</td><td>10</td><td>A</td></tr> <tr><td>4</td><td>12</td><td>B</td></tr> <tr><td>5</td><td>20</td><td>E</td></tr> <tr><td>6</td><td>20</td><td>C</td></tr> <tr><td>7</td><td>20</td><td>B</td></tr> <tr><td>8</td><td>30</td><td>C</td></tr> <tr><td>9</td><td>31</td><td>A</td></tr> <tr><td>10</td><td>32</td><td>D</td></tr> </tbody> </table>		A	B	1	NUMEROS	LETRAS	2	10	D	3	10	A	4	12	B	5	20	E	6	20	C	7	20	B	8	30	C	9	31	A	10	32	D	
	A	B																																																																		
1	NUMEROS	LETRAS																																																																		
2	10	A																																																																		
3	20	B																																																																		
4	30	C																																																																		
5	10	D																																																																		
6	20	E																																																																		
7	31	A																																																																		
8	12	B																																																																		
9	20	C																																																																		
10	32	D																																																																		
	A	B																																																																		
1	NUMEROS	LETRAS																																																																		
2	10	D																																																																		
3	10	A																																																																		
4	12	B																																																																		
5	20	E																																																																		
6	20	C																																																																		
7	20	B																																																																		
8	30	C																																																																		
9	31	A																																																																		
10	32	D																																																																		
<b>Inicio</b>		<b>Fin</b>																																																																		

## 2.4.2.69 SPEAK

### 2.4.2.69.1 Descripción

Permite que Excel vaya nombrando los valores y/o fórmulas existentes en el rango indicado.

### 2.4.2.69.2 Sintaxis:

Speak(SpeakDirection, SpeakFormulas)

### 2.4.2.69.3 Argumentos

Argumento	Comentario	
<b>SpeakDirection</b>	Permite seleccionar la dirección en la que serán leídas las celdas y puede ser alguno de los valores de la constante <b>xlSpeakDirection</b> :	
	0	<b>xlSpeakByRows</b> Leidas por filas.
	1	<b>xlSpeakByColumns</b> Leidas por columnas.
<b>SpeakFormulas</b>	Si <b>True</b> , lee las fórmulas que se halle en las celdas.	

### 2.4.2.69.4 Ejemplo

Código
<pre> Sub prueba_Speak()     Dim r As Range     Set r = Range("A1:D4")     ' Lee por filas y sólo valores (No fórmulas)     r.Speak     ' Lee por filas y lee fórmulas)     r.Speak SpeakDirection:=xlSpeakByColumns, SpeakFormulas:=True     Range("A7").Speak End Sub </pre>

Resultado									
	A	B	C	D		A	B	C	D
1	10	20	30	60	1	10	20	30	=SUMA(A1:C1)
2	11	21	31	63	2	11	21	31	=SUMA(A2:C2)
3	12	22	32	66	3	12	22	32	=SUMA(A3:C3)
4	33	63	93	189	4	=SUMA(A1:A3)	=SUMA(B1:B3)	=SUMA(C1:C3)	=SUMA(A4:C4)
5					5				
6	Hello Vito, how are you?				6	Hello Vito, how are you?			

Al ejecutar este procedimiento, si disponemos de unos altavoces o auriculares, observaremos como se lee el contenido de cada una de las celdas del rango indicado. La diferencia entre las 2 ejecuciones de **Speak** es que en la primera sólo se leen valores y en la segunda se leen fórmulas cuando las hay.

### 2.4.2.70 SPECIALCELLS

#### 2.4.2.70.1 Descripción

Proporciona un rango con las celdas que contienen un determinado tipo o valor indicado previamente.

#### 2.4.2.70.2 Sintaxis:

SpecialCells(Type, Value)

#### 2.4.2.70.3 Argumentos

Argumento	Comentario		
Type	Tipo de celdas a incluir. Puede tener alguno de los valores de la constante <b>XlCellType</b> :		
	-4175	<b>xlCellTypeSameValidation</b>	Celdas que tengan los mismos criterios de validación.
	-4174	<b>xlCellTypeAllValidation</b>	Celdas con criterios de validación
	-4173	<b>xlCellTypeSameFormatConditions</b>	Celdas que tengan el mismo formato.

	-4172	<b>xlCellTypeAllFormatConditions</b>	Celdas con cualquier formato.
	-4144	<b>xlCellTypeComments</b>	Celdas que contengan notas.
	-4123	<b>xlCellTypeFormulas</b>	Celdas que contengan fórmulas.
	2	<b>xlCellTypeConstants</b>	Celdas que contengan constantes.
	4	<b>xlCellTypeBlanks</b>	Celdas vacías.
	11	<b>xlCellTypeLastCell</b>	La última celda del rango.
	12	<b>xlCellTypeVisible</b>	Todas las celdas visibles.
<b>Value</b>	Determina los tipos de celdas a incluir en el resultado si <b>Type</b> es <b>xlCellTypeConstants</b> o <b>xlCellTypeFormulas</b> . Puede tener alguno de los valores de la constante <b>XlSpecialCellsValue</b> :		
	1	<b>xlNumbers</b>	
	2	<b>xlTextValues</b>	
	4	<b>xlLogical</b>	
	16	<b>xlErrors</b>	

#### 2.4.2.70.4 Ejemplo

Código
<pre> Sub prueba_SpecialCells()     Dim c, r As Range     Set r = Range("A1:D5")     For Each c In r.SpecialCells(xlCellTypeFormulas)         Debug.Print "Hay fórmulas en (" &amp; c.Address&amp; " ) -&gt; (" &amp; c.formula &amp; ")"     Next     For Each c In r.SpecialCells(xlCellTypeConstants, xlTextValues)         Debug.Print "Hay constantes de texto en (" &amp; c.Address&amp; " ) -&gt; (" &amp; c.formula &amp; ")"     Next     For Each c In r.SpecialCells(xlCellTypeConstants, xlNumbers)         Debug.Print "Hay constantes numéricas en (" &amp; c.Address&amp; " ) -&gt; (" &amp; c.formula &amp; ")"     Next End Sub </pre>

Resultado				
				Inmediato
	A	B	C	
1	Cantidad	Precio	Valor	Hay fórmulas en (\$A\$5) -> (=SUM(A2:A4))
2	2	10	20	Hay fórmulas en (\$C\$2) -> (=A2*B2)
3	4	20	80	Hay fórmulas en (\$C\$3) -> (=A3*B3)
4	6	30	180	Hay fórmulas en (\$C\$4) -> (=A4*B4)
5	12		280	Hay fórmulas en (\$C\$5) -> (=SUM(C2:C4))
				Hay constantes de texto en (\$A\$1) -> (Cantidad)
				Hay constantes de texto en (\$B\$1) -> (Precio)
				Hay constantes de texto en (\$C\$1) -> (Valor)
				Hay constantes numéricas en (\$A\$2) -> (2)
				Hay constantes numéricas en (\$B\$2) -> (10)
				Hay constantes numéricas en (\$A\$3) -> (4)
				Hay constantes numéricas en (\$B\$3) -> (20)
				Hay constantes numéricas en (\$A\$4) -> (6)
				Hay constantes numéricas en (\$B\$4) -> (30)

2.4.2.71 SUBTOTAL

2.4.2.71.1 Descripción

Crea subtotales para el rango (o la región actual, si el rango sólo tiene una celda).

2.4.2.71.2 Sintaxis:

Subtotal(GroupBy, Function, TotalList, Replace, PageBreaks, SummaryBelowData)

2.4.2.71.3 Argumentos



Argumento	Comentario		
GroupBy	Campo por el que se desea agrupar.		
Function	Función a utilizar. Puede ser alguno de los valores de la constante XlConsolidationFunction:		
	-4165	xlVarP	Variación (basada en toda la población).
	-4164	xlVar	Variación (basada en una muestra).

	-4157	xlSum	Sumar.						
	-4156	xlStDevP	Desviación estándar (basada en toda la población).						
	-4155	xlStDev	Desviación estándar (basada en una muestra).						
	-4149	xlProduct	Multiplicar.						
	-4139	xlMin	Mínimo.						
	-4136	xlMax	Máximo.						
	-4113	xlCountNums	Cuenta valores numéricos.						
	-4112	xlCount	Cuenta.						
	-4106	xlAverage	Promedio.						
	1000	xlUnknown	Sin función.						
TotalList	Campos sobre los que hay que aplicar la función de subtotal.								
Replace	Si True, se reemplazarán los posibles subtotales existentes. Por defecto es True.								
PageBreaks	Si True, se incluye un salto de página detrás de cada subtotal. Por defecto vale False.								
SummaryBelowData	Incluye datos de resumen relativos al subtotal. Puede tener alguno de los valores de la constante XISummaryRow: <table><tr><td>0</td><td>xlSummaryAbove</td><td>La fila de resumen estará sobre las filas de detalle en el esquema.</td></tr><tr><td>1</td><td>xlSummaryBelow</td><td>La fila de resumen estará debajo de las filas de detalle en el esquema.</td></tr></table>			0	xlSummaryAbove	La fila de resumen estará sobre las filas de detalle en el esquema.	1	xlSummaryBelow	La fila de resumen estará debajo de las filas de detalle en el esquema.
0	xlSummaryAbove	La fila de resumen estará sobre las filas de detalle en el esquema.							
1	xlSummaryBelow	La fila de resumen estará debajo de las filas de detalle en el esquema.							

2.4.2.71.4 Ejemplo

Supongamos una hoja con la siguiente información:

	A	B	C	D
1	Provincia	País	Importe	Unidades
2	Barcelona	España	80	16
3	Barcelona	España	150	30
4	Madrid	España	220	44
5	Madrid	España	230	46
6	Londres	Inglaterra	320	64
7	Manchester	Inglaterra	560	112
8	Lyon	Francia	155	31
9	Paris	Francia	740	148

Código 1	Código 2
<pre>Sub prueba_Subtotal()   Dim r As Range   Set r = Range("A1:D9")   r.Subtotal GroupBy:=2, _     Function:=xlCount, _     TotalList:=Array(4), _ SummaryBelowData:=xlSummaryAbove End Sub</pre>	<pre>Sub prueba_Subtotal2()   Dim r As Range   Set r = Range("A1:D9")   r.Subtotal GroupBy:=2, _     Function:=xlSum, _     TotalList:=Array(3, 4), _ SummaryBelowData:=xlSummaryBelow End Sub</pre>
Resultado 1	Resultado 2
	



## 2.4.2.72 TABLE

### 2.4.2.72.1 Descripción

Permite crear una tabla de datos basada en una serie de valores de entrada diversos.

### 2.4.2.72.2 Sintaxis:

Table(RowInput, ColumnInput)

### 2.4.2.72.3 Argumentos

Argumento	Comentario
<b>RowInput</b>	Celda usada como entrada de fila de la tabla.
<b>ColumnInput</b>	Celda usada como entrada de columna de la tabla.

### 2.4.2.72.4 Ejemplo

En el siguiente ejemplo, se calcula el valor de la longitud de diversas circunferencias a partir de varios radios de entrada. Para ello, definimos la fórmula para el cálculo de la misma en la esquina superior derecha de la tabla y usamos como celda de entrada de columna a la celda B1.

Código
<pre> Sub prueba_Table()     ' Radio de la circunferencia.     Range("B1").Value = 5     ' Fórmula para el cálculo de la longitud de la circunferencia     Range("B3").formula = "=2*PI()*B1"     ' Definición del rango para la tabla     Dim r As Range     Set r = Range("A3:B10")     r.Table ColumnInput:=Range("B1") End Sub </pre>

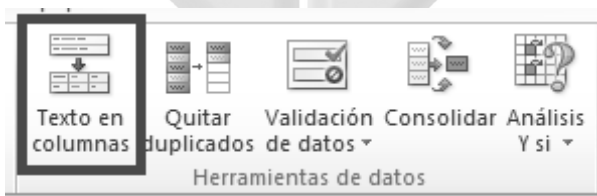
Ra-Ma®

Resultado				
	A	B		
1	Radio		1	Radio
2		Longitud	2	
3	Radios		3	Radios
4	10		4	10
5	12		5	12
6	14		6	14
7	16		7	16
8	18		8	18
9	20		9	20
10	25		10	25

### 2.4.2.73 TEXTTOCOLUMNS

#### 2.4.2.73.1 Descripción

Permite trocear u texto y distribuirlo en columnas. Equivale a utilizar la funcionalidad de Texto en columnas situada en el grupo de controles Herramientas de datos en la pestaña Datos:



#### 2.4.2.73.2 Sintaxis:

TextToColumns(Destination, DataType, TextQualifier, ConsecutiveDelimiter, Tab, Semicolon, Comma, Space, Other, OtherChar, FieldInfo, DecimalSeparator, ThousandsSeparator, TrailingMinusNumbers)

### 2.4.2.73.3 Argumentos

Argumento	Comentario		
Destination	Destino donde se colocará el resultado.		
DataType	Formato del texto a dividir. Puede tener alguno de los valores de la constante <b>xlTextParsingType</b> :		
	1	xlDelimited	Las columnas están separadas por un delimitador.
	2	xlFixedWidth	Las columnas poseen un ancho fijo.
TextQualifier	Indica si el texto se califica sin comillas, con comillas simples o con comillas dobles. Puede tener alguno de los valores de la constante <b>xlTextQualifier</b> :		
	-4142	xlTextQualifierNone	Sin delimitador.
	1	xlTextQualifierDoubleQuote	Comillas dobles.
	2	xlTextQualifierSingleQuote	Comillas simples.
ConsecutiveDelimiter	Si <b>True</b> , los delimitadores consecutivos se consideran como un solo delimitador.		
Tab	Si <b>True</b> , <b>DataType</b> tendrá el valor <b>xlDelimited</b> y el delimitador será un tabulador (ASCII(9)).		
Semicolon	Si <b>True</b> , <b>DataType</b> tendrá el valor <b>xlDelimited</b> y el delimitador será un punto y coma (;).		
Comma	Si <b>True</b> , <b>DataType</b> tendrá el valor <b>xlDelimited</b> y el delimitador será una coma (,).		
Space	Si <b>True</b> , <b>DataType</b> tendrá el valor <b>xlDelimited</b> y el delimitador será un espacio ( ).		

<b>Other</b>	Si <b>True</b> , <b>DataType</b> tendrá el valor <b>xlDelimited</b> y el delimitador será el carácter especificado en <b>OtherCar</b> .																														
<b>OtherChar</b>	Carácter a utilizar como delimitador si <b>Other</b> es <b>True</b> .																														
<b>FieldInfo</b>	<p>Matriz mediante la cual podemos analizar cada una de las columnas en las que se vaya a convertir cada línea. Se trata de una matriz de 2 elementos en la que el primer indica el número de columna y el segundo contendrá alguno de los valores de la constante <b>xlColumnDataType</b>:</p> <table><tr><td>1</td><td><b>xlGeneralFormat</b></td><td>General.</td></tr><tr><td>2</td><td><b>xlTextFormat</b></td><td>Texto.</td></tr><tr><td>3</td><td><b>xlMDYFormat</b></td><td>Fecha en formato MDA.</td></tr><tr><td>4</td><td><b>xlDMYFormat</b></td><td>Fecha en formato DMA.</td></tr><tr><td>5</td><td><b>xlYMDFormat</b></td><td>Fecha en formato AMD.</td></tr><tr><td>6</td><td><b>xlMYDFormat</b></td><td>Fecha en formato MAD.</td></tr><tr><td>7</td><td><b>xlDYMFormat</b></td><td>Fecha en formato DAM.</td></tr><tr><td>8</td><td><b>xlYDMFormat</b></td><td>Fecha en formato ADM.</td></tr><tr><td>9</td><td><b>xlSkipColumn</b></td><td>No analizar.</td></tr><tr><td>10</td><td><b>xlEMDFormat</b></td><td>Fecha en formato EMD.</td></tr></table>	1	<b>xlGeneralFormat</b>	General.	2	<b>xlTextFormat</b>	Texto.	3	<b>xlMDYFormat</b>	Fecha en formato MDA.	4	<b>xlDMYFormat</b>	Fecha en formato DMA.	5	<b>xlYMDFormat</b>	Fecha en formato AMD.	6	<b>xlMYDFormat</b>	Fecha en formato MAD.	7	<b>xlDYMFormat</b>	Fecha en formato DAM.	8	<b>xlYDMFormat</b>	Fecha en formato ADM.	9	<b>xlSkipColumn</b>	No analizar.	10	<b>xlEMDFormat</b>	Fecha en formato EMD.
1	<b>xlGeneralFormat</b>	General.																													
2	<b>xlTextFormat</b>	Texto.																													
3	<b>xlMDYFormat</b>	Fecha en formato MDA.																													
4	<b>xlDMYFormat</b>	Fecha en formato DMA.																													
5	<b>xlYMDFormat</b>	Fecha en formato AMD.																													
6	<b>xlMYDFormat</b>	Fecha en formato MAD.																													
7	<b>xlDYMFormat</b>	Fecha en formato DAM.																													
8	<b>xlYDMFormat</b>	Fecha en formato ADM.																													
9	<b>xlSkipColumn</b>	No analizar.																													
10	<b>xlEMDFormat</b>	Fecha en formato EMD.																													
<b>DecimalSeparator</b>	Separador decimal. Por defecto lo obtiene de la configuración del sistema.																														
<b>ThousandsSeparator</b>	Separador de miles. Por defecto lo obtiene de la configuración del sistema.																														
<b>TrailingMinusNumbers</b>	Números que empiezan con signo menos.																														

2.4.2.73.4 Ejemplo

Código

```

Sub prueba_TextToColumns()
    Dim r As Range
    Set r = Range("A1:A1")
    r.TextToColumns Destination:=Range("A2"), comma:=True
    Set r = Range("A3:A3")
    r.TextToColumns Destination:=Range("A4"), _
    DataType:=xlFixedWidth, _
    FieldInfo:=Array(Array(0, 1), _
    Array(3, 2), _
    Array(9, 1))
End Sub

```

Resultado

	A	B	C	D	E
1	uno,dos,tres,cuatro,cinco				
2					
3	ABCDEFGHIJKLM				
4					

	A	B	C	D	E
1	uno,dos,tres,cuatro,cinco				
2	uno	dos	tres	cuatro	cinco
3	ABCDEFGHIJKLM				
4	ABC	DEFGHI	JKL		

Inicio

Fin

En este ejemplo, puede observarse como el texto existente en la celda A1, se divide usando la coma (,) como separador y deja una palabra en cada celda.

También puede observarse como se trocea la cadena existente en la celda A3, indicando como FieldInfo un Array compuesto a su vez por 3 Arrays en los que se indica:

Array(0,1)	Posición inicial del trozo: 0, tipo de dato: 1 (General).
Array(2,2)	Posición inicial del trozo: 2, tipo de dato: 2 (Texto).
Array(9,1)	Posición inicial del trozo: 9, tipo de dato: 1 (General).

2.4.2.74 UNGROUP

2.4.2.74.1 Descripción

Este método desagrupa los elementos que se hallen incluidos en un grupo del rango indicado.

### 2.4.2.74.2 Sintaxis:

Ungroup

### 2.4.2.74.3 Argumentos

No tiene.

### 2.4.2.74.4 Ejemplo

Código

```
Sub prueba_Ungroup()  
    Dim r As Range  
    Set r = Range("E2")  
    r.Ungroup  
End Sub
```

Resultado

	A	B	CD	E	F
1	Fecha	Importe		Etiquetas de fila	Suma de Importe
2	01/01/2012	100		ene	600
3	02/01/2012	200		feb	2200
4	03/01/2012	300		mar	2700
5	01/02/2012	400		Total general	5500
6	02/02/2012	500			
7	03/02/2012	600			
8	04/02/2012	700			
9	01/03/2012	800			
10	02/03/2012	900			
11	03/03/2012	1000			

	A	B	CD	E	F
1	Fecha	Importe		Etiquetas de fila	Suma de Importe
2	01/01/2012	100		01/01/2012	100
3	02/01/2012	200		02/01/2012	200
4	03/01/2012	300		03/01/2012	300
5	01/02/2012	400		01/02/2012	400
6	02/02/2012	500		02/02/2012	500
7	03/02/2012	600		03/02/2012	600
8	04/02/2012	700		04/02/2012	700
9	01/03/2012	800		01/03/2012	800
10	02/03/2012	900		02/03/2012	900
11	03/03/2012	1000		03/03/2012	1000
12				Total general	5500

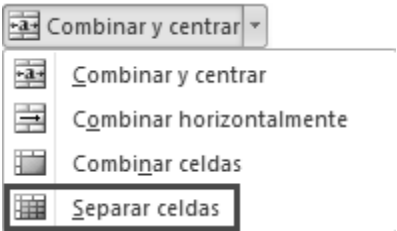
Inicio

Fin

## 2.4.2.75 UNMERGE

### 2.4.2.75.1 Descripción

Permite separar en celdas independientes un área combinada. Equivale a utilizar la función **Separar celdas** que se halla ubicada en el grupo de controles **Combinar y centrar** dentro del apartado **Alineación** de la pestaña **Inicio**.



2.4.2.75.2 Sintaxis:

UnMerge

2.4.2.75.3 Argumentos

No tiene.

2.4.2.75.4 Ejemplo

Código																													
<pre>Sub prueba_UnMerge()     Range("A1:D4").UnMerge End Sub</pre>																													
Resultado																													
<table><tr><th></th><th>A</th><th>B</th><th>C</th><th>D</th></tr><tr><td>1</td><td>uno</td><td></td><td></td><td></td></tr><tr><td>2</td><td>dos</td><td></td><td></td><td></td></tr><tr><td>3</td><td>tres</td><td></td><td></td><td></td></tr><tr><td>4</td><td>cuatro</td><td></td><td></td><td></td></tr></table>						A	B	C	D	1	uno				2	dos				3	tres				4	cuatro			
	A	B	C	D																									
1	uno																												
2	dos																												
3	tres																												
4	cuatro																												
<table><tr><th></th><th>A</th><th>B</th><th>C</th><th>D</th></tr><tr><td>1</td><td>uno</td><td></td><td></td><td></td></tr><tr><td>2</td><td>dos</td><td></td><td></td><td></td></tr><tr><td>3</td><td>tres</td><td></td><td></td><td></td></tr><tr><td>4</td><td>cuatro</td><td></td><td></td><td></td></tr></table>						A	B	C	D	1	uno				2	dos				3	tres				4	cuatro			
	A	B	C	D																									
1	uno																												
2	dos																												
3	tres																												
4	cuatro																												
<table><tr><td colspan="3">Inicio</td><td colspan="2">Fin</td></tr></table>					Inicio			Fin																					
Inicio			Fin																										





## CONTROLES

---

### 3.1 DE FORMULARIO

#### 3.1.1 CommandButton

##### 3.1.1.1 PROPIEDADES

Accelerator	Creator	Locked	PrintObject	WordWrap
Application	Enabled	MouseIcon	ProgID	ZOrder
AutoLoad	Font	MousePointer	Shadow	
Autosize	ForeColor	Name	ShapeRange	
AutoUpdate	Height	Object	SourceName	
BackColor	Index	OLEType	TakeFocusOnClick	
BackStyle	Interior	Parent	Top	®
Border	Left	Picture	TopLeftCell	
BottomRightCell	LinkedCell	PicturePosition	Visible	
Caption	ListFillRange	Placement	Width	

### 3.1.1.2 MÉTODOS

Activate	CopyPicture	Duplicate	Update
BringToFront	Cut	Select	Verb
Copy	Delete	SendToBack	

### 3.1.1.3 EVENTOS

BeforeDragOver	Error	KeyUp	MouseUp
BeforeDropOrPaste	GotFocus	LostFocus	
Click	KeyDown	MouseDown	
DblClick	KeyPress	MouseMove	

## 3.1.2 ComboBox

### 3.1.2.1 PROPIEDADES

Application	ColumnHeads	Left	MousePointer	SpecialEffect
AutoLoad	ColumnWidths	LineCount	Name	Style
AutoSize	Creator	LinkedCell	Object	Text
AutoTab	CurTargetX	List	OLEType	TextAlign
AutoUpdate	CurX	ListCount	Parent	TextColumn
AutoWordSelect	DragBehavior	ListFillRange	Placement	TextLength
BackColor	DropButtonStyle	ListIndex	PrintObject	Top
BackStyle	Enabled	ListRows	progID	TopIndex
Border	EnterFieldBehavior	ListStyle	SelectionMargin	TopLeftCell

BorderColor	Font	ListWidth	SelLength	Value
BorderStyle	ForeColor	Locked	SelStart	Visible
BottomRightCell	Height	MatchEntry	SelText	Width
BoundColumn	HideSelection	MatchFound	Shadow	WordWrap
CanPaste	IMEMode	MatchRequired	ShapeRange	ZOrder
Column	Index	MaxLength	ShowDropDownWhen	
ColumnCount	Interior	MouseIcon	SourceName	

### 3.1.2.2 MÉTODOS

Activate	Copy	DropDown	Select
AddItem	CopyPicture	Duplicate	SendToBack
BringToFront	Cut	Paste	Update
Clear	Delete	RemoveItem	Verb

### 3.1.2.3 EVENTOS

BeforeDragOver	DbClick	KeyDown	MouseDown
BeforeDropOrPaste	DropButtonClick	KeyPress	MouseMove
Change	Error	KeyUp	MouseUp
Click	GotFocus	LostFocus	

Ra-Ma®

### 3.1.3 CheckBox

#### 3.1.3.1 PROPIEDADES

Accelerator	Caption	LinkedCell	PicturePosition	TopLeftCell
Alignment	Creator	ListFillRange	Placement	TripleState
Application	Enabled	Locked	PrintObject	Value
AutoLoad	Font	MouseIcon	progID	Visible
AutoSize	ForeColor	MousePointer	Shadow	Width
AutoUpdate	GroupName	Name	ShapeRange	WordWrap
BackColor	Height	Object	SourceName	ZOrder
BackStyle	Index	OLEType	SpecialEffect	
Border	Interior	Parent	TextAlign	
BottomRightCell	Left	Picture	Top	

#### 3.1.3.2 MÉTODOS

Activate	Copy	Delete	SendToBack
BringToFront	CopyPicture	Duplicate	Update
Clear	Cut	Select	Verb

Ra-Ma®

### 3.1.3.3 EVENTOS

BeforeDragOver	DblClick	KeyPress	MouseUp
BeforeDropOrPaste	Error	KeyUp	
Change	GotFocus	MouseDown	
Click	KeyDown	MouseMove	

### 3.1.4 ListBox

#### 3.1.4.1 PROPIEDADES

Application	ColumnWidths	ListCount	Parent	Top
AutoLoad	Creator	ListFillRange	Placement	TopIndex
AutoUpdate	Enabled	ListIndex	PrintObject	TopLeftCell
BackColor	Font	ListStyle	progID	Value
Border	ForeColor	Locked	Selected	Visible
BorderColor	Height	MatchEntry	Shadow	Width
BorderStyle	IMEMode	MouseIcon	ShapeRange	ZOrder
BottomRightCell	IntegralHeight	MousePointer	SourceName	
BoundColumn	Interior	MultiSelect	SpecialEffect	
Column	Left	Name	Text	
ColumnCount	LinkedCell	Object	TextAlign	
ColumnHeads	List	OLEType	TextColumn	

### 3.1.4.2 MÉTODOS

Activate	Copy	Duplicate	Update
AddItem	CopyPicture	RemoveItem	Verb
BringToFront	Cut	Select	
Clear	Delete	SendToBack	

### 3.1.4.3 EVENTOS

BeforeDragOver	Click	KeyDown	MouseDown
BeforeDropOrPaste	DblClick	KeyPress	MouseMove
BeforUpdate	Error	KeyUp	MouseUp
Change	GotFocus	LostFocus	

## 3.1.5 TextBox

### 3.1.5.1 PROPIEDADES

Application	CurLine	Interior	PasswordChar	Style
AutoLoad	CurTargetX	Left	Placement	TabKeyBehavior
AutoSize	CurX	LineCount	PrintObject	TakeFocusOnClick
AutoTab	DragBehavior	LinkedCell	progID	Text
AutoUpdate	Enabled	ListFillRange	ScrollBars	TextAlign
AutoWordSelect	EnterFieldBehavior	Locked	SelectionMargin	TextColumn
BackColor	EnterKeyBehavior	MaxLength	SelLength	TextLength

BackStyle	Font	MouseIcon	SelStart	Top
Border	ForeColor	MousePointer	SelText	TopLeftCell
BorderColor	Height	MultiLine	Shadow	Value
BorderStyle	HideSelection	Name	ShapeRange	Visible
BottomRightCell	IMEMode	Object	ShowDropDownWhen	Width
CanPaste	Index	OLEType	SourceName	WordWrap
Creator	IntegralHeight	Parent	SpecialEffect	ZOrder

### 3.1.5.2 MÉTODOS

Activate	CopyPicture	Duplicate	SendToBack
BringToFront	Cut	Paste	Update
Copy	Delete	Select	Verb

### 3.1.5.3 EVENTOS

BeforeDragOver	DropButtonClick	KeyPress	MouseMove
BeforeDropOrPaste	Error	KeyUp	MouseUp
Change	GotFocus	LostFocus	
DblClick	KeyDown	MouseDown	

Ra-Ma®

## 3.1.6 ScrollBar

### 3.1.6.1 PROPIEDADES

Application	ForeColor	Max	Placement	TopLeftCell
AutoLoad	Height	Min	PrintObject	Value
AutoUpdate	Index	MouseIcon	progID	Visible
BackColor	Interior	MousePointer	ProportionalThumb	Width
Border	LargeChange	Name	Shadow	ZOrder
BottomRightCell	Left	Object	ShapeRange	
Creator	LinkedCell	OLEType	SmallChange	
Delay	ListFillRange	Orientation	SourceName	
Enabled	Locked	Parent	Top	

### 3.1.6.2 MÉTODOS

Activate	CopyPicture	Duplicate	Update
BringToFront	Cut	Select	Verb
Copy	Delete	SendToBack	

### 3.1.6.3 EVENTOS

BeforeDragOver	Error	KeyPress	Scroll
BeforeDropOrPaste	GotFocus	KeyUp	
Change	KeyDown	LostFocus	



### 3.1.7 SpinButton

#### 3.1.7.1 PROPIEDADES

Application	Enabled	Locked	Orientation	SourceName
AutoLoad	ForeColor	Max	Parent	Top
AutoUpdate	Height	Min	Placement	TopLeftCell
BackColor	Index	MouseIcon	PrintObject	Value
Border	Interior	MousePointer	prodID	Visible
BottomRightCell	Left	Name	Shadow	Width
Creator	LinkedCell	Object	ShapeRange	ZOrder
Delay	ListFillRange	OLEType	SmallChange	

#### 3.1.7.2 MÉTODOS

Activate	CopyPicture	Duplicate	Update
BringToFront	Cut	Select	Verb
Copy	Delete	SendToBack	

#### 3.1.7.3 EVENTOS

BeforeDragOver	Error	KeyPress	SpinDown
BeforeDropOrPaste	GotFocus	KeyUp	SpinUp
Change	KeyDown	LostFocus	

## 3.1.8 OptionButton

### 3.1.8.1 PROPIEDADES

Accelerator	Caption	LinkedCell	PicturePosition	TopLeftCell
Alignment	Creator	ListFillRange	Placement	TripleState
Application	Enabled	Locked	PrintObject	Value
AutoLoad	Font	MouseIcon	progID	Visible
Autosize	ForeColor	MousePointer	Shadow	Width
AutoUpdate	GroupName	Name	ShapeRange	WordWrap
BackColor	Height	Object	SourceName	ZOrder
BackStyle	Index	OLEType	SpecialEffect	
Border	Interior	Parent	TextAlign	
BottomRightCell	Left	Picture	Top	

### 3.1.8.2 MÉTODOS

Activate	CopyPicture	Duplicate	Update
BringToFront	Cut	Select	Verb
Copy	Delete	SendToBack	

### 3.1.8.3 EVENTOS

BeforeDragOver	DbClick	KeyPress	MouseMove
BeforeDropOrPaste	Error	KeyUp	MouseUp
Change	GotFocus	LostFocus	
Click	KeyDown	MouseDown	

## 3.1.9 Label

### 3.1.9.1 PROPIEDADES

Accelerator	BorderStyle	Interior	OLEType	SourceName
Application	BottomRightCell	Left	Parent	SpecialEffect
AutoLoad	Caption	LinkedCell	Picture	TextAlign
AutoSize	Creator	ListFillRange	PicturePosition	Top
AutoUpdate	Enabled	Locked	Placement	TopLeftCell
BackColor	Font	MouseIcon	PrintObject	Visible
BackStyle	ForeColor	MousePointer	progID	Width
Border	Height	Name	Shadow	WordWrap
BorderColor	Index	Object	ShapeRange	ZOrder

### 3.1.9.2 MÉTODOS

Activate	CopyPicture	Duplicate	Update
BringToFront	Cut	Select	Verb
Copy	Delete	SendToBack	

### 3.1.9.3 EVENTOS

BeforeDragOver	DbClick	LostFocus	MouseUp
BeforeDropOrPaste	Error	MouseDown	
Click	GotFocus	MouseMove	

## 3.1.10 Image

### 3.1.10.1 PROPIEDADES

Application	BottomRightCell	Locked	PictureSizeMode	Top
AutoLoad	Creator	MouseIcon	PictureTiling	TopLeftCell
Autosize	Enabled	MousePointer	Placement	Visible
AutoUpdate	Height	Name	PrintObject	Width
BackColor	Index	Object	progID	ZOrder
BackStyle	Interior	OLEType	Shadow	
Border	Left	Parent	ShapeRange	
BorderColor	LinkedCell	Picture	SourceName	
BorderStyle	ListFillRange	PictureAlignment	SpecialEffect	

### 3.1.10.2 MÉTODOS

Activate	CopyPicture	Duplicate	Update
BringToFront	Cut	Select	Verb
Copy	Delete	SendToBack	

### 3.1.10.3 EVENTOS

BeforeDragOver	DbClick	LostFocus	MouseUp
BeforeDropOrPaste	Error	MouseDown	
Click	GotFocus	MouseMove	

### 3.1.11 ToggleButton

#### 3.1.11.1 PROPIEDADES

Accelerator	Caption	LinkedCell	PicturePosition	TopLeftCell
Alignment	Creator	ListFillRange	Placement	TripleState
Application	Enabled	Locked	PrintObject	Value
AutoLoad	Font	MouseIcon	progID	Visible
Autosize	ForeColor	MousePointer	Shadow	Width
AutoUpdate	GroupName	Name	ShapeRange	WordWrap
BackColor	Height	Object	SourceName	ZOrder
BackStyle	Index	OLEType	SpecialEffect	
Border	Interior	Parent	TextAlign	
BottomRightCell	Left	Picture	Top	

#### 3.1.11.2 MÉTODOS

Activate	CopyPicture	Duplicate	Update
BringToFront	Cut	Select	Verb
Copy	Delete	SendToBack	

#### 3.1.11.3 EVENTOS

BeforeDragOver	DbClick	KeyPress	MouseMove
BeforeDropOrPaste	Error	KeyUp	MouseUp
Change	GotFocus	LostFocus	
Click	KeyDown	MouseDown	

### 3.1.12 Resumen de propiedades

Propiedad	Comentario	Valores						
Accelerator	Define o consulta la tecla de aceleración de un control.	String						
Alignment	<p>Define o consulta la posición del control en relación a su título. Puede ser alguno de los valores de la constante fmAlignment:</p> <table border="1"> <tr> <td>0</td><td><b>fmAlignmentLeft</b></td><td>Coloca el título a la izquierda del control.</td></tr> <tr> <td>1</td><td><b>FmAlignment-Right</b></td><td>Coloca el título a la derecha del control (predeterminado).</td></tr> </table>	0	<b>fmAlignmentLeft</b>	Coloca el título a la izquierda del control.	1	<b>FmAlignment-Right</b>	Coloca el título a la derecha del control (predeterminado).	fmAlignment
0	<b>fmAlignmentLeft</b>	Coloca el título a la izquierda del control.						
1	<b>FmAlignment-Right</b>	Coloca el título a la derecha del control (predeterminado).						
Application	Proporciona un objeto que representa la aplicación.	Application						
AutoLoad	Si True, el objeto se carga automáticamente cuando se abre el libro en el que se halla.	Boolean						
Autosize	Indica si el objeto ha de cambiar su tamaño automáticamente para mostrar su contenido.	Boolean						
AutoTab	Indica si se ha de producir una tabulación automática cuando el usuario sobrepasa el máximo de caracteres permitido en la parte del cuadro de texto.	Boolean						

AutoUpdate	Indica si el objeto se actualiza automáticamente al cambiar el origen.	Boolean						
AutoWordSelect	Indica si la selección ha de extenderse a una palabra o a un carácter.	Boolean						
BackColor	Color de fondo. La propiedad BackStyle debe establecerse a fmBackStyleOpaque. Puede definirse usando la función RGB (rojo, verde y azul).	Long						
BackStyle	Define el estilo del fondo. Puede ser alguno de los valores de la constante fmBackStyle <table border="1"> <tr> <td>0</td><td><b>fmBackStyle-Transparent</b></td><td>El fondo es transparente.</td></tr> <tr> <td>1</td><td><b>fmBackStyle-Opaque</b></td><td>El fondo es opaco. Valor por defecto.</td></tr> </table>	0	<b>fmBackStyle-Transparent</b>	El fondo es transparente.	1	<b>fmBackStyle-Opaque</b>	El fondo es opaco. Valor por defecto.	fmBackStyle
0	<b>fmBackStyle-Transparent</b>	El fondo es transparente.						
1	<b>fmBackStyle-Opaque</b>	El fondo es opaco. Valor por defecto.						
Border	Objeto border del control.	Border						
BorderColor	Indica el color del borde de un objeto.	Long						
BorderStyle	Indica el tipo de borde de un objeto. Puede ser alguno de los valores de la constante fmBorderStyle: <table border="1"> <tr> <td>0</td><td><b>fmBorderStyle-None</b></td><td>El control no tiene una línea de borde visible.</td></tr> <tr> <td>1</td><td><b>fmBorderStyle-Single</b></td><td>El control tiene un borde de una sola línea (predeterminado)</td></tr> </table>	0	<b>fmBorderStyle-None</b>	El control no tiene una línea de borde visible.	1	<b>fmBorderStyle-Single</b>	El control tiene un borde de una sola línea (predeterminado)	fmBorderStyle
0	<b>fmBorderStyle-None</b>	El control no tiene una línea de borde visible.						
1	<b>fmBorderStyle-Single</b>	El control tiene un borde de una sola línea (predeterminado)						

BottomRightCell	Proporciona un objeto que representa la esquina inferior derecha del objeto.	Range
BoundColumn	Indica el elemento o columna que se utilizará como valor del control cuando un usuario seleccione una fila. Por ejemplo, si una fila contiene 3 columnas y se indica 2, el valor del objeto será el correspondiente a la columna 2 de la fila seleccionada.	Variant
CanPaste	Indica si se puede efectuar un pegado desde el portapapeles.	Boolean
Caption	Descripción del botón.	String
Column	Indica uno o varios elementos de un control.	
ColumnCount	Indica el número de columnas a mostrar. Si se indica -1, se muestran todas las columnas disponibles.	Long
ColumnHeads	Muestra una fila de títulos de columna.	Boolean
ColumnWidths	Indica el ancho de cada una de las columnas del control. Los diferentes anchos se indican usando el punto y coma (;) como separador. Por ejemplo 50; 70; 60.	String
Creator	Proporciona un entero de 32 bits que indica cual fue la aplicación creadora del objeto.	Integer
CurLine	Indica la línea actual del control.	Long ®
CurTargetX	Indica la posición horizontal preferida del punto de inserción en un control medida en décimas de milímetro.	Long



CurX	Indica la posición horizontal actual del punto de inserción en un control medida en décimas de milímetro.	Long						
Delay	Indica el tiempo de retardo que se produce entre los eventos SpinUp, SpinDown y Change. Al hacer clic y presionar un botón, el primer evento se realiza inmediatamente pero, la segunda ocurrencia del evento será de 5 veces el valor de Delay. Los siguientes eventos se iniciarán después según lo indicado en Delay. Por defecto Delay vale 50 milisegundos.	Long						
DragBehavior	Indica si la característica de arrastrar y colocar está habilitada. Puede ser alguno de los valores de la constante fmDragBehavior: <table border="1"> <tr> <td>0</td><td><b>fmDragBehavior-Disabled</b></td><td>No permite arrastrar y colocar (predeterminado).</td></tr> <tr> <td>1</td><td><b>FmDragBehavior-Enabled</b></td><td>Permite arrastrar y colocar.</td></tr> </table>	0	<b>fmDragBehavior-Disabled</b>	No permite arrastrar y colocar (predeterminado).	1	<b>FmDragBehavior-Enabled</b>	Permite arrastrar y colocar.	fmDragBehavior
0	<b>fmDragBehavior-Disabled</b>	No permite arrastrar y colocar (predeterminado).						
1	<b>FmDragBehavior-Enabled</b>	Permite arrastrar y colocar.						
DropButtonStyle	Indica el símbolo a mostrar en el botón que despliega la lista de valores. Puede ser alguno de los valores de la constante fmDropButtonStyle:	fmDropButtonStyle						

	0	<b>fmDropButtonStyle-Plain</b>	Botón sin símbolo.
	1	<b>fmDropButtonStyle-Arrow</b>	Flecha hacia abajo (Valor por defecto).
	2	<b>fmDropButtonStyle-Ellipsis</b>	Puntos suspensivos (...).
	3	<b>fmDropButtonStyle-Reduce</b>	Muestra una línea horizontal.
Enabled	Indica si el objeto puede recibir el foco y si puede responder a eventos generados por un usuario.		Boolean
EnterFieldBehavior	Indica cómo se selecciona el texto y puede ser alguno de los valores de la constante fmEnterFieldBehavior:		fmEnterFieldBehavior
	0	<b>fmEnterFieldBehavior-SelectAll</b>	Selección a todo el contenido al entrar en el control (valor por defecto).
	1	<b>fmEnterFieldBehavior-RecallSelection</b>	Mantiene la selección.
EnterKeyBehavior	Define el comportamiento de la pulsación de la tecla ENTER. Si True, al pulsar ENTER se crea una nueva línea. Si False, el foco se mueve al siguiente control situado en el orden de tabulación.		Boolean

Font	Permite definir o consultar el tipo de fuente asociada al objeto.	Font
ForeColor	Color del primer plano del objeto. Puede definirse usando la función RGB (rojo, verde y azul) o un valor de tipo Long que represente un color.	Long
GroupName	Define o consulta un nombre de grupo para una serie de controles.	String
Height	Alto del objeto en puntos.	Single
HideSelection	Indica si se ha de mantener resaltado el texto seleccionado cuando el control pierde el foco.	Booleann
IMEMode	Indica el modo de tiempo de ejecución del editor de métodos de entrada (IME) y se utiliza sólo para aplicaciones escritas para Extremo Oriente.	fmIMEMode
Index	Indica el número de índice que ocupa el objeto dentro de una colección de controles.	Long
IntegralHeight	Indica si el control muestra las líneas completas o las muestra parciales. Si True, el cuadro cambia de tamaño automáticamente para poder mostrar las filas completas.	Boolean
Interior	Proporciona un objeto de tipo Interior que representa el interior de un objeto.	Interior
LargeChange	Indica la cantidad de desplazamiento que se ha de producir al hacer clic entre el cuadro de desplazamiento y la flecha.	Long <sup>®</sup>

Left	Distancia entre el borde izquierdo del objeto que lo contiene y el propio control.		Single
LineCount	Indica el número de líneas de texto del control.		Integer
LinkedCell	Define o consulta el rango asociado al control y que contiene el valor del mismo.		String
List	Define o consulta las entradas de la lista del control.		Matriz o conjunto de elementos
ListCount	Indica el número de entradas de la lista.		Long
ListFillRange	Define o consulta el rango utilizado para rellenar el cuadro de lista indicado.		String
ListIndex	Indica el elemento seleccionado actualmente en el control.		Variant
ListRows	Indica el número máximo de filas que se muestran.		Long
ListStyle	Indica la apariencia de la lista de elementos en el control. Puede ser alguno de los valores de la constante fmListStyle:		fmListStyle
	0	<b>fmListStylePlain</b>	Se muestra con el fondo de los elementos resaltado.
	1	<b>fmListStyleOption</b>	Muestra OptionButtons o CheckBox para una lista de selección múltiple (Valor por defecto).
ListWidth	Indica el ancho de la lista en el control.		Variant

Locked	Indica si el control puede ser editado	Boolean
MatchEntry	Define o consulta la forma de cómo se realiza la búsqueda en el control cuando el usuario escribe. Puede ser alguno de los valores de la constante fmMatchEntry:	fmMatchEntry
	0 <b>fmMatchEntryFirst-Letter</b>	El control busca entrada que empiece con el carácter introducido y así sucesivamente
	1 <b>fmMatchEntryComplete</b>	Según se van escribiendo caracteres se busca una entrada que coincida con todos los caracteres introducidos (valor por defecto).
	2 <b>fmMatchEntryNone</b>	Sin coincidencia.
MatchFound	Indica si el texto introducido coincide con alguna de las entradas del control.	Boolean
MatchRequired	Indica si es obligatorio que el texto introducido en el control coincida con alguna de las entradas del mismo. Si True, el usuario no puede salir del control hasta que el texto introducido coincida con alguna de las entradas.	Boolean
Max	Valor máximo para la propiedad Value.	Long
MaxLength	Indica el número máximo de caracteres permitido.	Long <sup>®</sup>
Min	Valor mínimo para la propiedad Value.	Long
MouseIcon	Icono asociado al control cuando el mouse pasa por encima del mismo. La	

	propiedad <code>MousePointer</code> debe establecerse a 99.		
<code>MousePointer</code>	Tipo de puntero mostrado. Debe tener alguno de los valores de la constante <code>fmMousePointer</code> :		
	0	<i><code>fmMousePointer-Default</code></i>	Puntero estándar. Es el valor por defecto.
	1	<i><code>fmMousePointer-Arrow</code></i>	Tipo flecha.
	2	<i><code>fmMousePointer-Cross</code></i>	Tipo cruz.
	3	<i><code>fmMousePointer-IBeam</code></i>	Tipo I.
	6	<i><code>fmMousePointerSize-NESW</code></i>	Flecha doble señalando al noreste y al suroeste.
	7	<i><code>fmMousePointerSize-NS</code></i>	Flecha doble señalando al norte y al sur.
	8	<i><code>fmMousePointerSize-NWSE</code></i>	Flecha doble señalando al noroeste y al sureste.
	9	<i><code>fmMousePointerSize-WE</code></i>	Flecha doble señalando al oeste y al este.
	10	<i><code>fmMousePointerUp-Arrow</code></i>	Flecha arriba.
	11	<i><code>fmMousePointer-Hourglass</code></i>	Reloj de arena.
	12	<i><code>fmMousePointer-NoDrop</code></i>	Círculo con una línea diagonal indicando 'destino prohibido'.

	13	<b><i>fmMousePointer-AppStarting</i></b>	Flecha con reloj de arena.
	14	<b><i>fmMousePointer-Help</i></b>	Flecha con interrogación.
	15	<b><i>fmMousePointer-SizeAll</i></b>	Flecha señalando al norte, al sur, al este y al oeste.
	99	<b><i>fmMousePointer-Custom</i></b>	Usa el icono indicado en la propiedad <b>MouseIcon</b> .
MultiLine	Define o consulta si se pueden introducir múltiples líneas de texto.		Boolean
MultiSelect	Indica si se pueden realizar selecciones múltiples. Puede ser alguno de los valores de la constante fmMultiSelect:		fmMultiSelect
	0	<b>fmMultiSelectSingle</b>	Sólo puede seleccionarse un elemento (valor por defecto).
	1	<b>fmMultiSelectMulti</b>	BARRA ESPACIADORA o clic selecciona o anula la selección de un elemento.
	2	<b>fmMultiSelectExtended</b>	Presionar <sup>®</sup> MAYÚSCULAS y hacer clic con el mouse o una de la flechas de dirección para

				seleccionar desde el elemento seleccionado hasta el elemento actual. CTRL y hacer clic con el mouse selecciona o anula la selección de un elemento.	
Name	Nombre del objeto.			String	
Object	Permite sobrescribir un método o propiedad estándar en caso de que un nuevo control tenga un método o propiedad con el mismo nombre.			Object	
OLEType	Proporciona el tipo de objeto OLE. Puede ser alguno de los valores de la constante XIOLEType: xIOLELink o xIOLEEmbed			Long	
Orientation	Indica si la orientación del control es horizontal o vertical. Puede ser alguno de los valores de la constante <b>fmOrientation</b> :			fmOrientation	
	- 1	<b>fmOrientation-Auto</b>	Determina automáticamente la orientación basándose en las dimensiones del control (predeterminado).		
	0	<b>fmOrientation-Vertical</b>	El control se representa verticalmente.		
	1	<b>fmOrientation-Horizontal</b>	El control se representa horizontalmente.		



Parent	Proporciona el nombre del objeto o colección que contiene al objeto en cuestión.	String	
PasswordChar	Permite introducir un determinado carácter en lugar del carácter tecleado. Útil para ocultar la entrada de texto.	String	
Picture	Indica el mapa de bits mostrado en el objeto. Para cargar dicho mapa de bits debe utilizar <b>LoadPicture</b> ( ruta ) siendo <b>ruta</b> la ruta donde se halla el archivo de imagen.		
PictureAlignment	Indica la ubicación de una imagen de fondo. Puede ser alguno de los valores de la constante fmPictureAlignment:	fmPictureAlignment	
	0	<b>fmPictureAlignmentTopLeft</b>	Esquina superior izquierda.
	1	<b>fmPictureAlignmentTop-Right</b>	Esquina superior derecha.
	2	<b>fmPictureAlignment-Center</b>	Centro.
	3	<b>fmPictureAlignmentBottom-Left</b>	Esquina inferior izquierda.
	4	<b>fmPictureAlignmentBottom-Right</b>	Esquina inferior derecha.
PicturePosition	Indica donde ha de colocarse la imagen en relación al título del objeto. Puede ser alguno de los valores de la constante <b>fmPicturePosition</b> :		

	0	<b>fmPicturePosition-LeftTop</b>	Imagen: izquierda del título. Título: parte superior de la imagen.
	1	<b>fmPicturePosition-LeftCenter</b>	Imagen: izquierda del título. Título: centrado en relación a la imagen.
	2	<b>fmPicturePosition-LeftBottom</b>	Imagen: izquierda del título. Título: parte inferior de la imagen.
	3	<b>fmPicturePosition-RightTop</b>	Imagen: derecha del título. Título: parte superior de la imagen.
	4	<b>fmPicturePosition-RightCenter</b>	Imagen: derecha del título. Título: centrado con la imagen.
	5	<b>fmPicturePosition-RightBottom</b>	Imagen: derecha del título. Título: parte inferior de la imagen.
	6	<b>fmPicturePosition-AboveLeft</b>	Imagen: sobre el título. Título: alineado con el borde izquierdo de la imagen.
	7	<b>fmPicturePosition-AboveCenter</b>	Imagen: sobre el título. Título: centrado debajo de la imagen (predeterminado).
	8	<b>fmPicturePosition-AboveRight</b>	Imagen: sobre el título. Título: alineado con el borde derecho de la imagen.
	9	<b>fmPicturePosition-</b>	Imagen: debajo del título.

		<b>BelowLeft</b>	Título: alineado con el borde izquierdo de la imagen.
	10	<b>fmPicturePosition-BelowCenter</b>	Imagen: debajo del título.  Título: centrado debajo de la imagen.
	11	<b>fmPicturePosition-BelowRight</b>	Imagen: debajo del título.  Título: alineado con el borde derecho de la imagen.
	12	<b>fmPicturePosition-Center</b>	Imagen: en el centro del control.  Título: centrado horizontal y verticalmente en la parte superior de la imagen.
<b>PictureSizeMode</b>		Indica cómo se ha de mostrar la imagen de fondo en un control. Puede tener alguno de los valores de la constante <b>fmPictureSizeMode</b> :	
	0	<b>fmPictureSizeMode-Clip</b>	Recorta la imagen si es más grande que el formulario o página (valor por defecto).
	1	<b>fmPictureSizeMode-Stretch</b>	Estira la imagen rellenando el formulario o página. Deforma la imagen.
	3	<b>fmPictureSizeMode-Zoom</b>	Amplía la imagen, sin deformarla.

PictureTiling	Indica si la imagen se ha de organizar en forma de mosaico cuando ésta posea un tamaño menor que el formulario o página que la contiene.	Boolean	
Placement	Indica cómo se une el objeto a las celdas que están debajo de él. Puede ser alguno de los valores de la constante <b>XIPlacement</b> :	XIPlacement	
	1	<b>xlMoveAndSize</b>	El objeto se mueve y cambia de tamaño al mismo tiempo que las celdas.
	2	<b>xlMove</b>	El objeto se ha de mover junto a las celdas.
	3	<b>xlFreeFloating</b>	El objeto se muestra flotando.
PrintObject	Indica si el objeto se ha de imprimir cuando se imprima el documento.	Boolean	
ProgID	Identificador programático de OLE del objeto. Puede ser alguno de los siguientes:	String	
	Control	Identificador	
	CheckBox	Forms.CheckBox.1	
	ComboBox	Forms.ComboBox.1	

	Command-Button	Forms.CommandButton.1	
	Frame	Forms.Frame.1	
	Image	Forms.Image.1	
	Label	Forms.Label.1	
	ListBox	Forms.ListBox.1	
	MultiPage	Forms.MultiPage.1	
	Option-Button	Forms.OptionButton.1	
	ScrollBar	Forms.ScrollBar.1	
	SpinButton	Forms.SpinButton.1	
	TabStrip	Forms.TabStrip.1	
	TextBox	Forms.TextBox.1	
	Toggle-Button	Forms.ToggleButton.1	
ProportionalThumb	Indica si el tamaño del cuadro de desplazamiento ha de ser proporcional al área de desplazamiento o fijo. Si False, el cuadro de desplazamiento será de tamaño fijo.		Boolean
ScrollBars	Indica si se ha de mostrar la barra de desplazamiento vertical, horizontal o ambas. Puede ser alguno de los valores de la constante fmScrollBars:		fmScrollBars

	0	<b>fmScrollBarsNone</b>	No se muestra ninguna barra de desplazamiento (valor por defecto).
	1	<b>fmScrollBars-Horizontal</b>	Muestra barra de desplazamiento horizontal.
	2	<b>fmScrollBars-Vertical</b>	Muestra barra de desplazamiento vertical.
	3	<b>fmScrollBarsBoth</b>	Muestra ambas barras de desplazamiento.
Selected	Define o consulta el estado de selección de los elementos del control.		Boolean
SelectionMargin	Indica si la selección puede realizarse haciendo clic en el margen que hay a la izquierda del texto.		Boolean
SelLength	Indica el número de caracteres seleccionados en el texto.		Long
SelStart	Indica el inicio del texto seleccionado.		Long
SelText	Indica el texto seleccionado.		String
Shadow	Define o consulta si el objeto tiene sombra.		Boolean
ShapeRange	Proporciona un objeto de tipo ShapeRange el cual representa un conjunto de formas.		ShapeRange
ShowDropButtonWhen	Indica cuando se ha de mostrar el botón de desplegar del control. Puede ser alguno de los valores de la constante fmShowDropButtonWhen:		fmShowDrop-ButtonWhen

	0	<b>fmShowDropButtonWhen-Never</b>	Nunca.
	1	<b>FmShowDropButtonWhen-Focus</b>	Muestra el botón cuando tiene el foco.
	2	<b>FmShowDropButtonWhen-Always</b>	Siempre lo muestra (valor por defecto).
SmallChange	Indica la cantidad de desplazamiento que se ha de producir al hacer clic en una flecha de desplazamiento.		Long
SourceName	Define o consulta el nombre de origen del vínculo del objeto.		String
SpecialEffect	Indica la apariencia visual de un objeto. Puede ser alguno de los valores de la constante fmSpecialEffect:		fmSpecialEffect
	0	<b>fmSpecialEffect-Flat</b>	El objeto se muestra plano, distinguido por un borde, un cambio de color o ambos. (Valor por defecto).
	1	<b>FmSpecialEffect-Raised</b>	Resaltado en la parte superior - izquierda y con una sombra en la parte inferior - derecha. No válido para CheckBox ni OptionButton.

	2	<b>FmSpecialEffect-Sunken</b>	El objeto tiene una sombra en la parte superior - izquierda y está resaltado en la parte inferior - derecha. Valor por defecto para los controles CheckBox y OptionButton.
	3	<b>FmSpecialEffect-Etched</b>	El borde parece estar cincelado alrededor del control.  No válido para CheckBox ni OptionButton.
	6	<b>FmSpecialEffect-Bump</b>	Relieve en la parte inferior - derecha, y plano en la parte superior - izquierda. No válido para CheckBox ni OptionButton.
Style	Indica cómo puede seleccionar o definir el valor del control. Puede ser alguno de los valores de la constante fmStyle:		fmStyle



	0	<b>fmStyleDropDown-Combo</b>	Se puede escribir un valor en el área de edición o seleccionar un valor de la lista (valor por defecto).
	2	<b>FmStyleDropDown-List</b>	El usuario debe seleccionar un valor de la lista.
TakeFocusOnClick	Indica si el control recibe el foco cuando se hace clic sobre el mismo.		Boolean
Text	Define o consulta el texto del cuadro de texto del control.		String
TextAlign	Indica cómo debe alinearse el texto. Puede ser alguno de los valores de a constante fmTextAlign:		fmTextAlign
	1	<b>fmTextAligLeft</b>	Alineación a la izquierda. (valor por defecto).
	2	<b>FmTextAlignCenter</b>	Centra el texto.
	3	<b>FmTextAlignRight</b>	Alineación a la derecha.
TextColumn	Indica cual es la columna que debe almacenarse en la propiedad text cuando el usuario selecciona una fila.		Variant
TextLength	Indica la longitud del texto existente en el área de edición del control.		Long
Top	Distancia entre el borde superior del objeto que lo contiene y el propio control.		Single®
TopIndex	Define o consulta el primer elemento de la lista.		Variant

TopLeftCell	Proporciona un objeto de tipo Range con la celda que se halla debajo de la esquina superior izquierda del objeto.	Range
TriState	Indica si se puede definir el estado nulo para el control.	Boolean
Value	Indica el valor o estado de un control.	Variant
Visible	Indica si el objeto está visible o no.	Boolean
Width	Ancho del objeto en puntos.	Single
WordWrap	Indica si el contenido del control se ajusta al final de la línea automáticamente.	Boolean
ZOrder	Indica la posición del objeto en el orden z (orden en el que se dibujan unas imágenes por encima o por debajo de otras).	Long

### 3.1.13 Resumen de métodos

Nombre	Comentario
Activate	Activa el objeto.
AddItem	Añade un elemento a la lista.
BringToFront	Ubica el objeto encima de todos (primer plano del orden z – Zorder-)
Clear	Elimina todos los elementos de la lista.
Copy	Copia el objeto en el portapapeles.
CopyPicture	Copia el objeto en el portapapeles en formato imagen.
Cut	Corta el objeto pegándolo en el portapapeles.

Delete	Elimina el objeto.						
DropDown	Permite desplegar la lista de elementos.						
Duplicate	Duplica el objeto proporcionando una referencia a la copia.						
Paste	Copia el contenido del portapapeles al objeto.						
RemoveItem	Elimina una fila del control.						
Select	Selecciona el objeto.						
SendToBack	Ubicar el objeto abajo de todos (ultimo plano del orden z –Zorder-)						
Update	Actualiza el vínculo del objeto.						
Verb	Enviar un verbo al servidor del objeto OLE. Puede ser alguno de los valores de la constante <b>XIOLEVerb</b> : <table><tr><td>1</td><td><b>xlVerbPrimary</b></td><td>Realizar la acción principal del servidor.</td></tr><tr><td>2</td><td><b>xlVerbOpen</b></td><td>Abrir el objeto.</td></tr></table>	1	<b>xlVerbPrimary</b>	Realizar la acción principal del servidor.	2	<b>xlVerbOpen</b>	Abrir el objeto.
1	<b>xlVerbPrimary</b>	Realizar la acción principal del servidor.					
2	<b>xlVerbOpen</b>	Abrir el objeto.					

### 3.1.14 Resumen de eventos

Nombre	Comentario
BeforeDragOver	Producido al arrastrar y colocar un objeto.
BeforeDropOrPaste	Producido al colocar o pegar datos en un objeto.
BeforUpdate	Producido antes de que cambien los datos del control.
Change	Producido al cambiar el valor de la propiedad Value.
Click	Producido al hacer clic con el botón izquierdo del mouse sobre un objeto o al seleccionar un valor en caso de que haya más de uno.

DbClick	Producido al hacer clic 2 veces seguidas con el mouse sobre un objeto.
DropButtonClick	Producido al desplegar u ocultar la lista de elementos del control.
Error	Producido al detectar un error.
GotFocus	Producido al recibir el foco de entrada. Dicho foco puede obtenerse mediante una acción de usuario o mediante el método SetFocus.
KeyDown	Producido cuando el usuario pulsa una tecla.
KeyPress	Producido cuando el usuario ha pulsado una tecla.
KeyUp	Producido cuando el usuario libera una tecla pulsada.
LostFocus	Producido al perder el foco de entrada
MouseDown	Producido al pulsar un botón del mouse.
MouseMove	Producido al mover el mouse.
MouseUp	Producido al soltar un botón del mouse.
SpinDown	Se produce al hacer clic sobre la flecha inferior o izquierda.
SpinUp	Se produce al hacer clic sobre la flecha superior o derecha.
Zoom	Producido al cambiar el valor de la propiedad Zoom (en controles Frame y MultiPage).

Ra-Ma®

## FORMULARIOS

### 4.1.1 Label

#### 4.1.1.1 PROPIEDADES

Accelerator	ControlSource	Left	Parent	TextAlign
Autosize	ControlTipText	MouseIcon	Picture	Top
BackColor	Default	MousePointer	PicturePosition	Visible
BackStyle	Enabled	Name	RowSource	Width
BorderColor	Font	Object	RowSourceType	WordWrap
BorderStyle	ForeColor	OldHeight	SpecialEffect	
Cancel	Height	OldLeft	TabIndex	
BottomRightCell	HelpContextID	OldTop	TabStop	
Caption	LayoutEffect	OldWidth	Tag	

### 4.1.1.2 MÉTODOS

Move
SetFocus
ZOrder

### 4.1.1.3 EVENTOS

BeforeDragOver	Error
BeforeDropOrPaste	MouseDown
Click	MouseMove
DbClick	MouseUp

## 4.1.2 TextBox

### 4.1.2.1 PROPIEDADES

AutoSize	CurX	LayoutEffect	OldWidth	TabStop
AutoTab	Default	Left	Parent	Tag
AutoWordSelect	DragBehavior	LineCount	PasswordChar	Text
BackColor	Enabled	Locked	RowSource	TextAlign
BackStyle	EnterFieldBehavior	MaxLength	RowSourceType	TextLength
BorderColor	EnterKeyBehavior	MouseIcon	ScrollBars	Top
BorderStyle	Font	MousePointer	SelectionMargin	Value
Cancel	ForeColor	MultiLine	SelLength	Visible
CanPaste	Height	Name	SelStart	Width

ControlSource	HelpContextID	Object	SelText	WordWrap
ControlTipText	HideSelection	OldHeight	SpecialEffect	
CurLine	IMEMode	OldLeft	TabIndex	
CurTargetX	IntegralHeight	OldTop	TabKeyBehavior	

#### 4.1.2.2 MÉTODOS

Copy
Cut
Move
Paste
SetFocus
ZOrder

#### 4.1.2.3 EVENTOS

AfterUpdate	Change	Error	KeyUp
BeforeDragOver	DblClick	Exit	MouseDown
BeforeDropOrPaste	DropButtonClick	KeyDown	MouseMove
BeforUpdate	Enter	KeyPress	MouseUp

Ra-Ma®

## 4.1.3 ComboBox

### 4.1.3.1 PROPIEDADES

AutoSize	ControlTipText	Left	MousePointer	SourceName
AutoTab	CurTargetX	LineCount	Name	SpecialEffect
AutoWordSelect	CurX	List	Object	Style
BackColor	Default	ListCount	OldHeight	TabIndex
BackStyle	DragBehavior	ListFillRange	OldLeft	TabStop
BorderColor	DropButtonStyle	ListIndex	OldTop	Tag
BorderStyle	Enabled	ListRows	OldWidth	Text
BoundColumn	EnterFieldBehavior	ListStyle	Parent	TextAlign
Cancel	Font	ListWidth	RowSource	TextColumn
CanPaste	ForeColor	Locked	RowSourceType	TextLength
Column	Height	MatchEntry	SelectionMargin	Top
ColumnCount	HelpContextID	MatchFound	SelLength	TopIndex
ColumnHeads	HideSelection	MatchRequired	SelStart	Value
ColumnWidths	IMEMode	MaxLength	SelText	Visible
ControlSource	LayoutEffect	MouseIcon	ShowDropButtonWhen	Width

### 4.1.3.2 MÉTODOS

AddItem	DropDown	SetFocus
Clear	Move	ZOrder
Copy	Paste	
Cut	RemoveItem	



### 4.1.3.3 EVENTOS

AfterUpdate	Click	Exit	MouseMove
BeforeDragOver	DblClick	KeyDown	MouseUp
BeforeDropOrPaste	DropButtonClick	KeyPress	
BeforUpdate	Enter	KeyUp	
Change	Error	MouseDown	

## 4.1.4 ListBox

### 4.1.4.1 PROPIEDADES

BackColor	ControlTipText	ListCount	OldLeft	Text
BorderColor	Default	ListIndex	OldTop	TextAlign
BorderStyle	Enabled	ListStyle	OldWidth	TextColumn
BottomRightCell	Font	Locked	Parent	Top
BoundColumn	ForeColor	MatchEntry	RowSource	TopIndex
Cancel	Height	MouseIcon	RowSourceType	Value
Column	HelpContextID	MousePointer	Selected	Visible
ColumnCount	IntegralHeight	MultiSelect	SpecialEffect	Width
ColumnHeads	LayoutEffect	Name	TabIndex	
ColumnWidths	Left	Object	TabStop	
ControlSource	List	OldHeight	Tag	

### 4.1.4.2 MÉTODOS

AddItem	RemoveItem
Clear	SetFocus
Move	ZOrder

### 4.1.4.3 EVENTOS

AfterUpdate	Click	Error	KeyUp
BeforeDragOver	Change	Exit	MouseDown
BeforeDropOrPaste	DbtClick	KeyDown	MouseMove
BeforUpdate	Enter	KeyPress	MouseUp

## 4.1.5 CheckBox

### 4.1.5.1 PROPIEDADES

Accelerator	Default	Locked	Parent	Text
Alignment	Enabled	MouseIcon	Picture	TextAlign
Autosize	Font	MousePointer	PicturePosition	TextColumn
BackColor	ForeColor	Name	RowSource	Top
BackStyle	GroupName	Object	RowSourceType	TripleState
Cancel	Height	OldHeight	SpecialEffect	Value
Caption	HelpContextID	OldLeft	TabIndex	Visible
ControlSource	LayoutEffect	OldTop	TabStop	Width
ControlTipText	Left	OldWidth	Tag	WordWrap

### 4.1.5.2 MÉTODOS

Move
SetFocus
ZOrder

### 4.1.5.3 EVENTOS

AfterUpdate	Change	Error	KeyUp
BeforeDragOver	Click	Exit	MouseDown
BeforeDropOrPaste	DblClick	KeyDown	MouseMove
BeforUpdate	Enter	KeyPress	MouseUp

## 4.1.6 OptionButton

### 4.1.6.1 PROPIEDADES

Accelerator	Default	Locked	Parent	TextAlign
Alignment	Enabled	Mouselcon	Picture	Top
Autosize	Font	MousePointer	PicturePosition	TripleState
BackColor	ForeColor	Name	RowSource	Value
BackStyle	GroupName	Object	RowSourceType	Visible
Cancel	Height	OldHeight	SpecialEffect	Width
Caption	HelpContextID	OldLeft	TabIndex	WordWrap
ControlSource	LayoutEffect	OldTop	TabStop	
ControlTipText	Left	OldWidth	Tag	

### 4.1.6.2 MÉTODOS

Move
SetFocus
ZOrder

### 4.1.6.3 EVENTOS

AfterUpdate	Change	Error	KeyUp
BeforeDragOver	Click	Exit	MouseDown
BeforeDropOrPaste	DblClick	KeyDown	MouseMove
BeforUpdate	Enter	KeyPress	MouseUp

## 4.1.7 ToggleButton

### 4.1.7.1 PROPIEDADES

Accelerator	Default	Locked	Parent	TextAlign
Alignment	Enabled	MouseIcon	Picture	Top
Autosize	Font	MousePointer	PicturePosition	TripleState
BackColor	ForeColor	Name	RowSource	Value
BackStyle	GroupName	Object	RowSourceType	Visible
Cancel	Height	OldHeight	SpecialEffect	Width
Caption	HelpContextID	OldLeft	TabIndex	WordWrap
ControlSource	LayoutEffect	OldTop	TabStop	
ControlTipText	Left	OldWidth	Tag	

### 4.1.7.2 MÉTODOS

Move
SetFocus
ZOrder

### 4.1.7.3 EVENTOS

AfterUpdate	Change	Error	KeyUp
BeforeDragOver	Click	Exit	MouseDown
BeforeDropOrPaste	DblClick	KeyDown	MouseMove
BeforUpdate	Enter	KeyPress	MouseUp

## 4.1.8 Frame

### 4.1.8.1 PROPIEDADES

ActiveControl	ControlTipText	LayoutEffect	Picture	SpecialEffect
BackColor	Cycle	Left	PictureAlignment	TabIndex
BorderColor	Default	MouseIcon	PictureSizeMode	TabStop
BorderStyle	Enabled	MousePointer	PictureTiling	Tag
Cancel	Font	Name	RowSource	Top
CanPaste	ForeColor	Object	RowSourceType	TripleState
CanRedo	Height	OldHeight	ScrollBars	Value
CanUndo	HelpContextID	OldLeft	ScrollHeight	VerticalScrollBar-Side
Caption	InsideHeight	OldTop	ScrollLeft	Visible
Controls	InsideWidth	OldWidth	ScrollTop	Width
ControlSource	KeepScrollBarsVisible	Parent	ScrollWidth	Zoom

### 4.1.8.2 MÉTODOS

Copy	RedoAction	SetFocus
Cut	Repaint	UndoAction
Move	Scroll	ZOrder
Paste	SetDefaultTabOrder	

### 4.1.8.3 EVENTOS

AddControl	Change	KeyDown	MouseUp
AfterUpdate	DblClick	KeyPress	RemoveControl
BeforeDragOver	Enter	KeyUp	Scroll
BeforeDropOrPaste	Error	MouseDown	Zoom
Click	Exit	MouseMove	

## 4.1.9 CommandButton

### 4.1.9.1 PROPIEDADES

Accelerator	Default	Locked	OldWidth	TabStop
Autosize	Enabled	MouseIcon	Parent	Tag
BackColor	Font	MousePointer	Picture	TakeFocusOnClick
BackStyle	ForeColor	Name	PicturePosition	Top
Cancel	Height	Object	RowSource	Visible
Caption	HelpContextID	OldHeight	RowSourceType	Width
ControlSource	LayoutEffect	OldLeft	SpecialEffect	WordWrap
ControlTipText	Left	OldTop	TabIndex	

### 4.1.9.2 MÉTODOS

Move
SetFocus
ZOrder

### 4.1.9.3 EVENTOS

BeforeDragOver	Enter	KeyPress	MouseUp
BeforeDropOrPaste	Error	KeyUp	
Click	Exit	MouseDown	
DblClick	KeyDown	MouseMove	

## 4.1.10 TabStrip

### 4.1.10.1 PROPIEDADES

BackColor	Enabled	MultiRow	RowSourceType	Tag
Cancel	Font	Name	SelectedItem	Top
ClientHeight	ForeColor	Object	Style	Value
ClientLeft	Height	OldHeight	TabFixedHeight	Visible
ClientTop	HelpContextID	OldLeft	TabFixedWidth	Width
ClientWidth	LayoutEffect	OldTop	TabIndex	
ControlSource	Left	OldWidth	TabOrientation	
ControlTipText	MouseIcon	Parent	Tabs	
Default	MousePointer	RowSource	TabStop	

### 4.1.10.2 MÉTODOS

Move
SetFocus
ZOrder

### 4.1.10.3 EVENTOS

BeforeDragOver	DbClick	KeyDown	MouseMove
BeforeDropOrPaste	Enter	KeyPress	MouseUp
Click	Error	KeyUp	
Change	Exit	MouseDown	

## 4.1.11 MultiPage

### 4.1.11.1 PROPIEDADES

BackColor	ForeColor	Object	RowSource	TabOrientation
Cancel	Height	OldHeight	RowSourceType	TabStop
ControlSource	HelpContextID	OldLeft	SelectedItem	Tag
ControlTipText	LayoutEffect	OldTop	Style	Top
Default	Left	OldWidth	TabFixedHeight	Value
Enabled	MultiRow	Pages	TabFixedWidth	Visible
Font	Name	Parent	TabIndex	Width



### 4.1.11.2 MÉTODOS

Move
SetFocus
ZOrder

### 4.1.11.3 EVENTOS

AddControl	DblClick	KeyPress	MouseUp
BeforeDragOver	Enter	KeyUp	RemoveControl
BeforeDropOrPaste	Error	Layout	Scroll
Click	Exit	MouseDown	Zoom
Change	KeyDown	MouseMove	

## 4.1.12 ScrollBar

### 4.1.12.1 PROPIEDADES

BackColor	Height	MouseIcon	Orientation	Tag
Cancel	HelpContextID	MousePointer	Parent	Top
ControlSource	IntegralHeight	Name	ProportionalThumb	Value
ControlTipText	LargeChange	Object	RowSource	Visible
Default	LayoutEffect	OldHeight	RowSourceType	Width
Delay	Left	OldLeft	SmallChange	
Enabled	Max	OldTop	TabIndex	
ForeColor	Min	OldWidth	TabStop	

### 4.1.12.2 MÉTODOS

Move
SetFocus
ZOrder

### 4.1.12.3 EVENTOS

AfterUpdate	Change	Exit	KeyUp
BeforeDragOver	Enter	KeyDown	Scroll
BeforeDropOrPaste	Error	KeyPress	

## 4.1.13 SpinButton

### 4.1.13.1 PROPIEDADES

BackColor	ForeColor	MouseIcon	OldWidth	TabIndex
Cancel	Height	MousePointer	Orientation	TabStop
ControlSource	HelpContextID	Name	Parent	Tag
ControlTipText	LayoutEffect	Object	ProportionalThumb	Top
Default	Left	OldHeight	RowSource	Value
Delay	Max	OldLeft	RowSourceType	Visible
Enabled	Min	OldTop	SmallChange	Width

### 4.1.13.2 MÉTODOS

Move
SetFocus
ZOrder

### 4.1.13.3 EVENTOS

AfterUpdate	Change	KeyDown	SpinUp
BeforeDragOver	Enter	KeyPress	
BeforeDropOrPaste	Error	KeyUp	
BeforUpdate	Exit	SpinDown	

## 4.1.14 Image

### 4.1.14.1 PROPIEDADES

AutoSize	Default	MouseIcon	Parent	TabIndex
BackColor	Enabled	MousePointer	Picture	TabStop
BackStyle	Font	Name	PictureAlignment	Tag
BorderColor	ForeColor	Object	PictureSizeMode	Top
BorderStyle	Height	OldHeight	PictureTiling	Visible
Cancel	HelpContextID	OldLeft	RowSource	Width
ControlSource	LayoutEffect	OldTop	RowSourceType	
ControlTipText	Left	OldWidth	SpecialEffect	

### 4.1.14.2 MÉTODOS

Move
SetFocus
ZOrder

### 4.1.14.3 EVENTOS

BeforeDragOver	DblClick	MouseMove
BeforeDropOrPaste	Error	MouseUp
Click	MouseDown	

## 4.2 RESUMEN DE PROPIEDADES

Propiedad	Comentario	Valores
Accelerator	Define o consulta la tecla de aceleración de un control.	String
ActiveControl	Permite manipular e identificar el control que tiene el foco.	
Alignment	Define o consulta la posición del control en relación a su título. Puede ser alguno de los valores de la constante fmAlignment:	fmAlignment
	0 <b>fmAlignment-Left</b>	Coloca el título a la izquierda del control.
	1 <b>FmAlignment-Right</b>	Coloca el título a la derecha del control (predeterminado).

Autosize	Indica si el objeto ha de cambiar su tamaño automáticamente para mostrar su contenido.	Boolean						
AutoTab	Indica si se ha de producir una tabulación automática cuando el usuario sobrepasa el máximo de caracteres permitido en la parte del cuadro de texto.	Boolean						
AutoWordSelect	Indica si la selección ha de extenderse a una palabra o a un carácter.	Boolean						
BackColor	Color de fondo. La propiedad BackStyle debe establecerse a fmBackStyleOpaque. Puede definirse usando la función RGB (rojo, verde y azul).	Long						
BackStyle	Define el estilo del fondo. Puede ser alguno de los valores de la constante <b>fmBackStyle</b>	fmBackStyle						
	<table><tr><td>0</td><td><b>fmBackStyle-Transparent</b></td><td>El fondo transparente</td></tr><tr><td>1</td><td><b>fmBackStyleOpaque</b></td><td>El fondo opaco. Valor por defecto</td></tr></table>		0	<b>fmBackStyle-Transparent</b>	El fondo transparente	1	<b>fmBackStyleOpaque</b>	El fondo opaco. Valor por defecto
	0		<b>fmBackStyle-Transparent</b>	El fondo transparente				
1	<b>fmBackStyleOpaque</b>	El fondo opaco. Valor por defecto						
BorderColor	Indica el color del borde de un objeto.	Long						
BorderStyle	Indica el tipo de borde de un objeto. Puede ser alguno de los valores de la constante <b>fmBorderStyle</b> :	fmBorderStyle						

	0	<b>fmBorderStyle-None</b>	El control no tiene una línea de borde visible.
	1	<b>fmBorderStyle-Single</b>	El control tiene un borde de una sola línea (predeterminado).
BoundColumn	Indica el elemento o columna que se utilizará como valor del control cuando un usuario seleccione una fila. Por ejemplo, si una fila contiene 3 columnas y se indica 2, el valor del objeto será el correspondiente a la columna 2 de la fila seleccionada.		Variant
Cancel	Permite cancelar cambios no confirmados.		Boolean
CanPaste	Indica si se puede efectuar un pegado desde el portapapeles.		Boolean
CanRedo	Indica si es posible invertir la última ejecución del comando Deshacer.		Boolean
CanUndo	Indica si es posible Deshacer la última acción realizada por el usuario.		Boolean
Caption	Descripción de la etiqueta.		String
ClientHeight	Alto del área de presentación del control TabStrip.		Single
ClientLeft	Ubicación de la posición izquierda del área de presentación del control TabStrip.		Single
ClientTop	Ubicación de la posición superior del área de presentación del control TabStrip.		Single ®
ClientWidth	Ancho del área de presentación del control TabStrip.		Single

Controls	Representa un objeto de tipo Controls con los controles que se hallan contenidos dentro del objeto indicado.	Controls
Column	Indica uno o varios elementos de un control ListBox o ComboBox.	
ColumnCount	Indica el número de columnas a mostrar. Si se indica -1, se muestran todas las columnas disponibles.	Long
ColumnHeads	Muestra una fila de títulos de columna.	Boolean
ColumnWidths	Indica el ancho de cada una de las columnas del control. Los diferentes anchos se indican usando el punto y coma (;) como separador. Por ejemplo 50; 70; 60.	String
ControlSource	Especifica la ubicación de los datos que se utiliza para la propiedad Value. Por ejemplo, si introducimos “Hoja1!A1” en esta propiedad, el control mostrará el contenido de la celda A1 y si el valor del control se modifica, dicha celda también lo hará.	String
ControlTipText	Permite indicar el texto que se mostrará al pasar el puntero del mouse por encima del control.	String
CurLine	Indica la línea actual del control.	Long
CurTargetX	Indica la posición horizontal preferida del punto de inserción en un control medida en décimas de milímetro.	Long
CurX	Indica la posición horizontal actual del punto de inserción en un control medida en décimas de milímetro.	Long

Cycle	Indica el recorrido cíclico a seguir cuando se abandone el último control de un <b>Frame</b> o de un control <b>MultiPage</b> . Puede tener alguno de los valores de la constante <b>fmCycle</b> :		fmCycle
	0	<b>fmCycleAllForms</b>	Recorrido a través de todos los controles del formulario, Frame y MultiPage que se muestran en el formulario.
	2	<b>FmCycleCurrentForm</b>	Recorrido y foco dentro del control en curso (formulario, Frame o MultiPage).
Default	Determina cual es control por defecto en un formulario. Usado especialmente en los controles de tipo botón.		Boolean
Delay	Indica el tiempo de retardo que se produce entre los eventos SpinUp, SpinDown y Change. Al hacer clic y presionar un botón, el primer evento se realiza inmediatamente pero, la segunda ocurrencia del evento será de 5 veces el valor de Delay. Los siguientes eventos se iniciarán después según lo indicado en Delay. Por defecto Delay vale 50 milisegundos.		Long
DragBehavior	Indica si la característica de arrastrar y colocar está habilitada. Puede ser alguno de los valores de la constante fmDragBehavior:		fmDragBehavior
	0	<b>fmDragBehavior-Disabled</b>	No permite arrastrar y colocar (predeterminado).
	1	<b>FmDragBehavior-Enabled</b>	Permite arrastrar y colocar.



DropButtonStyle	Indica el símbolo a mostrar en el botón que despliega la lista de valores. Puede ser alguno de los valores de la constante fmDropButtonStyle:		fmDropButtonStyle
	0	<b>fmDropButtonStylePlain</b>	Botón sin símbolo.
	1	<b>fmDropButtonStyle-Arrow</b>	Flecha hacia abajo (Valor por defecto).
	2	<b>fmDropButtonStyle-Ellipsis</b>	Puntos suspensivos (...).
	3	<b>fmDropButtonStyle-Reduce</b>	Muestra una línea horizontal.
Enabled	Indica si el objeto puede recibir el foco y si puede responder a eventos generados por un usuario.		Boolean
EnterFieldBehavior	Indica cómo se selecciona el texto y puede ser alguno de los valores de la constante fmEnterFieldBehavior:		fmEnterFieldBehavior
	0	<b>fmEnterFieldBehavior-SelectAll</b>	Selecciona todo el contenido al entrar en el control (valor por defecto).
	1	<b>fmEnterFieldBehaviorRecall-Selection</b>	Mantiene la selección.
EnterKeyBehavior	Define el comportamiento de la pulsación de la tecla ENTER. Si True, al pulsar ENTER se crea una nueva línea. Si False, el foco se mueve al siguiente control situado en el orden de tabulación.		Boolean
Font	Permite definir o consultar el tipo de fuente asociada al objeto.		Font

ForeColor	Color del primer plano del objeto. Puede definirse usando la función RGB (rojo, verde y azul) o un valor de tipo Long que represente un color.	Long
GroupName	Define o consulta un nombre de grupo para una serie de controles.	String
Height	Alto del objeto en puntos.	Single
HelpContextID	Permite asociar un tema determinado de un archivo de ayuda de MS Windows.	Long
HideSelection	Indica si se ha de mantener resaltado el texto seleccionado cuando el control pierde el foco.	Boolean
IMEMode	Indica el modo de tiempo de ejecución del editor de métodos de entrada (IME) y se utiliza sólo para aplicaciones escritas para Extremo Oriente.	fmIMEMode
InsideHeight	Devuelve el alto en puntos de la región del cliente (parte de la ventana donde se muestra la salida) dentro de un formulario.	Single
InsideWidth	Devuelve el ancho en puntos de la región del cliente (parte de la ventana donde se muestra la salida) dentro de un formulario.	Single
IntegralHeight	Indica si el control muestra las líneas completas o las muestra parciales. Si True, el cuadro cambia de tamaño automáticamente para poder mostrar las filas completas.	Boolean

KeepScrollBars-Visible	Indica si se han de mostrar o no las barras de desplazamiento. Puede tener alguno de los valores de la constante fmScrollBars:			fmScrollBars
	0	<b>fmScrollBarsNone</b>	No muestra ninguna barra.	
	1	<b>fmScrollBarsHorizontal</b>	Muestra barra horizontal.	
	2	<b>fmScrollBarsVertical</b>	Muestra barra vertical.	
	3	<b>fmScrollBarsBoth</b>	Muestra ambas barras (valor por defecto).	
LargeChange	Indica la cantidad de desplazamiento que se ha de producir al hacer clic entre el cuadro de desplazamiento y la flecha.			Long
LayoutEffect	Indica si un control se ha movido en tiempo de diseño. Puede tener alguno de los valores de la constante fmLayoutEffect:			fmLayoutEffect
	0	<b>fmLayoutEffectNone</b>	El control no se ha movido.	
	1	<b>fmLayoutEffectInitiate</b>	El control se ha movido.	
Left	Distancia entre el borde izquierdo del objeto que lo contiene y el propio control.			Single
LineCount	Indica el número de líneas de texto del control.			Integer

List	Define o consulta las entradas de la lista del control.	Matriz o conjunto de elementos						
ListCount	Indica el número de entradas de la lista.	Long						
ListFillRange	Define o consulta el rango utilizado para rellenar el cuadro de lista indicado.	String						
ListIndex	Indica el elemento seleccionado actualmente en el control.	Variant						
ListRows	Indica el número máximo de filas que se muestran.	Long						
ListStyle	Indica la apariencia de la lista de elementos en el control. Puede ser alguno de los valores de la constante <b>fmListStyle</b> :	fmListStyle						
	<table> <tr> <td>0</td><td><b>fmListStylePlain</b></td><td>Se muestra con el fondo de los elementos resaltado.</td></tr> <tr> <td>1</td><td><b>fmListStyleOption</b></td><td>Muestra <b>OptionButtons</b> o <b>CheckBox</b> para una lista de selección múltiple (Valor por defecto).</td></tr> </table>	0	<b>fmListStylePlain</b>	Se muestra con el fondo de los elementos resaltado.	1	<b>fmListStyleOption</b>	Muestra <b>OptionButtons</b> o <b>CheckBox</b> para una lista de selección múltiple (Valor por defecto).	
0	<b>fmListStylePlain</b>	Se muestra con el fondo de los elementos resaltado.						
1	<b>fmListStyleOption</b>	Muestra <b>OptionButtons</b> o <b>CheckBox</b> para una lista de selección múltiple (Valor por defecto).						
ListWidth	Indica el ancho de la lista en el control.	Variant						
Locked	Indica si el control puede ser editado	Boolean						
MatchEntry	Define o consulta la forma de cómo se realiza la búsqueda en el control cuando el usuario escribe. Puede ser alguno de los valores de la constante <b>fmMatchEntry</b> :	fmMatchEntry						

	0	<b>fmMatchEntry-FirstLetter</b>	El control busca entrada que empiece con el carácter introducido y así sucesivamente.
	1	<b>fmMatchEntry-Complete</b>	Según se van escribiendo caracteres se busca una entrada que coincida con todos los caracteres introducidos (valor por defecto).
	2	<b>fmMatchEntry-None</b>	Sin coincidencia.
MatchFound	Indica si el texto introducido coincide con alguna de las entradas del control.		Boolean
MatchRequired	Indica si es obligatorio que el texto introducido en el control coincida con alguna de las entradas del mismo. Si True, el usuario no puede salir del control hasta que el texto introducido coincida con alguna de las entradas.		Boolean
Max	Valor máximo para la propiedad Value.		Long
MaxLength	Indica el número máximo de caracteres permitido.		Long
Min	Valor mínimo para la propiedad Value.		Long
MouseIcon	Icono asociado al control cuando el mouse pasa por encima del mismo. La propiedad MousePointer debe establecerse a 99.		
MousePointer	Tipo de puntero mostrado. Debe tener alguno de los valores de la constante <b>fmMousePointer</b> :		

0	<b>fmMousePointer-Default</b>	Puntero estándar. Es el valor por defecto.
1	<b>fmMousePointer-Arrow</b>	Tipo flecha.
2	<b>fmMousePointer-Cross</b>	Tipo cruz.
3	<b>fmMousePointer-IBeam</b>	Tipo I.
6	<b>fmMousePointerSize-NESW</b>	Flecha doble señalando al noreste y al suroeste.
7	<b>fmMousePointerSize-NS</b>	Flecha doble señalando al norte y al sur.
8	<b>fmMousePointerSize-NWSE</b>	Flecha doble señalando al noroeste y al sureste.
9	<b>fmMousePointerSize-WE</b>	Flecha doble señalando al oeste y al este.
10	<b>fmMousePointerUp-Arrow</b>	Flecha arriba.
11	<b>fmMousePointer-Hourglass</b>	Reloj de arena.
12	<b>fmMousePointerNo-Drop</b>	Círculo con una línea diagonal indicando 'destino prohibido'.
13	<b>fmMousePointer-AppStarting</b>	Flecha con reloj de arena.
14	<b>fmMousePointer-Help</b>	Flecha con interrogación.
15	<b>fmMousePointer-SizeAll</b>	Flecha señalando al norte, al sur, al este y al oeste.
99	<b>fmMousePointer-Custom</b>	Usa el icono indicado en la propiedad

MultiLine	Define o consulta si se pueden introducir múltiples líneas de texto.		Boolean
MultiRow	Indica si el control puede tener más de una fila de fichas. Si True, puede haber mas de una fila de fichas. Utilizado en el control TabStrip.		Boolean
MultiSelect	Indica si se pueden realizar selecciones múltiples. Puede ser alguno de los valores de la constante fmMultiSelect:		fmMultiSelect
	0	fmMultiSelectSingle	Sólo puede seleccionarse un elemento (valor por defecto).
	1	fmMultiSelectMulti	BARRA ESPACIADORA o clic selecciona o anula la selección de un elemento.
	2	fmMultiSelectExtended	Presionar MAYÚSCULAS y hacer clic con el mouse o una de la flechas de dirección para seleccionar desde el elemento seleccionado hasta el elemento actual. CTRL y hacer clic con el mouse selecciona o anula la selección de un elemento.
Name	Nombre del objeto.		String
Object	Permite sobrescribir un método o propiedad estándar en caso de que un nuevo control tenga un método o propiedad con el mismo nombre.		Object
OldHeight	Indica el alto que tenía el control antes de que se ajustara su tamaño.		Single

OldLeft	Indica la posición izquierda que tenía el control antes de que se moviera.		Single
OldTop	Indica la posición superior que tenía el control antes de que se moviera.		Single
OldWidth	Indica el ancho que tenía el control antes de que se ajustara su tamaño.		Single
Orientation	Indica si la orientación del control es horizontal o vertical. Puede ser alguno de los valores de la constante <b>fmOrientation</b> :		fmOrientation
	-	<b>fmOrientation-Auto</b>	Determina automáticamente la orientación basándose en las dimensiones del control (predeterminado).
	0	<b>fmOrientation-Vertical</b>	El control se representa verticalmente.
	1	<b>fmOrientation-Horizontal</b>	El control se representa horizontalmente.
Pages	Colección de objetos de tipo Page que contiene un control de tipo MultiPage.		Pages
Parent	Proporciona el nombre del objeto o colección que contiene al objeto en cuestión.		String
PasswordChar	Permite introducir un determinado carácter en lugar del carácter tecleado. Útil para ocultar la entrada de texto.		String



Picture	Indica el mapa de bits mostrado en el objeto. Para cargar dicho mapa de bits debe utilizar LoadPicture( ruta ) siendo <b>ruta</b> la ruta donde se halla el archivo de imagen.																
PictureAlignment	Indica la ubicación de una imagen de fondo. Puede ser alguno de los valores de la constante fmPictureAlignment:	fmPictureAlignment															
<table border="1"> <tr> <td>0</td><td><b>fmPictureAlignmentTop-Left</b></td><td>Esquina superior izquierda.</td></tr> <tr> <td>1</td><td><b>fmPictureAlignmentTop-Right</b></td><td>Esquina superior derecha.</td></tr> <tr> <td>2</td><td><b>fmPictureAlignment-Center</b></td><td>Centro.</td></tr> <tr> <td>3</td><td><b>fmPictureAlignment-BottomLeft</b></td><td>Esquina inferior izquierda.</td></tr> <tr> <td>4</td><td><b>fmPictureAlignment-BottomRight</b></td><td>Esquina inferior derecha.</td></tr> </table>			0	<b>fmPictureAlignmentTop-Left</b>	Esquina superior izquierda.	1	<b>fmPictureAlignmentTop-Right</b>	Esquina superior derecha.	2	<b>fmPictureAlignment-Center</b>	Centro.	3	<b>fmPictureAlignment-BottomLeft</b>	Esquina inferior izquierda.	4	<b>fmPictureAlignment-BottomRight</b>	Esquina inferior derecha.
0	<b>fmPictureAlignmentTop-Left</b>	Esquina superior izquierda.															
1	<b>fmPictureAlignmentTop-Right</b>	Esquina superior derecha.															
2	<b>fmPictureAlignment-Center</b>	Centro.															
3	<b>fmPictureAlignment-BottomLeft</b>	Esquina inferior izquierda.															
4	<b>fmPictureAlignment-BottomRight</b>	Esquina inferior derecha.															
PicturePosition	Indica donde ha de colocarse la imagen en relación al título del objeto (ver CommandButton).																
PictureSizeMode	Indica cómo se ha de mostrar la imagen de fondo en un control. Puede tener alguno de los valores de la constante fmPictureSizeMode:	fmPictureSizeMode															

	0	<b>fmPictureSizeModeClip</b>	Recorta la imagen si es más grande que el formulario o página (valor por defecto).
	1	<b>fmPictureSizeMode-Stretch</b>	Estira la imagen rellenando e formulario o página. Deforma la imagen.
	3	<b>fmPictureSizeMode-Zoom</b>	Amplía la imagen, sin deformarla.
PictureTiling	Indica si la imagen se ha de organizar en forma de mosaico cuando ésta posea un tamaño menor que el formulario o página que la contiene.		Boolean
ProportionalThumb	Indica si el tamaño del cuadro de desplazamiento ha de ser proporcional al área de desplazamiento o fijo. Si False, el cuadro de desplazamiento será de tamaño fijo.		Boolean
RowSource	Indica cual es la fuente de datos que proporciona el conjunto de elementos a proporcionar como lista a un control. Por ejemplo, si introducimos “Hoja1!A1:A4” en esta propiedad para un control de tipo ListBox, observaremos que cada uno de los valores existentes en cada celda pasará a visualizarse como un elemento del control.		String

Ra-Ma®

RowSourceType	Permite indicar deben proporcionarse los datos a un ListBox, ComboBox u objeto OLE. Puede tener alguno de los siguientes valores: <table><tr><td>Table/Query</td></tr><tr><td>Value List</td></tr><tr><td>Field List</td></tr></table>	Table/Query	Value List	Field List	String									
Table/Query														
Value List														
Field List														
ScrollBars	Indica si se ha de mostrar la barra de desplazamiento vertical, horizontal o ambas. Puede ser alguno de los valores de la constante fmScrollBars:	fmScrollBars												
	<table><tr><td>0</td><td><b>fmScrollBarsNone</b></td><td>No se muestra ninguna barra de desplazamiento (valor por defecto).</td></tr><tr><td>1</td><td><b>fmScrollBars-Horizontal</b></td><td>Muestra barra de desplazamiento horizontal.</td></tr><tr><td>2</td><td><b>fmScrollBars-Vertical</b></td><td>Muestra barra de desplazamiento vertical.</td></tr><tr><td>3</td><td><b>fmScrollBarsBoth</b></td><td>Muestra ambas barras de desplazamiento.</td></tr></table>	0	<b>fmScrollBarsNone</b>	No se muestra ninguna barra de desplazamiento (valor por defecto).	1	<b>fmScrollBars-Horizontal</b>	Muestra barra de desplazamiento horizontal.	2	<b>fmScrollBars-Vertical</b>	Muestra barra de desplazamiento vertical.	3	<b>fmScrollBarsBoth</b>	Muestra ambas barras de desplazamiento.	
0	<b>fmScrollBarsNone</b>	No se muestra ninguna barra de desplazamiento (valor por defecto).												
1	<b>fmScrollBars-Horizontal</b>	Muestra barra de desplazamiento horizontal.												
2	<b>fmScrollBars-Vertical</b>	Muestra barra de desplazamiento vertical.												
3	<b>fmScrollBarsBoth</b>	Muestra ambas barras de desplazamiento.												
ScrollHeight	Indica el alto en puntos que puede mostrarse al mover las barras de desplazamiento.	Single												
ScrollLeft	Indica la distancia en puntos desde el borde izquierdo del formulario.	Single												
ScrollTop	Indica la distancia en puntos desde el borde superior del formulario.													
ScrollWidth	Indica el ancho en puntos que puede mostrarse al mover las barras de desplazamiento.	Single												

Selected	Define o consulta el estado de un elemento de un ListBox indicando si está seleccionado o no.	Boolean	
SelectedItem	Indica cual es el elemento seleccionado en la actualidad. Usado en controles de tipo TabStrip y MultiPage.		
SelectionMargin	Indica si la selección puede realizarse haciendo clic en el margen que hay a la izquierda del texto.	Boolean	
SelLength	Indica el número de caracteres seleccionados en el texto.	Long	
SelStart	Indica el inicio del texto seleccionado.	Long	
SelText	Indica el texto seleccionado.	String	
ShowDrop-ButtonWhen	Indica cuando se ha de mostrar el botón de desplegar del control. Puede ser alguno de los valores de la constante fmShowDropButtonWhen:	fmShowDrop-ButtonWhen	
	0	<b>fmShowDropButtonWhen-Never</b>	Nunca.
	1	<b>FmShowDropButtonWhen-Focus</b>	Muestra el botón cuando tiene el foco.
	2	<b>FmShowDropButtonWhen-Always</b>	Siempre lo muestra (valor por defecto).
SmallChange	Indica la cantidad de desplazamiento que se ha de producir al hacer clic en una flecha de desplazamiento.	Long	
SourceName	Define o consulta el nombre de origen del vínculo del objeto.	String ®	
SpecialEffect	Indica la apariencia visual de un objeto. Puede ser alguno de los valores de la constante <b>fmSpecialEffect</b> :	fmSpecialEffect	

	0	<b>fmSpecialEffectFlat</b>	El objeto se muestra plano, distinguido por un borde, un cambio de color o ambos. (Valor por defecto).
	1	<b>FmSpecialEffect-Raised</b>	Resaltado en la parte superior - izquierda y con una sombra en la parte inferior - derecha. No válido para CheckBox ni OptionButton.
	2	<b>FmSpecialEffect-Sunken</b>	El objeto tiene una sombra en la parte superior - izquierda y está resaltado en la parte inferior - derecha. Valor por defecto para los controles CheckBox y OptionButton.
	3	<b>FmSpecialEffect-Etched</b>	El borde parece estar cincelado alrededor del control.  No válido para CheckBox ni OptionButton.
	6	<b>FmSpecialEffect-Bump</b>	Relieve en la parte inferior - derecha, y plano en la parte superior - izquierda. No válido para CheckBox ni OptionButton.
Style	Indica cómo puede seleccionar o definir el valor del control. Puede ser alguno de los valores de la constante <b>fmStyle</b> :		<b>fmStyle</b> ®

	0	<b>fmStyleDropDown-Combo</b>	Se puede escribir un valor en el área de edición o seleccionar un valor de la lista (valor por defecto).
	2	<b>FmStyleDropDownList</b>	El usuario debe seleccionar un valor de la lista.
TabFixedHeight	Define o consulta el alto en puntos de las fichas de los controles <b>TabStrip</b> o <b>MultiPage</b> .		Single
TabFixedWidth	Define o consulta el ancho en puntos de las fichas de los controles <b>TabStrip</b> o <b>MultiPage</b> .		Single
TabIndex	Orden en el que se halla el control dentro del orden de tabulación definido para el paso de un control a otro cada vez que se pulsa la tecla TAB.		Integer
TabOrientation	Define o consulta dónde han de ubicarse las fichas en los controles <b>TabStrip</b> o <b>MultiPage</b> . Puede tener alguno de los valores de la constante <b>fmTabOrientation</b> :		fmTabOrientation

Ra-Ma®

	0	<b>fmTabOrientationTop</b>	Mostradas en la parte superior (valor por defecto).
	1	<b>FmTabOrientation-Bottom</b>	Mostradas en la parte inferior.
	2	<b>FmTabOrientationLeft</b>	Mostradas en la parte izquierda.
	3	<b>FmTabOrientationRight</b>	Mostradas en la parte derecha.
Tabs	Colección de objetos que representa todos los Tabs de un control de tipo TabStrip.		Tabs
TabStop	Define o consulta si el control puede recibir el foco cuando se ha accedido al mismo mediante la pulsación de la tecla TAB.		Boolean
Tag	Permite almacenar información adicional sobre el objeto.		String
TakeFocusOnClick	Indica si el control recibe el foco cuando se hace clic sobre el mismo.		Boolean
Text	Define o consulta el texto del cuadro de texto del control.		String
TextAlign	Indica cómo debe alinearse el texto. Puede ser alguno de los valores de a constante fmTextAlign:		fmTextAlign

Ra-Ma®

	1	<b>fmTextAlignLeft</b>	Alineación a la izquierda. (valor por defecto).
	2	<b>FmTextAlignCenter</b>	Centra el texto.
	3	<b>FmTextAlignRight</b>	Alineación a la derecha.
TextColumn	Indica cual es la columna que debe almacenarse en la propiedad text cuando el usuario selecciona una fila.		Variant
TextLength	Indica la longitud del texto existente en el área de edición del control.		Long
Top	Distancia entre el borde superior del objeto que lo contiene y el propio control.		Single
TopIndex	Define o consulta el primer elemento de la lista.		Variant
TripleState	Indica si se puede definir el estado nulo para el control.		Boolean
Value	Indica el valor o estado de un control.		Variant
VerticalScrollBar Side	Define o consulta en qué lado se desea visualizar la barra de desplazamiento. Puede tener alguno de los valores de la constante fmVerticalScrollbarSide:		fmVerticalScrollbar Side
	0	<b>fmVerticalScrollbar-SideRight</b>	Se coloca a la derecha (valor por defecto).
	1	<b>fmVerticalScrollbar-SideLeft</b>	Se coloca a la izquierda.
Visible	Indica si el objeto está visible o no.		Boolean



Width	Ancho del objeto en puntos.	Single
WordWrap	Indica si el contenido del control se ajusta al final de la línea automáticamente.	Boolean
Zoom	Indica el % del cambio que se ha de aplicar al tamaño del objeto que se está mostrando.	Integer

### 4.3 RESUMEN DE MÉTODOS

Nombre	Comentario
AddItem	Añade un elemento a la lista.
Clear	Elimina todos los elementos de la lista.
Copy	Copia el objeto en el portapapeles.
Cut	Corta el objeto pegándolo en el portapapeles.
DropDown	Permite desplegar la lista de elementos.
Move	Permite mover un control a una posición determinada.
Paste	Copia el contenido del portapapeles al objeto.
RedoAction	Permite invertir el resultado de la última acción Undo ejecutada-
RemoveItem	Elimina una fila del control.
Repaint	Vuelve a dibujar la pantalla o formulario.
Scroll	Permite mover la barra de desplazamiento de un control.
SetDefaultTabOrder	Permite establecer el orden en el que se recorrerán los objetos cada vez que se pulse la tecla TAB. El orden establecido por defecto empieza con el control situado más a

	la izquierda de la parte superior.
SetFocus	Permite colocar el foco en el control indicado.
UndoAction	Invierte la acción realizada por el último comando Deshacer.
ZOrder	Permite colocar el objeto al principio o al final del orden-z (orden en el que se dibujan unas imágenes por encima o por debajo de otras).

## 4.4 RESUMEN DE EVENTOS

Nombre	Comentario										
AddControl	Producido cuando se inserta un control en un formulario en tiempo de ejecución, Frame u objeto Page.										
AfterUpdate	Producido tras cambiar los datos de un control.										
BeforeDragOver	<p>Producido durante la acción de arrastrar y colocar. El estado del arrastre puede tener alguno de los valores de la constante <b>fmDragState</b>:</p> <table> <tr> <td>0</td><td><b>fmDragStatusEnter</b></td><td>El puntero del mouse está dentro del destino.</td></tr> <tr> <td>1</td><td><b>FmDragStatusLeave</b></td><td>El puntero del mouse está fuera del destino.</td></tr> <tr> <td>2</td><td><b>FmDragStatusOver</b></td><td>El puntero del mouse está en una posición nueva, pero dentro del destino.</td></tr> </table>		0	<b>fmDragStatusEnter</b>	El puntero del mouse está dentro del destino.	1	<b>FmDragStatusLeave</b>	El puntero del mouse está fuera del destino.	2	<b>FmDragStatusOver</b>	El puntero del mouse está en una posición nueva, pero dentro del destino.
0	<b>fmDragStatusEnter</b>	El puntero del mouse está dentro del destino.									
1	<b>FmDragStatusLeave</b>	El puntero del mouse está fuera del destino.									
2	<b>FmDragStatusOver</b>	El puntero del mouse está en una posición nueva, pero dentro del destino.									

	<p>El efecto producido por el arrastre puede tener alguno de los siguientes valores:</p> <table><tr><td>0</td><td><b>fmDropEffectNone</b></td><td>No realiza ninguna acción.</td></tr><tr><td>1</td><td><b>FmDropEffectCopy</b></td><td>Copia el origen al destino.</td></tr><tr><td>2</td><td><b>FmDropEffectMove</b></td><td>Mueve el origen al destino.</td></tr><tr><td>3</td><td><b>FmDropEffectCopyOrMove</b></td><td>Copia o mueve el origen al destino.</td></tr></table>	0	<b>fmDropEffectNone</b>	No realiza ninguna acción.	1	<b>FmDropEffectCopy</b>	Copia el origen al destino.	2	<b>FmDropEffectMove</b>	Mueve el origen al destino.	3	<b>FmDropEffectCopyOrMove</b>	Copia o mueve el origen al destino.
0	<b>fmDropEffectNone</b>	No realiza ninguna acción.											
1	<b>FmDropEffectCopy</b>	Copia el origen al destino.											
2	<b>FmDropEffectMove</b>	Mueve el origen al destino.											
3	<b>FmDropEffectCopyOrMove</b>	Copia o mueve el origen al destino.											
BeforeDropOrPaste	<p>Producido antes de pegar o colocar datos en un objeto. Los valores que puede tomar la acción a realizar puede ser alguno de los siguientes:</p> <table><tr><td>2</td><td><b>fmActionPaste</b></td><td>Pega el objeto seleccionado en el destino de colocación.</td></tr><tr><td>3</td><td><b>FmActionDragDrop</b></td><td>Indica que el usuario ha arrastrado el objeto desde su origen al destino de colocación y lo ha colocado en el destino deseado.</td></tr></table> <p>Los valores del efecto, puede consultarlos en la descripción del evento <b>BeforeDragOver</b>.</p>	2	<b>fmActionPaste</b>	Pega el objeto seleccionado en el destino de colocación.	3	<b>FmActionDragDrop</b>	Indica que el usuario ha arrastrado el objeto desde su origen al destino de colocación y lo ha colocado en el destino deseado.						
2	<b>fmActionPaste</b>	Pega el objeto seleccionado en el destino de colocación.											
3	<b>FmActionDragDrop</b>	Indica que el usuario ha arrastrado el objeto desde su origen al destino de colocación y lo ha colocado en el destino deseado.											
BeforUpdate	Producido antes de cambiar los datos de un control.												
Change	Producido al cambiar el valor de la propiedad Value.												
Click	Producido al hacer clic con el botón izquierdo del mouse sobre un objeto o al seleccionar un valor en caso de que haya más de uno.												

DbClick	Producido al hacer clic 2 veces seguidas con el mouse sobre un objeto.
DropButtonClick	Producido al desplegar u ocultar la lista de elementos del control.
Enter	Producido un instante antes de que el control reciba el foco.
Error	Producido al detectar un error.
Exit	Producido justo después de que el control pierda el foco.
KeyDown	Producido cuando el usuario pulsa una tecla.
KeyPress	Producido cuando el usuario ha pulsado una tecla.
KeyUp	Producido cuando el usuario libera una tecla pulsada.
Layout	Producido al cambiar el tamaño de un formulario, Frame o control MultiPage.
MouseDown	Producido al pulsar un botón del mouse.
MouseMove	Producido al mover el mouse.
MouseUp	Producido al soltar un botón del mouse.
RemoveControl	Producido al eliminar un control de un contenedor.
Scroll	Producido al mover un cuadro de desplazamiento en un ScrollBar, MultiPage o Frame.
SpinDown	Producido al hacer clic sobre la flecha inferior o izquierda.
SpinUp	Producido al hacer clic sobre la flecha superior o derecha.
Zoom	Producido al cambiar el valor de la propiedad Zoom (en controles Frame y MultiPage).

## CONTROLES DE FORMULARIO EN VBA

---

### 5.1 CONTROLES.

#### 5.1.1 Label

##### 5.1.1.1 PROPIEDADES

Accelerator	ControlSource	Left	Parent	TextAlign
Autosize	ControlTipText	MouseIcon	Picture	Top
BackColor	Default	MousePointer	PicturePosition	Visible
BackStyle	Enabled	Name	RowSource	Width
BorderColor	Font	Object	RowSourceType	WordWrap
BorderStyle	ForeColor	OldHeight	SpecialEffect	®
Cancel	Height	OldLeft	TabIndex	
BottomRightCell	HelpContextID	OldTop	TabStop	
Caption	LayoutEffect	OldWidth	Tag	

### 5.1.1.2 MÉTODOS

Move
SetFocus
ZOrder

### 5.1.1.3 EVENTOS

BeforeDragOver	Error
BeforeDropOrPaste	MouseDown
Click	MouseMove
DbClick	MouseUp

## 5.1.2 TextBox

### 5.1.2.1 PROPIEDADES

AutoSize	CurX	LayoutEffect	OldWidth	TabStop
AutoTab	Default	Left	Parent	Tag
AutoWordSelect	DragBehavior	LineCount	PasswordChar	Text
BackColor	Enabled	Locked	RowSource	TextAlign
BackStyle	EnterFieldBehavior	MaxLength	RowSourceType	TextLength
BorderColor	EnterKeyBehavior	MouseIcon	ScrollBars	Top
BorderStyle	Font	MousePointer	SelectionMargin	Value
Cancel	ForeColor	MultiLine	SelLength	Visible
CanPaste	Height	Name	SelStart	Width

ControlSource	HelpContextID	Object	SelText	WordWrap
ControlTipText	HideSelection	OldHeight	SpecialEffect	
CurLine	IMEMode	OldLeft	TabIndex	
CurTargetX	IntegralHeight	OldTop	TabKeyBehavior	

### 5.1.2.2 MÉTODOS

Copy
Cut
Move
Paste
SetFocus
ZOrder

### 5.1.2.3 EVENTOS

AfterUpdate	Change	Error	KeyUp
BeforeDragOver	DblClick	Exit	MouseDown
BeforeDropOrPaste	DropButtonClick	KeyDown	MouseMove
BeforUpdate	Enter	KeyPress	MouseUp

Ra-Ma®

## 5.1.3 ComboBox

### 5.1.3.1 PROPIEDADES

AutoSize	ControlTipText	Left	MousePointer	SourceName
AutoTab	CurTargetX	LineCount	Name	SpecialEffect
AutoWordSelect	CurX	List	Object	Style
BackColor	Default	ListCount	OldHeight	TabIndex
BackStyle	DragBehavior	ListFillRange	OldLeft	TabStop
BorderColor	DropButtonStyle	ListIndex	OldTop	Tag
BorderStyle	Enabled	ListRows	OldWidth	Text
BoundColumn	EnterFieldBehavior	ListStyle	Parent	TextAlign
Cancel	Font	ListWidth	RowSource	TextColumn
CanPaste	ForeColor	Locked	RowSourceType	TextLength
Column	Height	MatchEntry	SelectionMargin	Top
ColumnCount	HelpContextID	MatchFound	SelLength	TopIndex
ColumnHeads	HideSelection	MatchRequired	SelStart	Value
ColumnWidths	IMEMode	MaxLength	SelText	Visible
ControlSource	LayoutEffect	MouseIcon	ShowDropButtonWhen	Width

### 5.1.3.2 MÉTODOS

AddItem	DropDown	SetFocus
Clear	Move	ZOrder
Copy	Paste	
Cut	RemoveItem	



### 5.1.3.3 EVENTOS

AfterUpdate	Click	Exit	MouseMove
BeforeDragOver	DblClick	KeyDown	MouseUp
BeforeDropOrPaste	DropButtonClick	KeyPress	
BeforUpdate	Enter	KeyUp	
Change	Error	MouseDown	

### 5.1.4 ListBox

#### 5.1.4.1 PROPIEDADES

BackColor	ControlTipText	ListCount	OldLeft	Text
BorderColor	Default	ListIndex	OldTop	TextAlign
BorderStyle	Enabled	ListStyle	OldWidth	TextColumn
BottomRightCell	Font	Locked	Parent	Top
BoundColumn	ForeColor	MatchEntry	RowSource	TopIndex
Cancel	Height	MouseIcon	RowSourceType	Value
Column	HelpContextID	MousePointer	Selected	Visible
ColumnCount	IntegralHeight	MultiSelect	SpecialEffect	Width
ColumnHeads	LayoutEffect	Name	TabIndex	
ColumnWidths	Left	Object	TabStop	
ControlSource	List	OldHeight	Tag	

### 5.1.4.2 MÉTODOS

AddItem	RemoveItem
Clear	SetFocus
Move	ZOrder

### 5.1.4.3 EVENTOS

AfterUpdate	Click	Error	KeyUp
BeforeDragOver	Change	Exit	MouseDown
BeforeDropOrPaste	DblClick	KeyDown	MouseMove
BeforUpdate	Enter	KeyPress	MouseUp

## 5.1.5 CheckBox

### 5.1.5.1 PROPIEDADES

Accelerator	Default	Locked	Parent	Text
Alignment	Enabled	MouseIcon	Picture	TextAlign
Autosize	Font	MousePointer	PicturePosition	TextColumn
BackColor	ForeColor	Name	RowSource	Top
BackStyle	GroupName	Object	RowSourceType	TripleState
Cancel	Height	OldHeight	SpecialEffect	Value
Caption	HelpContextID	OldLeft	TabIndex	Visible
ControlSource	LayoutEffect	OldTop	TabStop	Width
ControlTipText	Left	OldWidth	Tag	WordWrap

### 5.1.5.2 MÉTODOS

Move
SetFocus
ZOrder

### 5.1.5.3 EVENTOS

AfterUpdate	Change	Error	KeyUp
BeforeDragOver	Click	Exit	MouseDown
BeforeDropOrPaste	DblClick	KeyDown	MouseMove
BeforUpdate	Enter	KeyPress	MouseUp

## 5.1.6 OptionButton

### 5.1.6.1 PROPIEDADES

Accelerator	Default	Locked	Parent	TextAlign
Alignment	Enabled	MouseIcon	Picture	Top
Autosize	Font	MousePointer	PicturePosition	TripleState
BackColor	ForeColor	Name	RowSource	Value
BackStyle	GroupName	Object	RowSourceType	Visible
Cancel	Height	OldHeight	SpecialEffect	Width
Caption	HelpContextID	OldLeft	TabIndex	WordWrap
ControlSource	LayoutEffect	OldTop	TabStop	
ControlTipText	Left	OldWidth	Tag	

### 5.1.6.2 MÉTODOS

Move
SetFocus
ZOrder

### 5.1.6.3 EVENTOS

AfterUpdate	Change	Error	KeyUp
BeforeDragOver	Click	Exit	MouseDown
BeforeDropOrPaste	DblClick	KeyDown	MouseMove
BeforUpdate	Enter	KeyPress	MouseUp

## 5.1.7 ToggleButton

### 5.1.7.1 PROPIEDADES

Accelerator	Default	Locked	Parent	TextAlign
Alignment	Enabled	MouseIcon	Picture	Top
Autosize	Font	MousePointer	PicturePosition	TripleState
BackColor	ForeColor	Name	RowSource	Value
BackStyle	GroupName	Object	RowSourceType	Visible
Cancel	Height	OldHeight	SpecialEffect	Width
Caption	HelpContextID	OldLeft	TabIndex	WordWrap
ControlSource	LayoutEffect	OldTop	TabStop	
ControlTipText	Left	OldWidth	Tag	

### 5.1.7.2 MÉTODOS

Move
SetFocus
ZOrder

### 5.1.7.3 EVENTOS

AfterUpdate	Change	Error	KeyUp
BeforeDragOver	Click	Exit	MouseDown
BeforeDropOrPaste	DblClick	KeyDown	MouseMove
BeforUpdate	Enter	KeyPress	MouseUp

## 5.1.8 Frame

### 5.1.8.1 PROPIEDADES

ActiveControl	ControlTipText	LayoutEffect	Picture	SpecialEffect
BackColor	Cycle	Left	PictureAlignment	TabIndex
BorderColor	Default	MouseIcon	PictureSizeMode	TabStop
BorderStyle	Enabled	MousePointer	PictureTiling	Tag
Cancel	Font	Name	RowSource	Top
CanPaste	ForeColor	Object	RowSourceType	TripleState
CanRedo	Height	OldHeight	ScrollBars	Value
CanUndo	HelpContextID	OldLeft	ScrollHeight	VerticalScrollBar-Side
Caption	InsideHeight	OldTop	ScrollLeft	Visible
Controls	InsideWidth	OldWidth	ScrollTop	Width
ControlSource	KeepScrollBarsVisible	Parent	ScrollWidth	Zoom

### 5.1.8.2 MÉTODOS

Copy	RedoAction	SetFocus
Cut	Repaint	UndoAction
Move	Scroll	ZOrder
Paste	SetDefaultTabOrder	

### 5.1.8.3 EVENTOS

AddControl	Change	KeyDown	MouseUp
AfterUpdate	DblClick	KeyPress	RemoveControl
BeforeDragOver	Enter	KeyUp	Scroll
BeforeDropOrPaste	Error	MouseDown	Zoom
Click	Exit	MouseMove	

## 5.1.9 CommandButton

### 5.1.9.1 PROPIEDADES

Accelerator	Default	Locked	OldWidth	TabStop
Autosize	Enabled	MouseIcon	Parent	Tag
BackColor	Font	MousePointer	Picture	TakeFocusOnClick
BackStyle	ForeColor	Name	PicturePosition	Top
Cancel	Height	Object	RowSource	Visible
Caption	HelpContextID	OldHeight	RowSourceType	Width
ControlSource	LayoutEffect	OldLeft	SpecialEffect	WordWrap
ControlTipText	Left	OldTop	TabIndex	

### 5.1.9.2 MÉTODOS

Move
SetFocus
ZOrder

### 5.1.9.3 EVENTOS

BeforeDragOver	Enter	KeyPress	MouseUp
BeforeDropOrPaste	Error	KeyUp	
Click	Exit	MouseDown	
DblClick	KeyDown	MouseMove	

## 5.1.10 TabStrip

### 5.1.10.1 PROPIEDADES

BackColor	Enabled	MultiRow	RowSourceType	Tag
Cancel	Font	Name	SelectedItem	Top
ClientHeight	ForeColor	Object	Style	Value
ClientLeft	Height	OldHeight	TabFixedHeight	Visible
ClientTop	HelpContextID	OldLeft	TabFixedWidth	Width
ClientWidth	LayoutEffect	OldTop	TabIndex	
ControlSource	Left	OldWidth	TabOrientation	
ControlTipText	MouseIcon	Parent	Tabs	
Default	MousePointer	RowSource	TabStop	

### 5.1.10.2 MÉTODOS

Move
SetFocus
ZOrder

### 5.1.10.3 EVENTOS

BeforeDragOver	DbClick	KeyDown	MouseMove
BeforeDropOrPaste	Enter	KeyPress	MouseUp
Click	Error	KeyUp	
Change	Exit	MouseDown	

## 5.1.11 MultiPage

### 5.1.11.1 PROPIEDADES

BackColor	ForeColor	Object	RowSource	TabOrientation
Cancel	Height	OldHeight	RowSourceType	TabStop
ControlSource	HelpContextID	OldLeft	SelectedItem	Tag
ControlTipText	LayoutEffect	OldTop	Style	Top
Default	Left	OldWidth	TabFixedHeight	Value
Enabled	MultiRow	Pages	TabFixedWidth	Visible
Font	Name	Parent	TabIndex	Width



### 5.1.11.2 MÉTODOS

Move
SetFocus
ZOrder

### 5.1.11.3 EVENTOS

AddControl	DblClick	KeyPress	MouseUp
BeforeDragOver	Enter	KeyUp	RemoveControl
BeforeDropOrPaste	Error	Layout	Scroll
Click	Exit	MouseDown	Zoom
Change	KeyDown	MouseMove	

## 5.1.12 ScrollBar

### 5.1.12.1 PROPIEDADES

BackColor	Height	MouseIcon	Orientation	Tag
Cancel	HelpContextID	MousePointer	Parent	Top
ControlSource	IntegralHeight	Name	ProportionalThumb	Value
ControlTipText	LargeChange	Object	RowSource	Visible
Default	LayoutEffect	OldHeight	RowSourceType	Width
Delay	Left	OldLeft	SmallChange	
Enabled	Max	OldTop	TabIndex	
ForeColor	Min	OldWidth	TabStop	

### 5.1.12.2 MÉTODOS

Move
SetFocus
ZOrder

### 5.1.12.3 EVENTOS

AfterUpdate	Change	Exit	KeyUp
BeforeDragOver	Enter	KeyDown	Scroll
BeforeDropOrPaste	Error	KeyPress	

## 5.1.13 SpinButton

### 5.1.13.1 PROPIEDADES

BackColor	ForeColor	MouseIcon	OldWidth	TabIndex
Cancel	Height	MousePointer	Orientation	TabStop
ControlSource	HelpContextID	Name	Parent	Tag
ControlTipText	LayoutEffect	Object	ProportionalThumb	Top
Default	Left	OldHeight	RowSource	Value
Delay	Max	OldLeft	RowSourceType	Visible
Enabled	Min	OldTop	SmallChange	Width

### 5.1.13.2 MÉTODOS

Move
SetFocus
ZOrder

### 5.1.13.3 EVENTOS

AfterUpdate	Change	KeyDown	SpinUp
BeforeDragOver	Enter	KeyPress	
BeforeDropOrPaste	Error	KeyUp	
BeforUpdate	Exit	SpinDown	

## 5.1.14 Image

### 5.1.14.1 PROPIEDADES

AutoSize	Default	MouseIcon	Parent	TabIndex
BackColor	Enabled	MousePointer	Picture	TabStop
BackStyle	Font	Name	PictureAlignment	Tag
BorderColor	ForeColor	Object	PictureSizeMode	Top
BorderStyle	Height	OldHeight	PictureTiling	Visible
Cancel	HelpContextID	OldLeft	RowSource	Width
ControlSource	LayoutEffect	OldTop	RowSourceType	
ControlTipText	Left	OldWidth	SpecialEffect	

### 5.1.14.2 MÉTODOS

Move
SetFocus
ZOrder

### 5.1.14.3 EVENTOS

BeforeDragOver	DblClick	MouseMove
BeforeDropOrPaste	Error	MouseUp
Click	MouseDown	

## 5.2 RESUMEN DE PROPIEDADES

Propiedad	Comentario		Valores
Accelerator	Define o consulta la tecla de aceleración de un control.		String
ActiveControl	Permite manipular e identificar el control que tiene el foco.		
Alignment	Define o consulta la posición del control en relación a su título. Puede ser alguno de los valores de la constante fmAlignment:		fmAlignment
	0	<b>fmAlignment-Left</b>	Coloca el título a la izquierda del control.
	1	<b>FmAlignment-Right</b>	Coloca el título a la derecha del control (predeterminado).

AutoSize	Indica si el objeto ha de cambiar su tamaño automáticamente para mostrar su contenido.	Boolean	
AutoTab	Indica si se ha de producir una tabulación automática cuando el usuario sobrepasa el máximo de caracteres permitido en la parte del cuadro de texto.	Boolean	
AutoWordSelect	Indica si la selección ha de extenderse a una palabra o a un carácter.	Boolean	
BackColor	Color de fondo. La propiedad BackStyle debe establecerse a fmBackStyleOpaque. Puede definirse usando la función RGB (rojo, verde y azul).	Long	
BackStyle	Define el estilo del fondo. Puede ser alguno de los valores de la constante <b>fmBackStyle</b>		fmBackStyle
	0	<b>fmBackStyle-Transparent</b>	El fondo es transparente.
	1	<b>fmBackStyleOpaque</b>	El fondo es opaco. Valor por defecto.
BorderColor	Indica el color del borde de un objeto.	Long	
BorderStyle	Indica el tipo de borde de un objeto. Puede ser alguno de los valores de la constante <b>fmBorderStyle</b> :	fmBorderStyle	
	0	<b>fmBorderStyle-None</b>	El control no tiene una línea de borde visible.
	1	<b>fmBorderStyle-Single</b>	El control tiene un borde de una sola línea (predeterminado).

BoundColumn	Indica el elemento o columna que se utilizará como valor del control cuando un usuario seleccione una fila. Por ejemplo, si una fila contiene 3 columnas y se indica 2, el valor del objeto será el correspondiente a la columna 2 de la fila seleccionada.	Variant
Cancel	Permite cancelar cambios no confirmados.	Boolean
CanPaste	Indica si se puede efectuar un pegado desde el portapapeles.	Boolean
CanRedo	Indica si es posible invertir la ultima ejecución del comando Deshacer.	Boolean
CanUndo	Indica si es posible Deshacer la última acción realizada por el usuario.	Boolean
Caption	Descripción de la etiqueta.	String
ClientHeight	Alto del área de presentación del control TabStrip.	Single
ClientLeft	Ubicación de la posición izquierda del área de presentación del control TabStrip.	Single
ClientTop	Ubicación de la posición superior del área de presentación del control TabStrip.	Single
ClientWidth	Ancho del área de presentación del control TabStrip.	Single
Controls	Representa un objeto de tipo Controls con los controles que se hallan contenidos dentro del objeto indicado.	Controls
Column	Indica uno o varios elementos de un control ListBox o ComboBox.	
ColumnCount	Indica el número de columnas a mostrar. Si se indica -1, se muestran todas las columnas disponibles.	Long

ColumnHeads	Muestra una fila de títulos de columna.	Boolean	
ColumnWidths	Indica el ancho de cada una de las columnas del control. Los diferentes anchos se indican usando el punto y coma (;) como separador. Por ejemplo 50; 70; 60.	String	
ControlSource	Especifica la ubicación de los datos que se utiliza para la propiedad Value. Por ejemplo, si introducimos “Hoja1!A1” en esta propiedad, el control mostrará el contenido de la celda A1 y si el valor del control se modifica, dicha celda también lo hará.	String	
ControlTipText	Permite indicar el texto que se mostrará al pasar el puntero del mouse por encima del control.	String	
CurLine	Indica la línea actual del control.	Long	
CurTargetX	Indica la posición horizontal preferida del punto de inserción en un control medida en décimas de milímetro.	Long	
CurX	Indica la posición horizontal actual del punto de inserción en un control medida en décimas de milímetro.	Long	
Cycle	Indica el recorrido cíclico a seguir cuando se abandone el último control de un <b>Frame</b> o de un control <b>MultiPage</b> . Puede tener alguno de los valores de la constante <b>fmCycle</b> :		
	0	<b>fmCycleAllForms</b>	Recorrido a través de todos los controles del formulario, Frame y MultiPage que se muestran en el formulario.
	2	<b>FmCycleCurrentForm</b>	Recorrido y foco dentro del control en curso (formulario, Frame o MultiPage).

Default	Determina cual es control por defecto en un formulario. Usado especialmente en los controles de tipo botón.		Boolean
Delay	Indica el tiempo de retardo que se produce entre los eventos SpinUp, SpinDown y Change. Al hacer clic y presionar un botón, el primer evento se realiza inmediatamente pero, la segunda ocurrencia del evento será de 5 veces el valor de Delay. Los siguientes eventos se iniciarán después según lo indicado en Delay. Por defecto Delay vale 50 milisegundos.		Long
DragBehavior	Indica si la característica de arrastrar y colocar está habilitada. Puede ser alguno de los valores de la constante fmDragBehavior:		fmDragBehavior
	0	<b>fmDragBehavior-Disabled</b>	No permite arrastrar y colocar (predeterminado).
	1	<b>FmDragBehavior-Enabled</b>	Permite arrastrar y colocar.
DropButtonStyle	Indica el símbolo a mostrar en el botón que despliega la lista de valores. Puede ser alguno de los valores de la constante fmDropButtonStyle:		fmDropButtonStyle
	0	<b>fmDropButtonStylePlain</b>	Botón sin símbolo.
	1	<b>fmDropButtonStyle-Arrow</b>	Flecha hacia abajo (Valor por defecto).
	2	<b>fmDropButtonStyle-Ellipsis</b>	Puntos suspensivos (...).
	3	<b>fmDropButtonStyle-Reduce</b>	Muestra una línea horizontal.



Enabled	Indica si el objeto puede recibir el foco y si puede responder a eventos generados por un usuario.	Boolean						
EnterField Behavior	Indica cómo se selecciona el texto y puede ser alguno de los valores de la constante fmEnterFieldBehavior:	fmEnterField Behavior						
	<table> <tr> <td>0</td><td>fmEnterFieldBehavior-SelectAll</td><td>Selecciona todo el contenido al entrar en el control (valor por defecto).</td></tr> <tr> <td>1</td><td>fmEnterFieldBehaviorRecall-Selection</td><td>Mantiene la selección.</td></tr> </table>	0	fmEnterFieldBehavior-SelectAll	Selecciona todo el contenido al entrar en el control (valor por defecto).	1	fmEnterFieldBehaviorRecall-Selection	Mantiene la selección.	
0	fmEnterFieldBehavior-SelectAll	Selecciona todo el contenido al entrar en el control (valor por defecto).						
1	fmEnterFieldBehaviorRecall-Selection	Mantiene la selección.						
EnterKeyBehavior	Define el comportamiento de la pulsación de la tecla ENTER. Si True, al pulsar ENTER se crea una nueva línea. Si False, el foco se mueve al siguiente control situado en el orden de tabulación.	Boolean						
Font	Permite definir o consultar el tipo de fuente asociada al objeto.	Font						
ForeColor	Color del primer plano del objeto. Puede definirse usando la función RGB (rojo, verde y azul) o un valor de tipo Long que represente un color.	Long						
GroupName	Define o consulta un nombre de grupo para una serie de controles.	String						
Height	Alto del objeto en puntos.	Single						
HelpContextID	Permite asociar un tema determinado de un archivo de ayuda de MS Windows.	Long						
HideSelection	Indica si se ha de mantener resaltado el texto seleccionado cuando el control pierde el foco.	Boolean						

IMEMode	Indica el modo de tiempo de ejecución del editor de métodos de entrada (IME) y se utiliza sólo para aplicaciones escritas para Extremo Oriente.	fmIMEMode												
InsideHeight	Devuelve el alto en puntos de la región del cliente (parte de la ventana donde se muestra la salida) dentro de un formulario.	Single												
InsideWidth	Devuelve el ancho en puntos de la región del cliente (parte de la ventana donde se muestra la salida) dentro de un formulario.	Single												
IntegralHeight	Indica si el control muestra las líneas completas o las muestra parciales. Si True, el cuadro cambia de tamaño automáticamente para poder mostrar las filas completas.	Boolean												
KeepScrollBars-Visible	Indica si se han de mostrar o no las barras de desplazamiento. Puede tener alguno de los valores de la constante fmScrollBars:	fmScrollBars												
	<table> <tr> <td>0</td><td><b>fmScrollBarsNone</b></td><td>No muestra ninguna barra.</td></tr> <tr> <td>1</td><td><b>fmScrollBarsHorizontal</b></td><td>Muestra barra horizontal.</td></tr> <tr> <td>2</td><td><b>fmScrollBarsVertical</b></td><td>Muestra barra vertical.</td></tr> <tr> <td>3</td><td><b>fmScrollBarsBoth</b></td><td>Muestra ambas barras (valor por defecto).</td></tr> </table>	0	<b>fmScrollBarsNone</b>	No muestra ninguna barra.	1	<b>fmScrollBarsHorizontal</b>	Muestra barra horizontal.	2	<b>fmScrollBarsVertical</b>	Muestra barra vertical.	3	<b>fmScrollBarsBoth</b>	Muestra ambas barras (valor por defecto).	
0	<b>fmScrollBarsNone</b>	No muestra ninguna barra.												
1	<b>fmScrollBarsHorizontal</b>	Muestra barra horizontal.												
2	<b>fmScrollBarsVertical</b>	Muestra barra vertical.												
3	<b>fmScrollBarsBoth</b>	Muestra ambas barras (valor por defecto).												
LargeChange	Indica la cantidad de desplazamiento que se ha de producir al hacer clic entre el cuadro de desplazamiento y la flecha.	Long												
LayoutEffect	Indica si un control se ha movido en tiempo de diseño. Puede tener alguno de los valores de la constante fmLayoutEffect:	fmLayoutEffect												

	0	<b>fmLayoutEffect-None</b>	El control no se ha movido.	
	1	<b>fmLayoutEffect-Initiate</b>	El control se ha movido.	
Left	Distancia entre el borde izquierdo del objeto que lo contiene y el propio control.			Single
LineCount	Indica el número de líneas de texto del control.			Integer
List	Define o consulta las entradas de la lista del control.			Matriz o conjunto de elementos
ListCount	Indica el número de entradas de la lista.			Long
ListFillRange	Define o consulta el rango utilizado para rellenar el cuadro de lista indicado.			String
ListIndex	Indica el elemento seleccionado actualmente en el control.			Variant
ListRows	Indica el número máximo de filas que se muestran.			Long
ListStyle	Indica la apariencia de la lista de elementos en el control. Puede ser alguno de los valores de la constante <b>fmListStyle</b> :			fmListStyle
	0	<b>fmListStylePlain</b>	Se muestra con el fondo de los elementos resaltado.	
	1	<b>fmListStyleOption</b>	Muestra OptionButtons o CheckBox para una lista de selección múltiple (Valor por defecto).	
ListWidth	Indica el ancho de la lista en el control.			Variant
Locked	Indica si el control puede ser editado			Boolean

MatchEntry	Define o consulta la forma de cómo se realiza la búsqueda en el control cuando el usuario escribe. Puede ser alguno de los valores de la constante fmMatchEntry:	fmMatchEntry									
	<table> <tr> <td>0</td><td><b>fmMatchEntry-FirstLetter</b></td><td>El control busca entrada que empiece con el carácter introducido y así sucesivamente.</td></tr> <tr> <td>1</td><td><b>fmMatchEntry-Complete</b></td><td>Según se van escribiendo caracteres se busca una entrada que coincida con todos los caracteres introducidos (valor por defecto).</td></tr> <tr> <td>2</td><td><b>fmMatchEntry-None</b></td><td>Sin coincidencia.</td></tr> </table>	0	<b>fmMatchEntry-FirstLetter</b>	El control busca entrada que empiece con el carácter introducido y así sucesivamente.	1	<b>fmMatchEntry-Complete</b>	Según se van escribiendo caracteres se busca una entrada que coincida con todos los caracteres introducidos (valor por defecto).	2	<b>fmMatchEntry-None</b>	Sin coincidencia.	
0	<b>fmMatchEntry-FirstLetter</b>	El control busca entrada que empiece con el carácter introducido y así sucesivamente.									
1	<b>fmMatchEntry-Complete</b>	Según se van escribiendo caracteres se busca una entrada que coincida con todos los caracteres introducidos (valor por defecto).									
2	<b>fmMatchEntry-None</b>	Sin coincidencia.									
MatchFound	Indica si el texto introducido coincide con alguna de las entradas del control.	Boolean									
MatchRequired	Indica si es obligatorio que el texto introducido en el control coincida con alguna de las entradas del mismo. Si True, el usuario no puede salir del control hasta que el texto introducido coincida con alguna de las entradas.	Boolean									
Max	Valor máximo para la propiedad Value.	Long									
MaxLength	Indica el número máximo de caracteres permitido.	Long									
Min	Valor mínimo para la propiedad Value.	Long									
MouseIcon	Icono asociado al control cuando el mouse pasa por encima del mismo. La propiedad MousePointer debe establecerse a 99.										
MousePointer	Tipo de puntero mostrado. Debe tener alguno de los valores de la constante <b>fmMousePointer</b> :										

0	<b>fmMousePointer-Default</b>	Puntero estándar. Es el valor por defecto.
1	<b>fmMousePointer-Arrow</b>	Tipo flecha.
2	<b>fmMousePointer-Cross</b>	Tipo cruz.
3	<b>fmMousePointer-IBeam</b>	Tipo I.
6	<b>fmMousePointerSize-NESW</b>	Flecha doble señalando al noreste y al suroeste.
7	<b>fmMousePointerSize-NS</b>	Flecha doble señalando al norte y al sur.
8	<b>fmMousePointerSize-NWSE</b>	Flecha doble señalando al noroeste y al sureste.
9	<b>fmMousePointerSize-WE</b>	Flecha doble señalando al oeste y al este.
10	<b>fmMousePointerUp-Arrow</b>	Flecha arriba.
11	<b>fmMousePointer-Hourglass</b>	Reloj de arena.
12	<b>fmMousePointerNo-Drop</b>	Círculo con una línea diagonal indicando ‘destino prohibido’.
13	<b>fmMousePointer-AppStarting</b>	Flecha con reloj de arena.
14	<b>fmMousePointer-Help</b>	Flecha con interrogación.
15	<b>fmMousePointer-SizeAll</b>	Flecha señalando al norte, al sur, al este y al oeste.
99	<b>fmMousePointer-Custom</b>	Usa el icono indicado en la propiedad <b>MouseIcon</b> .

MultiLine	Define o consulta si se pueden introducir múltiples líneas de texto.	Boolean									
MultiRow	Indica si el control puede tener más de una fila de fichas. Si True, puede haber mas de una fila de fichas. Utilizado en el control TabStrip.	Boolean									
MultiSelect	Indica si se pueden realizar selecciones múltiples. Puede ser alguno de los valores de la constante fmMultiSelect:	fmMultiSelect									
	<table><tr><td>0</td><td>fmMultiSelectSingle</td><td>Sólo puede seleccionarse un elemento (valor por defecto).</td></tr><tr><td>1</td><td>fmMultiSelectMulti</td><td>BARRA ESPACIADORA o clic selecciona o anula la selección de un elemento.</td></tr><tr><td>2</td><td>fmMultiSelectExtended</td><td>Presionar MAYÚSCULAS y hacer clic con el mouse o una de la flechas de dirección para seleccionar desde el elemento seleccionado hasta el elemento actual. CTRL y hacer clic con el mouse selecciona o anula la selección de un elemento.</td></tr></table>	0	fmMultiSelectSingle	Sólo puede seleccionarse un elemento (valor por defecto).	1	fmMultiSelectMulti	BARRA ESPACIADORA o clic selecciona o anula la selección de un elemento.	2	fmMultiSelectExtended	Presionar MAYÚSCULAS y hacer clic con el mouse o una de la flechas de dirección para seleccionar desde el elemento seleccionado hasta el elemento actual. CTRL y hacer clic con el mouse selecciona o anula la selección de un elemento.	
0	fmMultiSelectSingle	Sólo puede seleccionarse un elemento (valor por defecto).									
1	fmMultiSelectMulti	BARRA ESPACIADORA o clic selecciona o anula la selección de un elemento.									
2	fmMultiSelectExtended	Presionar MAYÚSCULAS y hacer clic con el mouse o una de la flechas de dirección para seleccionar desde el elemento seleccionado hasta el elemento actual. CTRL y hacer clic con el mouse selecciona o anula la selección de un elemento.									
Name	Nombre del objeto.	String									
Object	Permite sobrescribir un método o propiedad estándar en caso de que un nuevo control tenga un método o propiedad con el mismo nombre.	Object									
OldHeight	Indica el alto que tenia el control antes de que se ajustara su tamaño.	Single									

OldLeft	Indica la posición izquierda que tenía el control antes de que se moviera.			Single
OldTop	Indica la posición superior que tenía el control antes de que se moviera.			Single
OldWidth	Indica el ancho que tenía el control antes de que se ajustara su tamaño.			Single
Orientation	Indica si la orientación del control es horizontal o vertical. Puede ser alguno de los valores de la constante <b>fmOrientation</b> :			fmOrientation
	-1	<b>fmOrientation-Auto</b>	Determina automáticamente la orientación basándose en las dimensiones del control (predeterminado).	
	0	<b>fmOrientation-Vertical</b>	El control se representa verticalmente.	
	1	<b>fmOrientation-Horizontal</b>	El control se representa horizontalmente.	
Pages	Colección de objetos de tipo Page que contiene un control de tipo MultiPage.			Pages
Parent	Proporciona el nombre del objeto o colección que contiene al objeto en cuestión.			String
PasswordChar	Permite introducir un determinado carácter en lugar del carácter tecleado. Útil para ocultar la entrada de texto.			String
Picture	Indica el mapa de bits mostrado en el objeto. Para cargar dicho mapa de bits debe utilizar LoadPicture( ruta ) siendo <b>ruta</b> la ruta donde se halla el archivo de imagen.			

PictureAlignment	Indica la ubicación de una imagen de fondo. Puede ser alguno de los valores de la constante fmPictureAlignment:		fmPictureAlignment	
	0	<b>fmPictureAlignmentTop-Left</b>	Esquina superior izquierda.	
	1	<b>fmPictureAlignmentTop-Right</b>	Esquina superior derecha.	
	2	<b>fmPictureAlignment-Center</b>	Centro.	
	3	<b>fmPictureAlignment-BottomLeft</b>	Esquina inferior izquierda.	
	4	<b>fmPictureAlignment-BottomRight</b>	Esquina inferior derecha.	
PicturePosition	Indica donde ha de colocarse la imagen en relación al título del objeto (ver CommandButton).			
PictureSizeMode	Indica cómo se ha de mostrar la imagen de fondo en un control. Puede tener alguno de los valores de la constante fmPictureSizeMode:		fmPictureSizeMode	
	0	<b>fmPictureSizeModeClip</b>	Recorta la imagen si es más grande que el formulario o página (valor por defecto).	
	1	<b>fmPictureSizeMode-Stretch</b>	Estira la imagen rellenando el formulario o página. Deforma la imagen.	
	3	<b>fmPictureSizeMode-Zoom</b>	Amplía la imagen, sin deformarla.	



PictureTiling	Indica si la imagen se ha de organizar en forma de mosaico cuando ésta posea un tamaño menor que el formulario o página que la contiene.	Boolean			
ProportionalThumb	Indica si el tamaño del cuadro de desplazamiento ha de ser proporcional al área de desplazamiento o fijo. Si False, el cuadro de desplazamiento será de tamaño fijo.	Boolean			
RowSource	Indica cual es la fuente de datos que proporciona el conjunto de elementos a proporcionar como lista a un control. Por ejemplo, si introducimos “Hoja1!A1:A4” en esta propiedad para un control de tipo ListBox, observaremos que cada uno de los valores existentes en cada celda pasará a visualizarse como un elemento del control.	String			
RowSourceType	Permite indicar deben proporcionarse los datos a un ListBox, ComboBox u objeto OLE. Puede tener alguno de los siguientes valores: <div><table><tr><td>Table/Query</td></tr><tr><td>Value List</td></tr><tr><td>Field List</td></tr></table></div>	Table/Query	Value List	Field List	String
Table/Query					
Value List					
Field List					
ScrollBars	Indica si se ha de mostrar la barra de desplazamiento vertical, horizontal o ambas. Puede ser alguno de los valores de la constante fmScrollBars:	fmScrollBars			

	0	<b>fmScrollBarsNone</b>	No se muestra ninguna barra de desplazamiento (valor por defecto).
	1	<b>fmScrollBars-Horizontal</b>	Muestra barra de desplazamiento horizontal.
	2	<b>fmScrollBars-Vertical</b>	Muestra barra de desplazamiento vertical.
	3	<b>fmScrollBarsBoth</b>	Muestra ambas barras de desplazamiento.
ScrollHeight	Indica el alto en puntos que puede mostrarse al mover las barras de desplazamiento.		Single
ScrollLeft	Indica la distancia en puntos desde el borde izquierdo del formulario.		Single
ScrollTop	Indica la distancia en puntos desde el borde superior del formulario.		
ScrollWidth	Indica el ancho en puntos que puede mostrarse al mover las barras de desplazamiento.		Single
Selected	Define o consulta el estado de un elemento de un ListBox indicando si está seleccionado o no.		Boolean
SelectedItem	Indica cual es el elemento seleccionado en la actualidad. Usado en controles de tipo TabStrip y MultiPage.		
SelectionMargin	Indica si la selección puede realizarse haciendo clic en el margen que hay a la izquierda del texto.		Boolean
SelLength	Indica el número de caracteres seleccionados en el texto.		Long

SelStart	Indica el inicio del texto seleccionado.	Long
SelText	Indica el texto seleccionado.	String
ShowDrop-ButtonWhen	Indica cuando se ha de mostrar el botón de desplegar del control. Puede ser alguno de los valores de la constante fmShowDropButtonWhen:	fmShowDrop-ButtonWhen
	0 <b>fmShowDropButtonWhen-Never</b>	Nunca.
	1 <b>FmShowDropButtonWhen-Focus</b>	Muestra el botón cuando tiene el foco.
	2 <b>FmShowDropButtonWhen-Always</b>	Siempre lo muestra (valor por defecto).
SmallChange	Indica la cantidad de desplazamiento que se ha de producir al hacer clic en una flecha de desplazamiento.	Long
SourceName	Define o consulta el nombre de origen del vínculo del objeto.	String
SpecialEffect	Indica la apariencia visual de un objeto. Puede ser alguno de los valores de la constante <b>fmSpecialEffect</b> :	fmSpecialEffect

Ra-Ma®

	0	<b>fmSpecialEffectFlat</b>	El objeto se muestra plano, distinguido por un borde, un cambio de color o ambos. (Valor por defecto).
	1	<b>FmSpecialEffect-Raised</b>	Resaltado en la parte superior - izquierda y con una sombra en la parte inferior - derecha. No válido para CheckBox ni OptionButton.
	2	<b>FmSpecialEffect-Sunken</b>	El objeto tiene una sombra en la parte superior - izquierda y está resaltado en la parte inferior - derecha. Valor por defecto para los controles CheckBox y OptionButton.
	3	<b>FmSpecialEffect-Etched</b>	El borde parece estar cincelado alrededor del control.  No válido para CheckBox ni OptionButton.
	6	<b>FmSpecialEffect-Bump</b>	Relieve en la parte inferior - derecha, y plano en la parte superior - izquierda. No válido para CheckBox ni OptionButton.

Style	Indica cómo puede seleccionar o definir el valor del control. Puede ser alguno de los valores de la constante <b>fmStyle</b> :		fmStyle
	0	<b>fmStyleDropDown-Combo</b>	Se puede escribir un valor en el área de edición o seleccionar un valor de la lista (valor por defecto).
	2	<b>FmStyleDropDownList</b>	El usuario debe seleccionar un valor de la lista.
TabFixedHeight	Define o consulta el alto en puntos de las fichas de los controles <b>TabStrip</b> o <b>MultiPage</b> .		Single
TabFixedWidth	Define o consulta el ancho en puntos de las fichas de los controles <b>TabStrip</b> o <b>MultiPage</b> .		Single
TabIndex	Orden en el que se halla el control dentro del orden de tabulación definido para el paso de un control a otro cada vez que se pulsa la tecla TAB.		Integer
TabOrientation	Define o consulta dónde han de ubicarse las fichas en los controles <b>TabStrip</b> o <b>MultiPage</b> . Puede tener alguno de los valores de la constante <b>fmTabOrientation</b> :		fmTabOrientation
	0	<b>fmTabOrientationTop</b>	Mostradas en la parte superior (valor por defecto).
	1	<b>FmTabOrientation-Bottom</b>	Mostradas en la parte inferior.
	2	<b>FmTabOrientationLeft</b>	Mostradas en la parte izquierda.
	3	<b>FmTabOrientationRight</b>	Mostradas en la parte derecha.

Tabs	Colección de objetos que representa todos los Tabs de un control de tipo TabStrip.	Tabs	
TabStop	Define o consulta si el control puede recibir el foco cuando se ha accedido al mismo mediante la pulsación de la tecla TAB.	Boolean	
Tag	Permite almacenar información adicional sobre el objeto.	String	
TakeFocusOnClick	Indica si el control recibe el foco cuando se hace clic sobre el mismo.	Boolean	
Text	Define o consulta el texto del cuadro de texto del control.	String	
TextAlign	Indica cómo debe alinearse el texto. Puede ser alguno de los valores de a constante fmTextAlign:		
	1	<b>fmTextAlignLeft</b>	Alineación a la izquierda. (valor por defecto).
	2	<b>FmTextAlignCenter</b>	Centra el texto.
	3	<b>FmTextAlignRight</b>	Alineación a la derecha.
TextColumn	Indica cual es la columna que debe almacenarse en la propiedad text cuando el usuario selecciona una fila.	Variant	
TextLength	Indica la longitud del texto existente en el área de edición del control.	Long	
Top	Distancia entre el borde superior del objeto que lo contiene y el propio control.	Single	
TopIndex	Define o consulta el primer elemento de la lista.	Variant	
TripleState	Indica si se puede definir el estado nulo para el control.	Boolean	

Value	Indica el valor o estado de un control.		Variant
VerticalScrollbarSide	Define o consulta en qué lado se desea visualizar la barra de desplazamiento. Puede tener alguno de los valores de la constante fmVerticalScrollbarSide:		fmVerticalScrollbarSide
	0	<b>fmVerticalScrollbar-SideRight</b>	Se coloca a la derecha (valor por defecto).
	1	<b>fmVerticalScrollbar-SideLeft</b>	Se coloca a la izquierda.
Visible	Indica si el objeto está visible o no.		Boolean
Width	Ancho del objeto en puntos.		Single
WordWrap	Indica si el contenido del control se ajusta al final de la línea automáticamente.		Boolean
Zoom	Indica el % del cambio que se ha de aplicar al tamaño del objeto que se está mostrando.		Integer

### 5.3 RESUMEN DE MÉTODOS

Nombre	Comentario
AddItem	Añade un elemento a la lista.
Clear	Elimina todos los elementos de la lista.
Copy	Copia el objeto en el portapapeles.
Cut	Corta el objeto pegándolo en el portapapeles.
DropDown	Permite desplegar la lista de elementos.
Move	Permite mover un control a una posición determinada.
Paste	Copia el contenido del portapapeles al objeto.
RedoAction	Permite invertir el resultado de la última acción Undo ejecutada-
RemoveItem	Elimina una fila del control.

Repaint	Vuelve a dibujar la pantalla o formulario.
Scroll	Permite mover la barra de desplazamiento de un control.
SetDefaultTabOrder	Permite establecer el orden en el que se recorrerán los objetos cada vez que se pulse la tecla TAB. El orden establecido por defecto empieza con el control situado más a la izquierda de la parte superior.
SetFocus	Permite colocar el foco en el control indicado.
UndoAction	Invierte la acción realizada por el último comando Deshacer.
ZOrder	Permite colocar el objeto al principio o al final del orden-z (orden en el que se dibujan unas imágenes por encima o por debajo de otras).

## 5.4 RESUMEN DE EVENTOS

Nombre	Comentario		
AddControl	Producido cuando se inserta un control en un formulario en tiempo de ejecución, Frame u objeto Page.		
AfterUpdate	Producido tras cambiar los datos de un control.		
BeforeDragOver	Producido durante la acción de arrastrar y colocar. El estado del arrastre puede tener alguno de los valores de la constante <b>fmDragState</b> :		
	0	<b>fmDragStatusEnter</b>	El puntero del mouse está dentro del destino.
	1	<b>FmDragStatusLeave</b>	El puntero del mouse está fuera del destino.
	2	<b>FmDragStatusOver</b>	El puntero del mouse está en una posición nueva, pero dentro del destino.
	El efecto producido por el arrastre puede tener alguno de los siguientes valores:		



	0	<b>fmDropEffectNone</b>	No realiza ninguna acción.						
	1	<b>FmDropEffectCopy</b>	Copia el origen al destino.						
	2	<b>FmDropEffectMove</b>	Mueve el origen al destino.						
	3	<b>FmDropEffectCopyOrMove</b>	Copia o mueve el origen al destino.						
BeforeDropOrPaste	Producido antes de pegar o colocar datos en un objeto. Los valores que puede tomar la acción a realizar puede ser alguno de los siguientes: <table><tr><td>2</td><td><b>fmActionPaste</b></td><td>Pega el objeto seleccionado en el destino de colocación.</td></tr><tr><td>3</td><td><b>FmActionDragDrop</b></td><td>Indica que el usuario ha arrastrado el objeto desde su origen al destino de colocación y lo ha colocado en el destino deseado.</td></tr></table> <p>Los valores del efecto, puede consultarlos en la descripción del evento <b>BeforeDragOver</b>.</p>			2	<b>fmActionPaste</b>	Pega el objeto seleccionado en el destino de colocación.	3	<b>FmActionDragDrop</b>	Indica que el usuario ha arrastrado el objeto desde su origen al destino de colocación y lo ha colocado en el destino deseado.
2	<b>fmActionPaste</b>	Pega el objeto seleccionado en el destino de colocación.							
3	<b>FmActionDragDrop</b>	Indica que el usuario ha arrastrado el objeto desde su origen al destino de colocación y lo ha colocado en el destino deseado.							
BeforUpdate	Producido antes de cambiar los datos de un control.								
Change	Producido al cambiar el valor de la propiedad Value.								
Click	Producido al hacer clic con el botón izquierdo del mouse sobre un objeto o al seleccionar un valor en caso de que haya más de uno.								
DbClick	Producido al hacer clic 2 veces seguidas con el mouse sobre un objeto.								
DropButtonClick	Producido al desplegar u ocultar la lista de elementos del control.								

Enter	Producido un instante antes de que el control reciba el foco.
Error	Producido al detectar un error.
Exit	Producido justo después de que el control pierda el foco.
KeyDown	Producido cuando el usuario pulsa una tecla.
KeyPress	Producido cuando el usuario ha pulsado una tecla.
KeyUp	Producido cuando el usuario libera una tecla pulsada.
Layout	Producido al cambiar el tamaño de un formulario, Frame o control MultiPage.
MouseDown	Producido al pulsar un botón del mouse.
MouseMove	Producido al mover el mouse.
MouseUp	Producido al soltar un botón del mouse.
RemoveControl	Producido al eliminar un control de un contenedor.
Scroll	Producido al mover un cuadro de desplazamiento en un ScrollBar, MultiPage o Frame.
SpinDown	Producido al hacer clic sobre la flecha inferior o izquierda.
SpinUp	Producido al hacer clic sobre la flecha superior o derecha.
Zoom	Producido al cambiar el valor de la propiedad Zoom (en controles Frame y MultiPage).

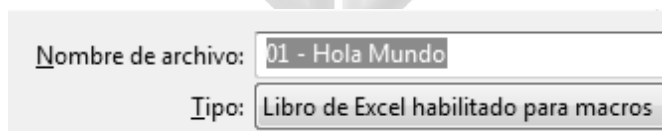
Ra-Ma®

## EJEMPLOS

---

### 6.1 HOLA MUNDO

Vamos a iniciar nuestra colección de ejemplos con el original ejemplo ‘**Hola Mundo**’. En este caso, el saludo lo realizaremos mediante un mensaje utilizando la instrucción **MsgBox** la cual, la ubicaremos en un procedimiento que fabricaremos en un módulo nuevo. Para ello, lo primero que haremos es crear un libro denominado **01 - Hola Mundo** y lo guardaremos mediante **Archivo->Guardar como** con el formato **Libro de Excel habilitado para macros**:



Nombre de archivo: 01 - Hola Mundo

Tipo: Libro de Excel habilitado para macros

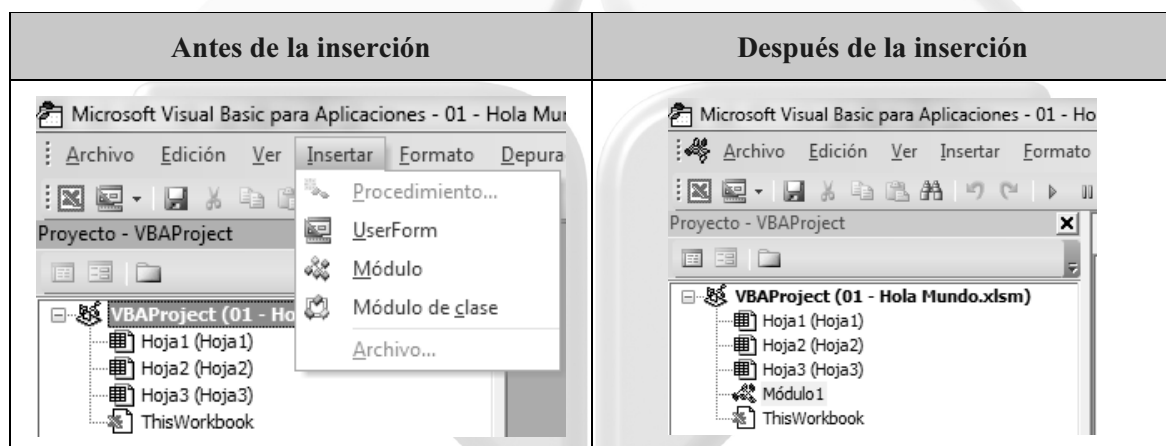
Como ya sabemos, los libros de Excel habilitados para contener macros, poseen la extensión **xlsm** y por tanto, el nombre completo del archivo será: **01 - Hola Mundo.xlsm**.

Una vez nos hallamos en el libro recién creado, hemos de pasar al entorno de **Visual Basic para Aplicaciones** y para ello, podemos simplemente acudir a la Ficha **Programador** y, dentro de la misma, haciendo clic sobre el icono **Visual Basic**



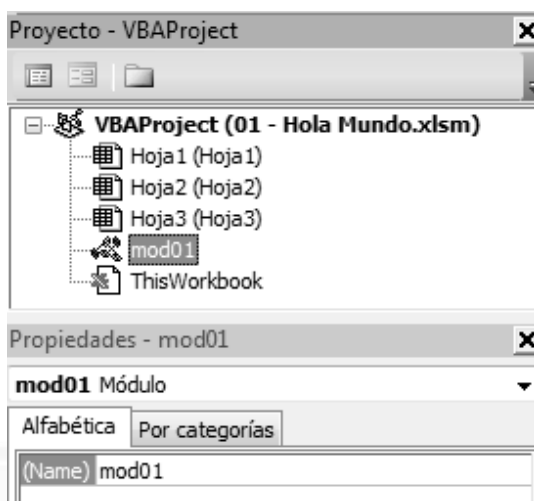
También podemos ir directamente a dicho entorno, pulsando simplemente la combinación de teclas **ALT** y **F11** simultáneamente.

En el entorno **VBA**, insertaremos un módulo para albergar nuestros procedimientos. Esta inserción la realizaremos seleccionando el proyecto denominado con el mismo nombre que el libro y a continuación, a través del menú **Insertar-> Módulo**:

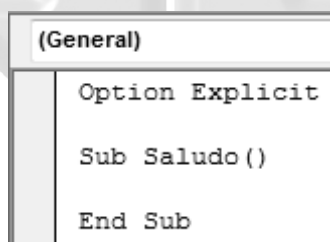


Observaremos que efectivamente se ha creado un módulo que por defecto se denomina **Módulo1**. Para cambiar el nombre al módulo, haremos clic sobre el mismo en el **Explorador de proyectos** y a continuación, sobre la **Ventana de propiedades** (pulsar **F4** si no está visible) localizaremos la propiedad **Name** y cambiaremos el nombre **Módulo1** a **mod01**:

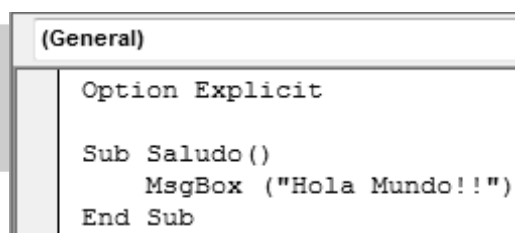
Ra-Ma®



A continuación, haremos clic sobre la **Ventana de código** (pulsar **F7** si no está visible) y nos dispondremos a escribir nuestro primer procedimiento denominado **Saludo()**. Según vimos en el capítulo de **Lenguaje de programación**, los procedimientos se definen introduciendo la partícula **Sub** y a continuación el nombre del mismo seguido de los parámetros separados por coma (si los tiene) entre paréntesis. Si no tiene parámetros, simplemente indicaremos un abrir y cerrar paréntesis. Por otra parte, el compilador reconoce el final del procedimiento cuando se encuentra con la instrucción **End Sub**:



Por último, introduciremos la instrucción **MsgBox** de forma que el procedimiento quedará de la siguiente manera:



Recordemos que por defecto, si no indicamos lo contrario, al crear un módulo nuevo, se incluye la sentencia **Option Explicit** la cual nos obligará a definir expresamente cada una de las variables que utilizemos ya que de otra forma, el compilador dará un mensaje de error.

### 6.1.1 Ejecución lanzada mediante Ejecutar

Si ejecutamos el procedimiento pulsando **F5** o a través del menú **Ejecutar->Ejecutar Sub/Userform** (▶) contemplaremos el siguiente cuadro de diálogo:



Como cualquier cuadro de diálogo, tendremos que cerrarlo para poder continuar trabajando. En este caso, al no haber indicado nada mas que un texto, nos propone por defecto el botón **Aceptar**. Ver capítulo **Lenguaje de programación** para examinar los detalles de la función **MsgBox** y todas sus posibilidades.

### 6.1.2 Ejecución desde otro procedimiento

Para ejecutar este procedimiento desde otro, podemos utilizar la instrucción **Call** o simplemente, escribir el nombre del procedimiento y ya está:

Sub lanza_Saludo()	Sub lanza_Saludo()
Call Saludo	Saludo
End Sub	End Sub

El resultado será exactamente el mismo que el indicado en el ejemplo anterior.

### 6.1.3 Ejecución desde el Microsoft Excel

Para ejecutar una macro desde Microsoft Excel, podemos situarnos en la ficha de **Programador** y, haciendo clic sobre el icono **Macro**, acceder al cuadro de diálogo **Macro**:



También es posible acceder a este cuadro pulsando la combinación de teclas **ALT + F8**.

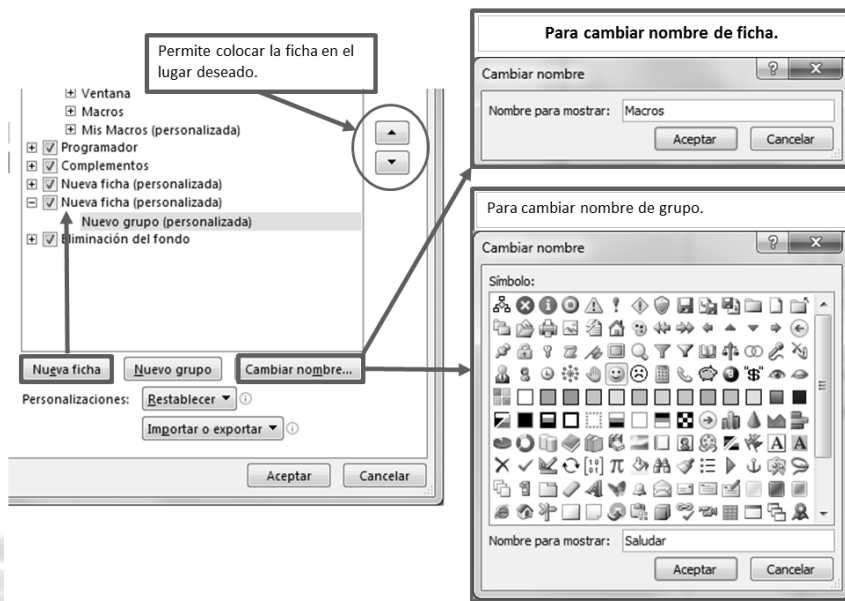
En el mismo, veremos los 2 procedimientos creados hasta el momento. Para ejecutar cualquiera de ellos, basta con seleccionar el que nos interese (por ejemplo **Saludo**) y hacer clic sobre el botón **Ejecutar**. El resultado será la aparición del mensaje de Saludo por pantalla.

Si la macro es de uso frecuente, podemos plantearnos crear una forma de ejecutarla a través de una opción de menú o mediante un icono.

#### 6.1.4 Ejecución desde un icono ubicado en la cinta de opciones

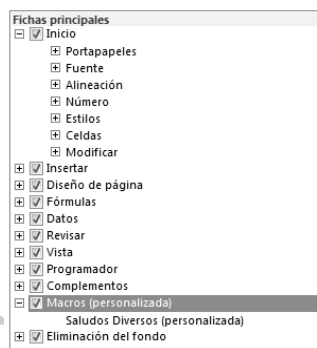
Para ejecutar una macro desde la cinta de opciones, tendremos que acceder al menú **Archivo->Opciones** y a continuación, seleccionar la opción **Personalizar cinta de opciones**.

Podemos crear una nueva Ficha para depositar nuestras macros y para ello, bastará con hacer clic sobre el botón **Nueva ficha** tal y como se muestra en la siguiente imagen:



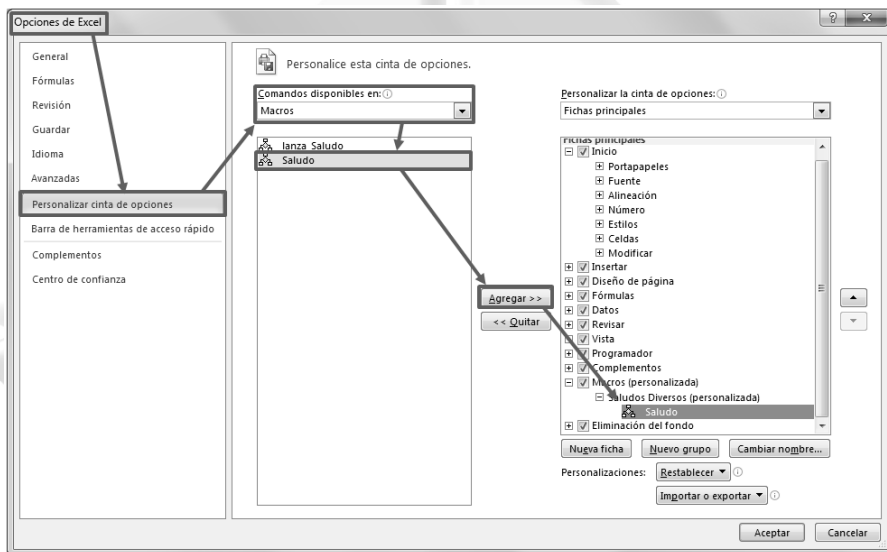
Observaremos que además de una nueva ficha personalizada, se ha creado también un nuevo grupo personalizado donde ubicaremos nuestra macro. Una vez creada la ficha y el grupo, les cambiaremos el nombre seleccionando cada elemento y haciendo clic con el mouse sobre el botón Cambiar nombre... Al cambiar el nombre del grupo podremos elegir también un icono representativo del mismo.

Posteriormente, con los botones situados a la derecha de la ventana, colocaremos la ficha en el lugar que más nos interese. En nuestro caso, la situaremos después del resto de fichas ya existentes. Al final de la inserción y de los cambios propuestos, nuestra nueva ficha y grupo quedará de la siguiente manera:





En este punto, seleccionaremos la categoría **Macros** de la lista desplegable asociada a **Comandos disponibles en:** y observaremos que se nos muestran todas las macros que tenemos disponibles en el libro en curso. Seguidamente, seleccionaremos la ficha Macros y el grupo Saludos Diversos y, haremos clic en **Agregar** para que efectivamente, se añada la opción que ejecutará la macro en la cinta de opciones. Las opciones indicadas se destacan en la siguiente imagen:



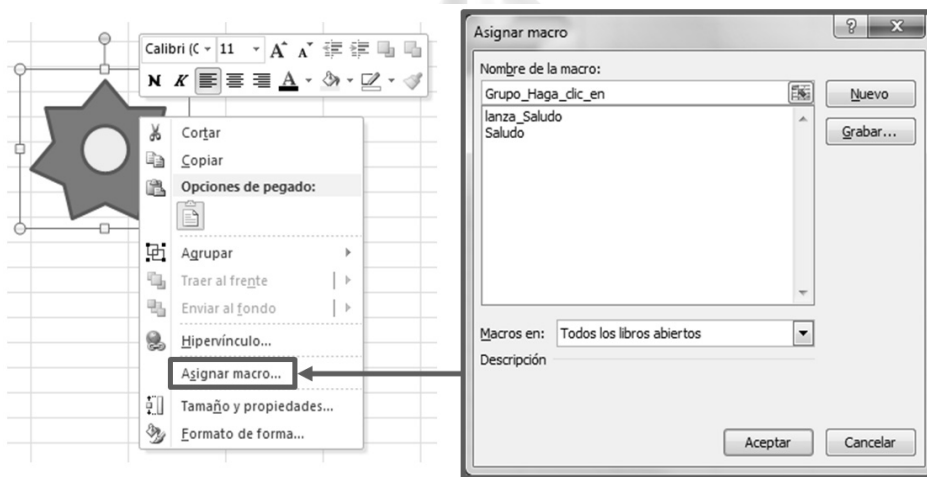
Si hacemos clic en **Aceptar** y regresamos a la pantalla principal de **MS Excel**, observaremos que efectivamente se ha creado una nueva Ficha denominada **Macros** en la que existe un grupo de controles con el nombre de **Saludos Diversos** el cual, contiene un icono con el nombre **Saludo** al que si le hacemos clic, ejecutará la macro **Saludo** mostrando el cuadro de diálogo con el mensaje de **Hola Mundo**. La ficha se verá de la siguiente manera:



### 6.1.5 Ejecución al hacer clic sobre una imagen

Para ejecutar una macro al hacer clic sobre una imagen, basta con seleccionar la imagen deseada y, haciendo clic con el botón derecho del mouse, invocaremos al menú contextual mediante el cual seleccionaremos la opción

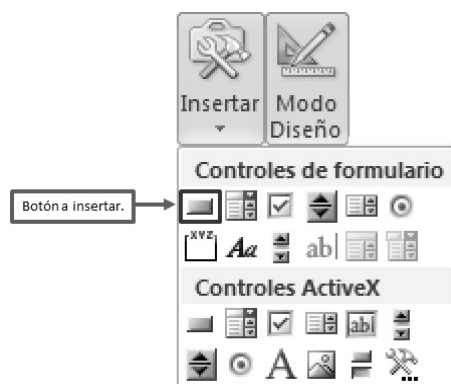
**Asignar macro...** que nos dará acceso al cuadro de diálogo **Asignar macro**. Supongamos que tenemos una imagen cualquiera, y queremos asignar la macro **Saludo**:



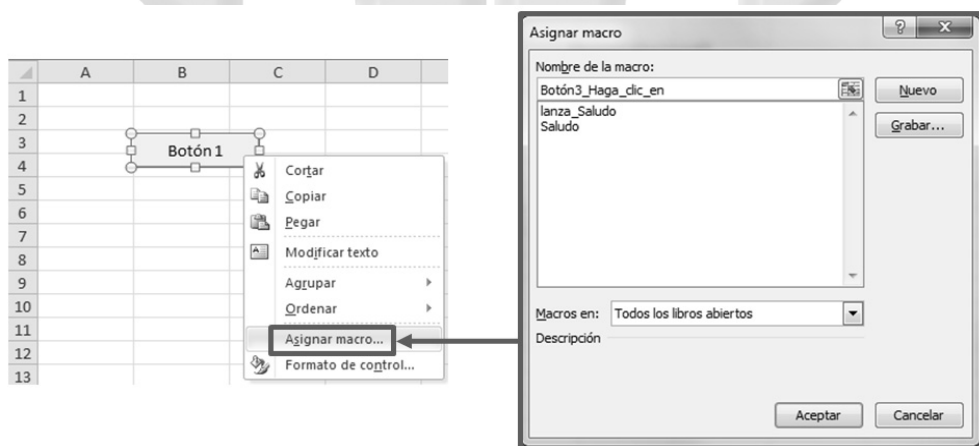
Observaremos que en el cuadro de diálogo **Asignar macro**, se nos ofrece la posibilidad de crear una nueva macro denominada **Grupo\_Haga\_clic\_en** además de ofrecernos las macros disponibles en este momento. Esto significa que si quisieramos crea una nueva macro, podríamos elegir entre pulsar sobre el botón **Nuevo** y, abrir la ventana de código de **VBA** o bien, iniciar la grabación de una macro haciendo clic sobre el botón **Grabar ...** para crearla en este momento. En nuestro caso, simplemente seleccionaremos la macro **Saludo** en la lista de macros y haremos clic sobre el botón **Aceptar**.

### 6.1.6 Ejecución desde un botón (Control de formulario)

De forma similar a lo comentado para la ejecución desde una imagen, podemos también lanzar un procedimiento desde un botón de tipo control de formulario. Para ello, empezaremos por crear un botón de este tipo y seguidamente, le asignaremos una macro que en nuestro caso será, la macro **Saludo**. La creación del botón la realizaremos acudiendo a la ficha **Programador** y seleccionando el icono con la imagen de un botón que tenemos en el grupo de controles de formulario que aparece al hacer clic sobre el icono **Insertar** que se halla en el grupo de controles **Controles**:

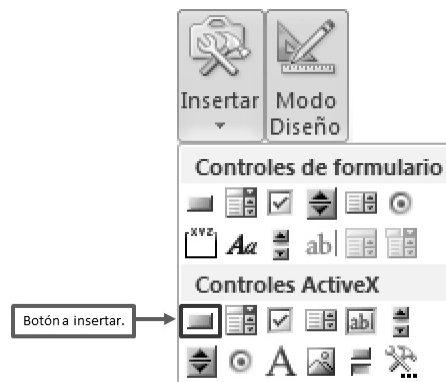


Una vez seleccionado el botón, haremos clic con el botón izquierdo del mouse sobre la zona que contendrá la esquina superior izquierda del botón y, manteniendo pulsado el botón del mouse arrastraremos el cursor hasta la esquina inferior derecha que delimitará la forma y tamaño del botón que estamos creando. Al soltar definitivamente el botón izquierdo del mouse, se abrirá el cuadro de diálogo **Asignar macro** el cual, al igual que lo comentado para la ejecución desde una imagen nos permitirá crear una nueva macro o bien asignar algún de las que ya existan. En nuestro caso, asignaremos la macro **Saludo** y haremos clic en **Aceptar**.

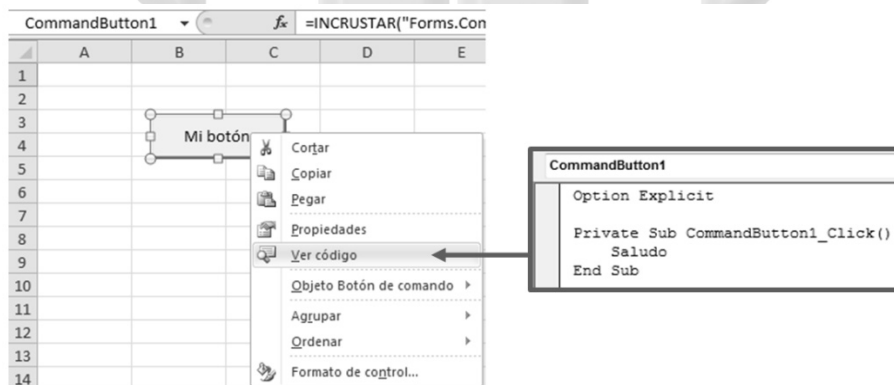


### 6.1.7 Ejecución desde un botón (Active X) <sup>®</sup>

Para la ejecución de una macro desde un botón de tipo Active X, en primer lugar, dibujaremos dicho botón seleccionando el icono de tipo botón que hay en el grupo de Controles de Active X que aparece al hacer clic sobre el icono **Insertar** que se halla en el grupo de controles **Controles**:

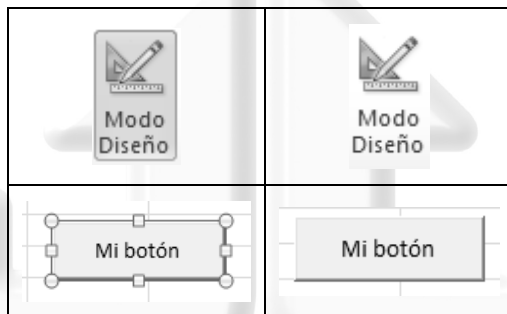


Una vez dibujado el botón, podemos cambiarle el nombre accediendo a sus propiedades a través de la opción **Propiedades** existente en el grupo de controles **Controles** o simplemente haciendo clic con el botón derecho del mouse sobre el botón y seleccionando la opción **Propiedades**. Una vez en la ventana Propiedades, localizaremos la opción Caption y teclearemos el nombre que deseemos que en nuestro caso será Mi botón. Seguidamente, accederemos al editor de código de VBA a través de la opción **Ver código** accesible desde el grupo de controles **Controles** o simplemente haciendo clic con el botón derecho del mouse sobre el botón y seleccionando la opción **Ver código**.



En el código, vemos que de forma automática ya se ha creado el inicio y el fin del procedimiento **on CommandButton1\_Click()** que es el nombre que por defecto se da al procedimiento ligado al evento **Click** del objeto que se está tratando (en nuestro caso **CommandButton1**). Para acabar el procedimiento, simplemente insertaremos la llamada a la macro **Saludo** y ya está. Si regresamos a **MS Excel** y pulsamos sobre el botón, es posible que no se ejecute la macro porque tengamos seleccionado el **Modo Diseño**. Si esto es así, observaremos que el botón aparece con los puntos de dimensionamiento alrededor de todo el contorno, lo cual

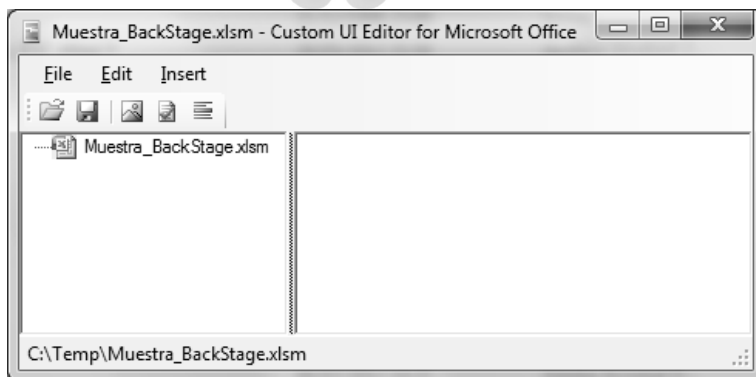
indica que al hacer clic sobre el mismo no se ejecutará ningún evento sino que simplemente se selecciona para acceder a sus propiedades o cambiar la forma. El botón Modo Diseño es bipoisicional. Es decir, funciona como un interruptor de 2 posiciones y cada vez que se pulsa pasa del estado activado al desactivado.



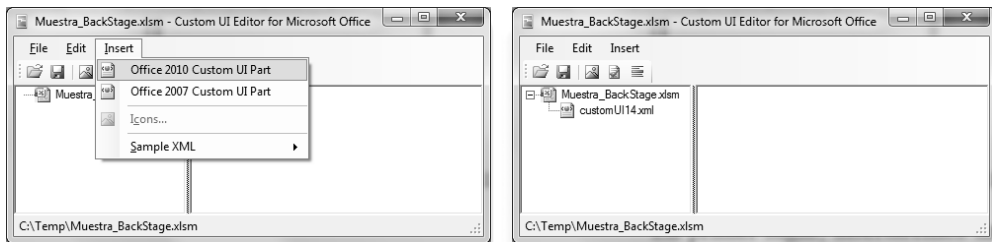
### 6.1.8 Ejecución desde opción de Backstage

En este ejemplo, añadiremos una opción en la vista **Backstage** (vista que se ve al hacer clic e la pestaña **Archivo**) utilizando la aplicación **Custom UI Editor for Microsoft Office**. Se trata de una aplicación que se puede descargar de forma gratuita y mediante a cual se pueden añadir y configurar diversos aspectos relacionados con las fichas y grupos de controles de **MS Excel**.

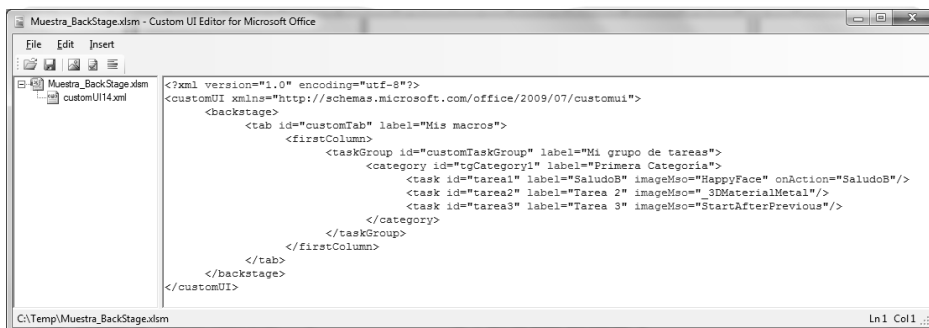
Una vez descargada e instalada la aplicación, la ejecutaremos accediendo a una ventana a través de la cual abriremos al libro al que le9 queremos añadir la nueva opción (en el ejemplo: **Muestra\_BackStage.xlsm**). Para abrir el libro, acudiremos a la opción **File->Open** y, mediante el cuadro de diálogo **Abrir**, localizaremos nuestro libro y lo abriremos:



En primer lugar, insertaremos un **Office 2010 Custom UI Part** para poder definir el tipo de elemento que queremos configurar mediante la siguiente opción:

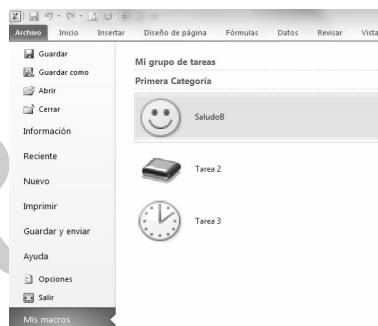


En este punto, insertaremos un ejemplo de **XML** denominado **Custom Outspace** para poder modificarlo posteriormente e incluir la llamada a nuestro procedimiento. Dicha inserción la realizaremos mediante **Insert->Sample XML->Custom Outspace**. Después de adaptarlo un poco para nuestro ejemplo, el código se verá de la siguiente manera:



Una vez modificado, lo guardaremos y cerraremos la referencia al libro.

Para comprobar el resultado, iremos a **MS Excel** y abriremos el libro pudiendo comprobar que, al acceder a la opción **Archivo**, vemos lo siguiente:




Efectivamente, si hacemos clic sobre el icono sonriente con la etiqueta SaludoB, veremos como aparece el cuadro de diálogo con el texto **Hola Mundo!!**.

Para que este ejemplo funcione bien, hemos de incluir un parámetro en el procedimiento tal y como mostramos a continuación:

```
Sub SaludoB(control As IRibbonControl)
    MsgBox ("Hola Mundo!!")
End Sub
```

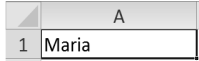

### 6.1.9 Saludo con parámetro

Podríamos hacer que el saludo fuera variable y dependiente de un parámetro que proporcionaríamos al procedimiento durante la llamada al mismo. Así pues, con una pequeña variación, podríamos realizar la siguiente prueba:

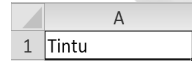

Código	Resultado
<pre>Sub lanza_Saludo_A1()     Call Saludo_A("Juanto") End Sub  Sub Saludo_A(n As String)     MsgBox "Hola " &amp; n End Sub</pre>	

En este caso, el nombre a utilizar para el saludo, se ha pasado indicando expresamente una cadena de tipo **String**. Podíamos haber utilizado el contenido de una celda de forma que, al variar ésta, el mensaje también variará. Por ejemplo, si colocamos en la celda **A1** un contenido y ejecutamos el procedimiento, observaremos que el mensaje utiliza dicho contenido para componer la cadena de texto a presentar en el saludo:

Ra-Ma®

Contenido celda	Código	Resultado
	<pre>Sub lanza_Saludo_A2()     Dim n As String     n = ActiveSheet.Range("A1").Value     Call Saludo_A(n) End Sub</pre>	

En este caso, hemos utilizado el objeto Range para hacer referencia a una celda y obtener su valor a través de la propiedad Value. También podemos utilizar el objeto Cells el cual, nos permite referenciar a una celda a través de sus propiedades columna y fila. El siguiente ejemplo proporciona un resultado idéntico al anterior pero, utilizando el objeto Cells:

Contenido celda	Código	Resultado
	<pre>Sub lanza_Saludo_A3()     Dim n As String     n = ActiveSheet.Cells(1, 1).Value     Call Saludo_A(n) End Sub</pre>	

## 6.2 EVENTOS Y CARACTERÍSTICAS DE LIBROS, HOJAS Y RANGOS

### 6.2.1 Al abrir y cerrar un libro

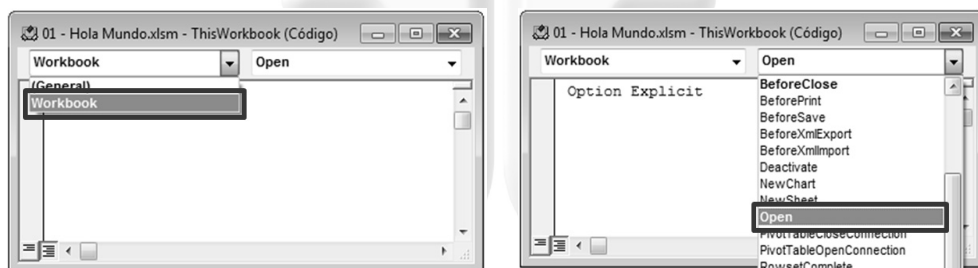
En ocasiones, es interesante poder realizar alguna acción cuando abrimos o cerramos un libro. Para ello, hemos de asociar código a los eventos Open y BeforeClose del objeto Workbook.

Para asociar código a los eventos asociados a un libro, debemos seleccionar el objeto en el Explorador de proyectos y hacer a continuación, seleccionar la opción Ver código (por ejemplo, pulsando F7 o simplemente haciendo doble clic sobre el objeto en cuestión). En nuestro caso, el objeto que representa al libro en curso se llama ThisWorkbook ya que es el nombre que MS Excel da al libro por defecto.



Una vez tengamos la ventana de código asociada a ThisWorkbook, observaremos que en la parte superior de la ventana de código existen 2 listas desplegables mediante las cuales, podemos escoger el objeto sobre el que queremos trabajar y, una vez seleccionado, seleccionar el evento deseado. Ver capítulo Entorno de desarrollo para obtener más detalles sobre la ventana de código.

En nuestro caso, seleccionaremos el objeto **Workbook** y a continuación, el evento **Open**:



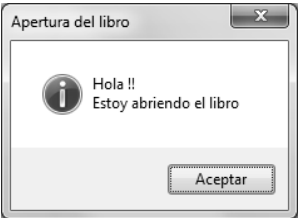
Tras seleccionar el evento **Open**, veremos que en la zona destinada a la introducción de código se ha creado un procedimiento con el principio y fin del mismo cuyo nombre es un nombre compuesto del objeto y evento que acabamos de seleccionar:

```
Private Sub Workbook_Open()  
End Sub
```

A continuación, sólo hemos de escribir las instrucciones que queremos ejecutar cada vez que un libro se abra y ya está. En nuestro ejemplo, lo que queremos es que al abrir el libro nos salude y que deposite en la celda A1 la hora y fecha de apertura. El código asociado al evento será:

### Código

```
Private Sub Workbook_Open()  
  
    ' Mensaje indicando apertura del libro  
    MsgBox Prompt:="Hola !!" & vbCrLf & "Estoy abriendo el libro", _  
        Buttons:=vbInformation, _  
        Title:="Apertura del libro"  
    If ActiveSheet.Name <> "Propiedades_Libro" Then  
        ' Coloca hora y fecha de apertura en la celda A1  
        ActiveSheet.Cells(1, 1) = "Libro abierto a las " & Time & "  
del " & Date  
    End If  
End Sub
```

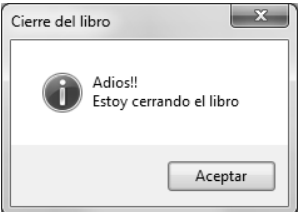
Resultado												
		<table border="1"> <thead> <tr> <th></th> <th>A</th> <th>B</th> <th>C</th> </tr> </thead> <tbody> <tr> <td>1</td> <td colspan="3">Libro abierto a las 9:41:04 del 11/08/2012</td> </tr> </tbody> </table>				A	B	C	1	Libro abierto a las 9:41:04 del 11/08/2012		
	A	B	C									
1	Libro abierto a las 9:41:04 del 11/08/2012											

Observamos que en esta ocasión, para inicializar los parámetros asociados a la función **MsgBox**, hemos utilizado la notación **parámetro:=** la cual, nos permite inicializar dichos parámetros en cualquier orden. Vemos también que hemos introducido un retorno de carro y un salto de línea en el mensaje utilizando la constante **vbCrLf**.

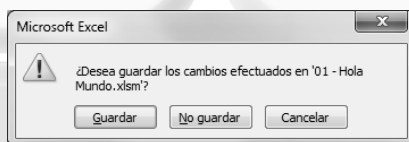
De igual forma, para escribir el código asociado al evento **BeforeClose**, seleccionaremos el evento desde la lista de eventos asociada al objeto e introduciremos su correspondiente código.

Código
<pre>Private Sub Workbook_BeforeClose(Cancel As Boolean)      ' Mensaje indicando cierre del libro     MsgBox Prompt:="Adios!! " &amp; vbCrLf &amp; "Estoy cerrando el libro",     _     Buttons:=vbInformation, _     Title:="Cierre del libro"      ' Coloca hora y fecha de apertura en la celda A2     ActiveSheet.Cells(2, 1) = "Libro cerrado a las " &amp; Time &amp; " del " &amp; Date      ThisWorkbook.Saved = True  End Sub</pre>

Ra-Ma®

Resultado																			
		<table border="1"> <thead> <tr> <th></th><th>A</th><th>B</th><th>C</th><th></th></tr> </thead> <tbody> <tr> <td>1</td><td colspan="4">Libro abierto a las 9:41:04 del 11/08/2012</td></tr> <tr> <td>2</td><td colspan="4">Libro cerrado a las 9:55:09 del 11/08/2012</td></tr> </tbody> </table>				A	B	C		1	Libro abierto a las 9:41:04 del 11/08/2012				2	Libro cerrado a las 9:55:09 del 11/08/2012			
	A	B	C																
1	Libro abierto a las 9:41:04 del 11/08/2012																		
2	Libro cerrado a las 9:55:09 del 11/08/2012																		

Al tratar de cerrar un libro que posee cambios pendientes de guardar desde la última vez que lo abrimos, encontraremos el siguiente mensaje en el que se nos invita a guardar o no los cambios realizados o a cancelar la acción de cierre del libro para poder continuar trabajando:



Para evitar esta pregunta tenemos 2 opciones:

- Salvar el libro de forma ‘silenciosa’ (método **Save**).
- Indicar al libro que ya está salvado (aunque no es cierto) modificando su propiedad **Saved**.

El código asociado a cada posible solución es el siguiente:

<b>Cerrar salvando el libro</b>	<pre>Private Sub Workbook_BeforeClose(Cancel As Boolean)      ' Mensaje indicando cierre del libro     MsgBox Prompt:="Adios!! " &amp; vbCrLf &amp; _     "Estoy cerrando el libro", _     Buttons:=vbInformation, _     Title:="Cierre del libro"      ' Coloca hora y fecha de apertura en la celda A2     ActiveSheet.Cells(2, 1) = "Libro cerrado a las " &amp;     Time &amp; " del " &amp; Date      <b>ThisWorkbook.Save</b>  End Sub</pre>

### Cerrar sin salvar el libro

```
Private Sub Workbook_BeforeClose(Cancel As Boolean)

    ' Mensaje indicando cierre del libro
    MsgBox Prompt:="Adios!! " & vbCrLf & _
    "Estoy cerrando el libro", _
    Buttons:=vbInformation, _
    Title:="Cierre del libro"

    ' Coloca hora y fecha de apertura en la celda A2
    ActiveSheet.Cells(2, 1) = "Libro cerrado a las " & _
    _
    Time & " del " & Date

    ThisWorkbook.Saved = True

End Sub
```

## 6.2.2 Al activar una hoja

Para ejecutar un código cada vez que accedemos a una hoja, podemos utilizar los eventos **SheetActivate** del objeto **Workbook** o bien, el método **Activate** de cada una de las hojas sobre las que nos interese actuar.

Si escribimos código a nivel de libro (**Workbook**), sólo hemos de escribir dicho código una vez y se ejecutará por igual cada vez que accedamos a cualquiera de las hojas del libro. Así pues, esta elección será ideal cuando queramos ejecutar un código genérico para todas las hojas.

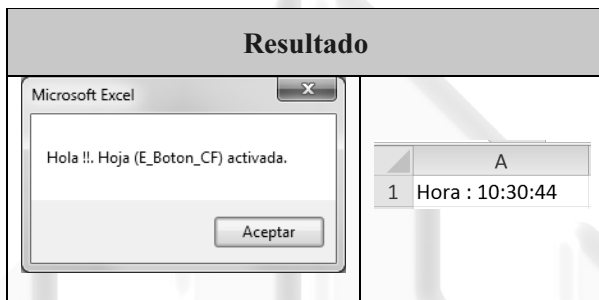
Supongamos que cada vez que accedemos a una hoja, queremos saludar y mostrar en la celda A1 la hora en la que se accede por última vez. Para realizar esto, seleccionaremos el objeto **ThisWorkbook** y buscaremos el evento **SheetActivate** para escribir el siguiente código:

### Código

```
Private Sub Workbook_SheetActivate(ByVal Sh As Object)
    MsgBox ("Hola !!. Hoja (" & Sh.Name & ") activada.")
    Cells(1, 1) = "Hora : " & Time
End Sub
```

Ra-Ma®

A partir de este momento, cada vez que vayamos accediendo a una hoja, nos iremos encontrando con un saludo indicando el nombre de la hoja y veremos que en la celda **A1** se introduce la hora en la que se accede:



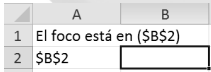
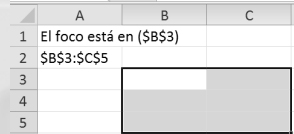
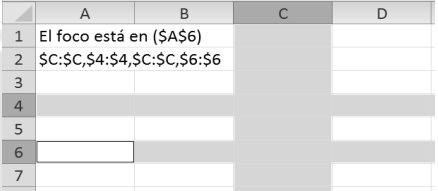
Si queremos que las acciones se realicen sólo en una determinada hoja, tendremos que codificar el evento **Activate** en dicha hoja para que solo se ejecute cuando dicha hoja sea activada. En el siguiente ejemplo, se muestra un posible código asociado al evento **Activate** de la hoja **EJ\_Imagen**:

Código
<pre>Private Sub Worksheet_Activate()     MsgBox ("Hola !!. " &amp; vbCrLf &amp; "(" &amp; ActiveSheet.Name &amp; "). " _         &amp; vbCrLf &amp; "He sido activada y poseo código específico.") End Sub</pre>
Resultado
<p>The image shows a screenshot of a Microsoft Excel message box. The text inside the box reads: 'Hola !!. (EJ_Imagen). He sido activada y poseo código específico.' There is an 'Aceptar' button at the bottom right of the message box.</p>

En el caso de que en una hoja coincidan ambos eventos (el producido por el evento **SheetActivate** de su libro y el evento **Activate** de la propia hoja), se ejecutará en primer lugar el evento de la hoja y seguidamente el del libro.

### 6.2.3 Al cambiar la selección de un rango

En este ejemplo, podamos analizar cual es el rango que se está seleccionando analizando el objeto **Selection** el cual, representa el rango que está seleccionado en todo momento. Para ello, dentro de la hoja sobre la que desea trabajar, insertaremos para el evento **SelectionChange** el siguiente ejemplo:

Código	
<pre>Private Sub Worksheet_SelectionChange(ByVal Target As Range)     Dim r As Range     Set r = Selection     Range("A1").Value = "El foco está en (" &amp; ActiveCell.Address &amp;     ") "     Range("A2").Value = r.Address End Sub</pre>	
Resultado	
	
	

Podemos observar que hemos definido una variable de tipo **Range** sobre la que depositar el contenido del objeto **Selection**.

En el último caso, se muestra el resultado de una selección múltiple (**filas 4 y 6 y columna C**).

### 6.2.4 Análisis de algunas propiedades de libros, hojas y rangos

En este ejemplo, se muestran algunas de las propiedades de los objetos principales del modelo de datos, descritas en el capítulo **El modelo de objetos**.

A continuación se muestra el código del procedimiento utilizado en el ejemplo:

### Código

```
Sub propiedades_Libro()  
    Dim w As Workbook  
    Set w = ThisWorkbook  
    Range("B3").Value = w.Name  
    Range("B4").Value = w.CodeName  
    Range("B5").Value = w.FullName  
    Range("B6").Value = w.Path  
    Range("B7").Value = w.HasPassword  
    Range("B8").Value = w.Worksheets.Count  
    Range("B9").Value = w.Worksheets(1).Name  
    Range("B10").Value = w.Worksheets(Worksheets.Count).Name  
    Dim ls As String  
    Dim h As Worksheet  
    For Each h In w.Worksheets  
        If ls <> "" Then  
            ls = ls & ", "  
        End If  
        ls = ls & h.Name  
    Next h  
    Range("B11").Select  
    With Range("B11").Validation  
        .Delete  
        .Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop,  
Operator:= _  
        xlBetween, Formula1:=ls  
    End With  
    Range("B11").Value = w.Worksheets(1).Name  
    Range("B12").Value = w.ActiveSheet.Name  
    Range("B13").Value = w.ActiveSheet.Comments.Count  
    Dim t As String  
    t = w.ActiveSheet.Type  
    Range("B14").FormulaR1C1 = "=VLOOKUP(" & t & ",r12c3:r16c4,2,0)"  
    Range("B15").Value =  
WorksheetFunction.Choose(WorksheetFunction.Weekday(Now), _  
    "Domingo", "Lunes", "Martes", "Miércoles", "Jueves", "Viernes",  
    "Sábado")  
    Range("B16").Value = WorksheetFunction.RandBetween(1, 1000)  
End Sub
```

Ra-Ma®

Resultado						
	A	B	C	D	E	
1						
2	Propiedades					
3	Nombre del libro	01 - Hola Mundo.xlsm				
4	Nombre de código	ThisWorkbook				
5	Nombre completo	C:\Cursos\Curso_Macros\Capitulos\14 - Ejemplos\01 - Hola Mundo.xlsm				
6	Path	C:\Cursos\Curso_Macros\Capitulos\14 - Ejemplos				
7	Tiene password?	FALSO				
8	Número de hojas	8				
9	Primer hoja	EJ_Imagen				
10	Ultima hoja	Hoja2				
11	Hojas en el libro	EJ_Imagen				
12	Hoja activa	Propiedades_Libro	-4109	xlChart		
13	Número de comentarios	2	-4116	xlDialogSheet		
14	Tipo de hoja	xlWorksheet	4	xlExcel4IntlMacroSheet		
15	Día de la semana	Martes	3	xlExcel4MacroSheet		
16	Aleatorio	536	-4167	xlWorksheet		

En este código puede observarse que hay un primer grupo de propiedades relativas al objeto **Workbook** como son:

- el nombre,
- el nombre interno,
- el nombre completo incluyendo la ruta,
- la ruta en la que se halla el libro,
- si tiene palabra de paso o no,
- cuantas hojas tiene,
- el nombre de la primera hoja,
- el nombre de la última hoja,
- una validación que contiene todas las hojas existentes en el libro,
- el nombre de la hoja activa,
- el número de comentarios existentes en la hoja activa,
- el tipo de hoja (muestra el nombre y se apoya en la fórmula VLOOKUP)
- un par de ejemplos de cómo como usar una función en VBA (Choose, RandBetween).

Para la actualización de los valores se ha colocado un botón con el rótulo Propiedades de forma que, cada vez que se hace clic sobre el mismo, se ejecuta el procedimiento que estamos describiendo y que se denomina **propiedades\_Libro**.

Tal y como hemos visto en capítulos anteriores, cada objeto posee decenas de propiedades que pueden ser consultadas o modificadas en algunos casos por nuestros métodos **VBA**.

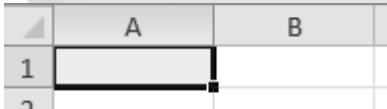



## 6.3 EJEMPLOS CON COLORES

### 6.3.1 Obtener un color

Para obtener un color, bastará con acceder al valor de la propiedad **ColorIndex** del objeto **Interior** que representa el interior de una celda.

Así pues, si aplicamos un color de relleno amarillo sobre la celda **A1** y queremos averiguar qué valor representa dicho color, podemos utilizar el siguiente código:

Código	
<pre>Sub muestra_color()     Range("B1").Value = Range("A1").Interior.ColorIndex End Sub</pre>	
Resultado	
	

En este ejemplo, vemos como obtenemos el color de relleno de la celda **A1** a partir del objeto **Interior** y de su propiedad **ColorIndex** y depositamos su valor en la celda **B1**.

Podríamos disponer de una función de usuario a la que pudiéramos pasar como parámetro la celda de la que queremos averiguar su color y, obtener como resultado de la misma el valor correspondiente a su color.

Si modificamos un poco el ejemplo anterior y, creamos una función denominada **obtener\_color**, podemos obtener el mismo resultado con el siguiente código:

Código
<pre>Sub muestra_color()     Range("B1").Value = obtener_color(Range("A1")) End Sub  Function obtener_color(celda As Range)     obtener_color = celda.Interior.ColorIndex End Function</pre>

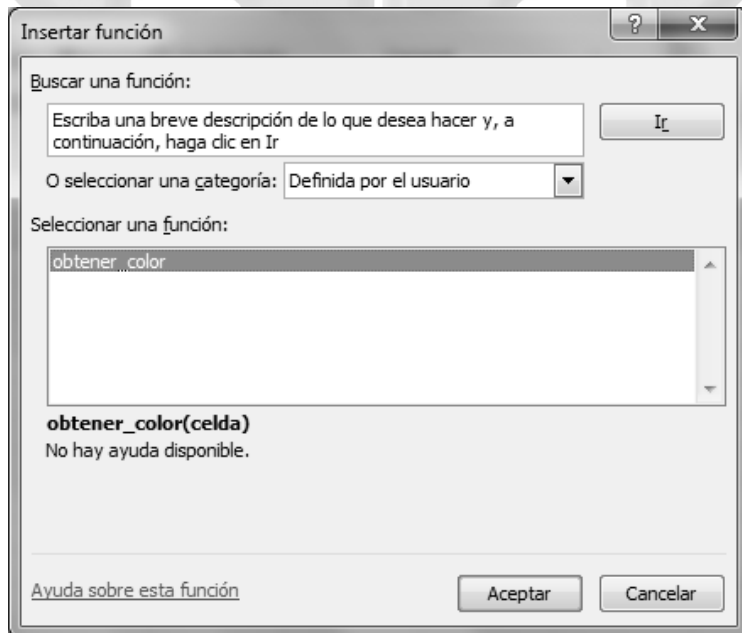
La ventaja de utilizar una función es que podríamos llegar a utilizarla desde el entorno de **MS Excel** como una función de usuario ampliando así la funcionalidad en nuestros libros.

Por ejemplo, si coloreamos la celda **A2** de color azul y, en la celda **B2** utilizamos la función **obtener\_color** para hallar el valor correspondiente al color, veremos como podemos utilizar una función creada mediante **VBA** como si se tratara de una función mas de **MS Excel**.

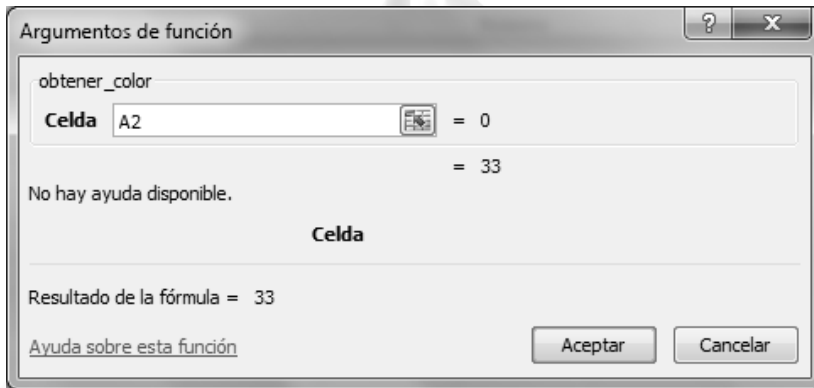
Ejemplo			
	A	B	C
1		6	
2		=obtener_color(A2)	

	A	B
1		6
2		33

Si deseamos utilizar el asistente de funciones para insertar la función, tendremos que localizar la función dentro de la categoría de funciones **Definida por el usuario**:



A partir de aquí, observaremos como el asistente funciona exactamente igual que para una función predefinida solicitando los argumentos necesarios y prevvisualizando el resultado antes de cerrar el cuadro de diálogo:



### 6.3.2 Mostrar una celda coloreada

Continuando con la propiedad **ColorIndex** del objeto **Interior**, veremos que para colorear un determinado rango basta con cambiar el valor de esta propiedad.

Vamos a crear un procedimiento que nos permita colorear un rango de forma que dicho procedimiento pueda ser reutilizado tantas veces como sea necesario. El procedimiento contendrá 2 parámetros: el **rango** a colorear y el **color** a utilizar. El código será el siguiente:

#### Código

```
Sub pon_color(rango As Range, color As Integer)
    rango.Interior.ColorIndex = color
End Sub
```

Para poner en color verde la celda **A1**, podemos escribir el siguiente código:

#### Código

```
Sub pon_colores1()
    Dim celda As Range
    Set celda = Range("A1")
    Call pon_color(celda, 4)
End Sub
```

### Resultado

	A
1	

## 6.3.3 Colorear aleatoriamente un rango

En el siguiente ejemplo, vamos a colorear un rango de celdas aleatoriamente al hacer clic sobre un botón. Además, haremos una pequeña pausa entre coloreado y coloreado para que podamos dar una cierta sensación de dinamismo.

Para ello, necesitaremos crear un método que sea capaz de esperar unos segundos determinados y otro, que calcule un color aleatoriamente y lo aplique un rango concreto.

### Código

```
Sub pon_colores_aleatoriamente()
    Dim i, j As Integer
    For j = 1 To 4
        For i = 2 To 11
            Call pon_color(Cells(i, 1), Int(16 * Rnd()))
        Next i
        espera (1)
    Next j
End Sub
```

### Resultado

	A	B	C
1	COLOR		
2			
3			Colorea
4			
5			
6			
7			
8			
9			
10			
11			

En este procedimiento, observamos que se colorean aleatoriamente las 10 celdas correspondientes al rango **A2:A11**. Vemos también que, una vez que se colorean las celdas, se llama a la función **espera** para esperar un segundo antes de volver a colorear de nuevo dichas celdas. Esta acción se realiza 4 veces gracias al bucle **For j = 1 To 4**.

La función **espera** es la siguiente:

Código
<pre>Sub espera(segundos As Integer)     Dim w As Date     w = DateAdd("s", segundos, Now())     Application.Wait w End Sub</pre>

Podemos observar en la misma que, usamos la función **DateAdd** para sumar un determinado número de segundos a la hora actual. Tales segundos son pasados como parámetro a la función y en nuestro caso es **1**.

Por último, al botón que hemos insertado en la hoja, le asociaremos la macro **pon\_colores\_aleatoriamente** y, al hacer clic sobre el mismo, observaremos que las celdas del rango **A1:A10** se colorean aleatoriamente 4 veces esperando un segundo entre coloreado y coloreado.

### 6.3.4 Ordenar por colores

Para ordenar por color, podemos crear utilizar la función creada anteriormente denominada **obtener\_color**.

La idea sería la de apoyarnos en una columna auxiliar que contenga el valor correspondiente al color de cada una de las celdas que queremos ordenar y, posteriormente ordenemos por dicha columna.

Supongamos que disponemos de una hoja en la que hemos incluido una serie de anotaciones y posteriormente, hemos rellenado con un determinado fondo que para nosotros tiene un determinado significado. Si las notas las hemos colocado en la columna **A**, podríamos utilizar la columna **B** para obtener el color incluyendo en cada celda de dicha columna **B** la función **obtener\_color** y obtener el color de la celda de la columna **A** ubicada en la misma fila tal y como se muestra a continuación:

Ejemplo					
A			B		
1	NOTAS		COLOR		
2	Nota 1		=obtener_color(A2)		
3	Nota 10		=obtener_color(A3)		
4	Nota 2		=obtener_color(A4)		
5	Nota 3		=obtener_color(A5)		
6	Nota 4		=obtener_color(A6)		
7	Nota 5		=obtener_color(A7)		
8	Nota 6		=obtener_color(A8)		
9	Nota 7		=obtener_color(A9)		
10	Nota 8		=obtener_color(A10)		
11	Nota 9		=obtener_color(A11)		

A			B		
1	NOTAS		COLOR		
2	Nota 1		6		
3	Nota 10		3		
4	Nota 2		4		
5	Nota 3		4		
6	Nota 4		6		
7	Nota 5		3		
8	Nota 6		6		
9	Nota 7		4		
10	Nota 8		3		
11	Nota 9		6		

A			B		
1	NOTAS		COLOR		
2	Nota 10		3		
3	Nota 5		3		
4	Nota 8		3		
5	Nota 2		4		
6	Nota 3		4		
7	Nota 7		4		
8	Nota 1		6		
9	Nota 4		6		
10	Nota 6		6		
11	Nota 9		6		

6.3.5 Ordenar por colores a través de un procedimiento

Para ordenar el rango del ejemplo anterior a través de un procedimiento podríamos utilizar el siguiente código:

## Código

```
Sub ordenar()  
    Dim old As Variant  
    old = Application.Calculation  
    Application.Calculation = xlCalculationManual  
  
    Dim r As Range  
    Set r = Range("A1:B11")  
    r.Sort Key1:="COLOR", order1:=xlAscending, Header:=xlYes  
  
    Application.Calculation = old  
End Sub
```

## Resultado

	A	B
1	<b>NOTAS</b>	<b>COLOR</b>
2	Nota 1	6
3	Nota 10	3
4	Nota 2	4
5	Nota 3	4
6	Nota 4	6
7	Nota 5	3
8	Nota 6	6
9	Nota 7	4
10	Nota 8	3
11	Nota 9	6

	A	B
1	<b>NOTAS</b>	<b>COLOR</b>
2	Nota 10	3
3	Nota 5	3
4	Nota 8	3
5	Nota 2	4
6	Nota 3	4
7	Nota 7	4
8	Nota 1	6
9	Nota 4	6
10	Nota 6	6
11	Nota 9	6

Podemos observar que modificamos la propiedad **Calculation** a nivel de aplicación para evitar que se produzca ningún tipo de recálculo mientras ordenamos el rango ya que esto puede provocar algún error. Para ello, lo primero que hacemos es almacenar el valor actual de dicha propiedad para poder dejarlo como estaba al final de la ejecución del procedimiento.

## 6.4 'LOGS' AL ABRIR Y CERRAR LIBRO

En ocasiones, puede ser interesante fabricar un registro de aperturas y cierres de un libro para saber cuando y quien fue la última vez que accedió al mismo. Este tipo de registro, es lo que denominamos 'log' (término anglosajón equivalente a la palabra bitácora).

Para poder desarrollar esta funcionalidad, utilizaremos los eventos Open y Deactivate del libro en cuestión.

La idea es crear un fichero de texto que almacene el nombre de usuario, la fecha y la hora cada vez que se abra un libro o se cierre. El nombre que recibirá el fichero de texto será el mismo que posee el libro pero, con la extensión 'txt'. La ubicación de dicho fichero será la misma en la que se halle el libro.

Para realizar este ejemplo, simplemente insertaremos un módulo para escribir el método **graba\_log** el cual será llamado desde los eventos de apertura y cierre asociados al objeto **ThisWorkbook**. El código asociado a dichos eventos es el siguiente:

```
Private Sub Workbook_Deactivate()  
    graba_log("Close")  
End Sub  
  
Private Sub Workbook_Open()  
    graba_log("Open")  
End Sub
```

En el modulo, insertaremos el siguiente código:

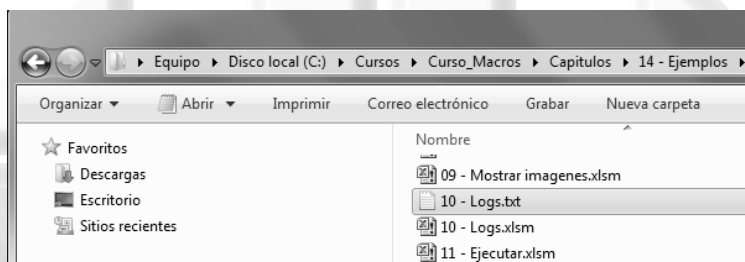
```
Sub graba_log(accion As String)  
    Dim fichero, extension As String, partes As Object  
    partes = Split(ThisWorkbook.Name, ".")  
    extension = partes(UBound(partes))  
    fichero = ThisWorkbook.path & "\" & Replace(ThisWorkbook.Name,  
extension, "txt")  
    Dim linea As String  
    If accion = "Open" Then  
        linea = "-> Abre el usuario " & Environ("username") & " a
```

```

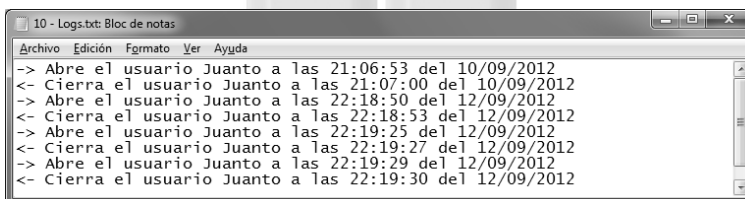
las " & Time & " del " & Date
Else
    linea = "<- Cierra el usuario " & Environ("username") & " a
las " & Time & " del " & Date
End If
Open fichero For Append As #1
Print #1, linea
Close #1
End Sub

```

A partir de aquí, si abrimos y cerramos varias veces el libro y luego vamos a consultar el fichero de texto que se ha creado, podremos comprobar que efectivamente se halla en la misma carpeta que el libro:



También podremos comprobar que el fichero de texto se ha creado con una información parecida a la siguiente:



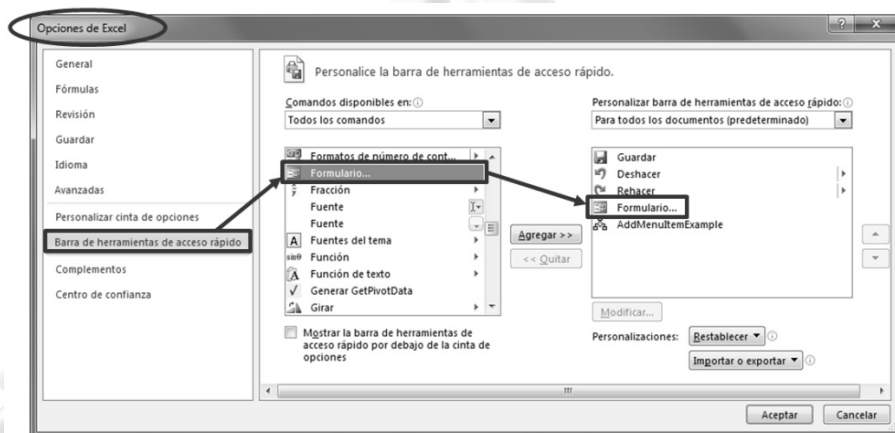
## 6.5 FORMULARIOS

### 6.5.1 Utilidad 'Formulario'

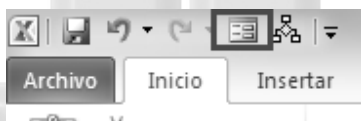
Una forma sencilla de añadir, modificar o eliminar datos en una tabla, es utilizar el comando o utilidad **Formulario**. A pesar de que no es necesario escribir ningún tipo de código para su utilización, es una utilidad muy interesante que no hemos querido dejar de mencionar y que nos permite introducir el concepto de formulario que posteriormente ampliaremos mediante otros ejemplos.



Esta opción puede añadirse a cualquier ficha o a la barra de herramientas de acceso rápido a través del cuadro de diálogo **Opciones de Excel** accesible desde la ficha **Archivo->Opciones** seleccionando **Todos los comandos** y localizando el comando **Formulario** tal y como se muestra a continuación:



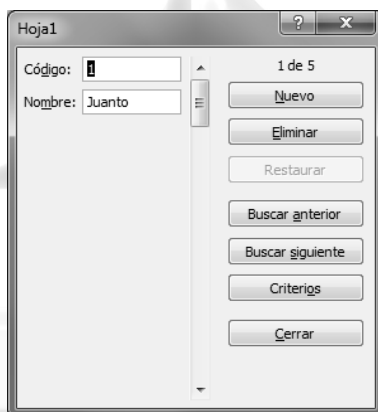
Al aceptar, observaremos que en la barra de herramientas de inicio rápido aparece un nuevo icono representando al comando **Formulario**:



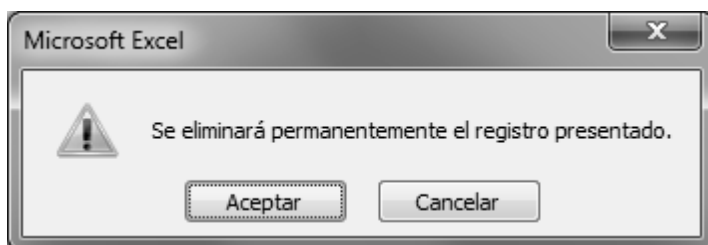
A partir de este momento, ya podríamos utilizar este comando seleccionando la tabla que queremos mantener o simplemente haciendo clic sobre alguna de las celdas incluidas en dicha tabla. Supongamos que tenemos la siguiente tabla:

	A	B
1	Código	Nombre
2		1 Juanto
3		2 Paco
4		3 Sonia
5		4 Xavi
6		5 Jordi

Si hacemos clic sobre la celda **A1** y a continuación, hacemos clic sobre el icono que representa el comando **Formulario**, observaremos que aparece la siguiente ventana:



En la misma comprobaremos que existe un cuadro de texto por cada columna cuyos r tulos se corresponden con las cabeceras de las columnas de la tabla (C digo y Nombre). Asimismo, observaremos que existe una barra de desplazamiento mediante la cual podemos pasar de una fila a otra. Si la tabla es muy grande, es posible que necesitemos realizar alguna b squeda para localizar la fila que contiene la informaci n buscada. Para ello, pulsari mos sobre el bot n criterios y a continuaci n introduciri mos la part cula de b squeda en el campo sobre el que buscaremos. Por  ltimo, haremos clic sobre el bot n que indica el sentido de la b squeda (**Buscar anterior** o **Buscar siguiente**) para navegar hacia la parte superior o inferior de la tabla. Si deseamos crear una nueva fila, pulsaremos sobre el bot n **Nuevo**. Si deseamos eliminar una fila (o registro) pulsaremos sobre el bot n **Eliminar** recibiendo el siguiente mensaje de confirmaci n antes de eliminarlo definitivamente:



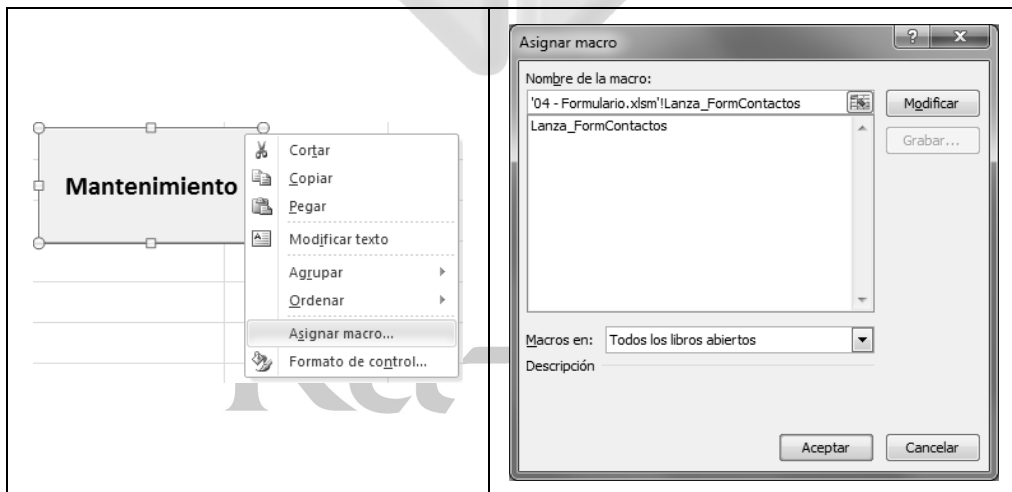
## 6.5.2 Mantenimiento

A continuación vamos a desarrollar un formulario con una funcionalidad similar a la descrita para el comando Formulario pero, aprovechando la versatilidad de VBA e incorporando alguna prestación que nos permita explicar algunas de las instrucciones del lenguaje.

Supongamos que tenemos una tabla muy simple en la que queremos almacenar los teléfonos y los nombres de ciertas personas. A continuación mostramos dicha tabla y también un botón de tipo **Control de formulario** que hemos incluido simplemente para poder invocar al mantenimiento de la tabla. Dicho botón posee el rótulo **Mantenimiento**:

	A	B	C
1	Teléfono	Nombre	
2	913333333	Juan Antonio	
3	915555555	Sebi	
4	932222222	Jose Antonio	
5	938888888	Vito	
6	942555555	Juan Luis	
7	963333333	Cuatro	
8	967777777	Antonio Pérez	

Haciendo clic con el botón derecho del mouse sobre el botón, podremos invocar al menú contextual mediante el cual acceder a la opción **Asignar macro...** y, una vez en el cuadro de diálogo **Asignar macro**, asignarle la macro que abrirá el formulario cuando éste no esté abierto:



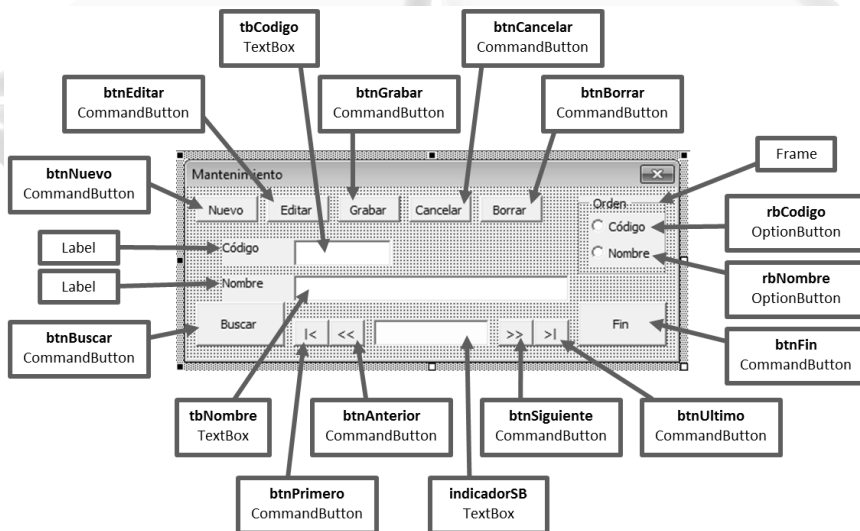
El código asociado a la macro **Lanza\_FrmContactos** es el siguiente:

```
Sub Lanza_FormContactos()  
    frmContactos.Show()  
End Sub
```

El método **Show** permite mostrar el objeto formulario que describiremos a continuación.

### 6.5.2.1 ELEMENTOS DEL FORMULARIO

En primer lugar, mostramos el formulario principal denominado **frmContactos** y sus diferentes elementos:

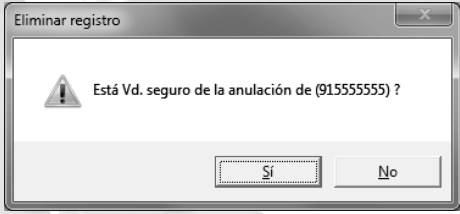


### 6.5.2.2 EJECUCIÓN DEL FORMULARIO

En ejecución, el formulario presenta el siguiente aspecto:

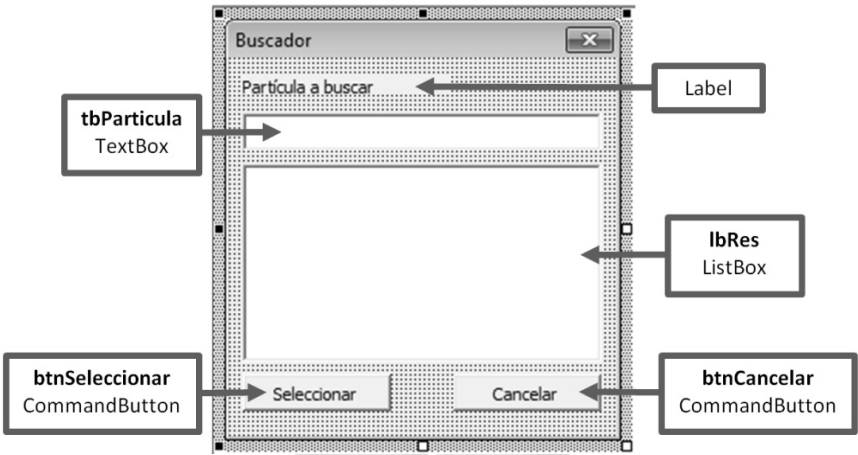
Captura de pantalla del formulario 'Mantenimiento' en ejecución. El formulario muestra los campos 'Código' (91333333) y 'Nombre' (Juan Antonio). Los botones 'Nuevo', 'Editar', 'Grabar', 'Cancelar' y 'Borrar' están visibles. El botón 'Buscar' está activado. El indicador de página muestra '1 de 7'. El botón 'Fin' está visible. El formulario tiene un título 'Mantenimiento' y un botón de cerrar 'X'.

La funcionalidad prevista para cada elemento es la siguiente:

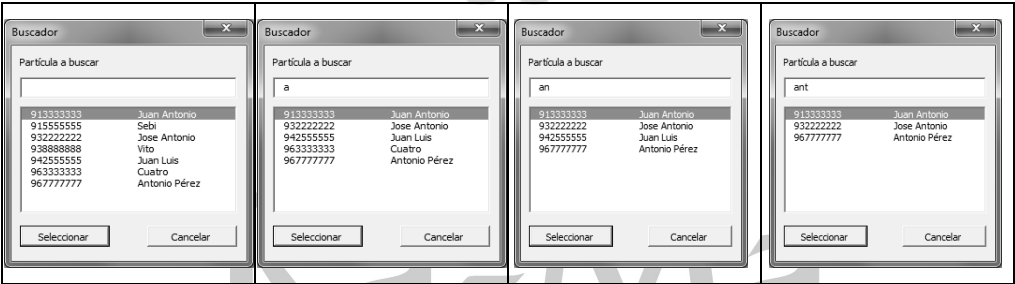
Objeto	Comentario
Botón <b>Nuevo</b>	Permite crear un nuevo registro.
Botón <b>Editar</b>	Permite modificar el registro en curso.
Botón <b>Grabar</b>	Permite grabar el registro en curso.
Botón <b>Cancelar</b>	Deja el registro como está sin hacer efectivo ninguno de los cambios que se hayan producido sobre el formulario.
Botón <b>Borrar</b>	<p>Permite borrar el registro en curso. En caso de seleccionar esta opción, se solicitará confirmación mediante un mensaje como el siguiente:</p> 
Cuadro de texto <b>Código</b>	Permite introducir el código del registro.
Cuadro de texto <b>Nombre</b>	Permite introducir el nombre.
Panel <b>Orden</b>	Agrupar las opciones de ordenación.
Opción <b>Código</b>	Ordena por código.
Opción <b>Nombre</b>	Ordena por nombre.
Botón <b>Buscar</b>	Abre el cuadro de diálogo de búsqueda.
Botón <b>Primero</b>	Avanza hasta el primer registro de la tabla.
Botón <b>Anterior</b>	Avanza hasta el registro anterior al registro en curso.

Cuadro de texto <b>Indicador</b>	Indica la posición del registro en curso y cuantos registros hay en la tabla.
Botón <b>Siguiente</b>	Avanza hasta el registro siguiente al registro en curso.
Botón <b>Ultimo</b>	Avanza hasta el último registro de la tabla.
Botón <b>Fin</b>	Permite cerrar el mantenimiento.

Con respecto al buscador, tenemos el siguiente formulario (**frmBuscador**):



En ejecución, el buscador posee el siguiente aspecto:



Podemos observar que, en la medida que se indique parte del texto que se desea buscar, la lista con los registros a seleccionar se irá reduciendo para mostrar sólo aquellos que efectivamente contengan dicha partícula en su nombre.

La funcionalidad de los elementos de la pantalla es la siguiente:

Objeto	Comentario
Botón <b>Seleccionar</b>	Cierra el buscador devolviendo al mantenimiento el registro seleccionado.
Botón <b>Cancelar</b>	Cierra el buscador sin seleccionar ningún registro.
Cuadro de Texto <b>Partícula</b>	Partícula a buscar.
Lista	Contiene los registros que se pueden elegir. También es posible seleccionar un registro haciendo doble clic sobre un elemento de la lista.

### 6.5.2.3 CÓDIGO FUENTE DEL FORMULARIO

El código correspondiente al formulario principal (**frmContactos**) es el siguiente

```
Option Explicit On
```

```
Dim nuevo As Boolean
```

```
Public respuesta As String
```

```
Private Sub pon_btn_inicio()
```

```
    btnNuevo.Enabled = True
```

```
    btnEditar.Enabled = True
```

```
    btnCancelar.Enabled = False
```

```
    btnBuscar.Enabled = True
```

```
    btnGrabar.Enabled = False
```

```
    btnBorrar.Enabled = True
```

```
    btnPrimero.Enabled = True
```

```
    btnAnterior.Enabled = True
```

```
    btnSiguiente.Enabled = True
```

```
    btnUltimo.Enabled = True
```

```
    btnSalir.Enabled = True
```

```
    tbCodigo.Enabled = False
```

```
    tbNombre.Enabled = False
```

```
End Sub
```

```
Sub pon_btn_normal()  
    btnNuevo.Enabled = False  
    btnEditar.Enabled = False  
    btnCancelar.Enabled = True  
    btnBuscar.Enabled = False  
    btnGrabar.Enabled = True  
    btnBorrar.Enabled = False  
    btnPrimero = False  
    btnAnterior = False  
    btnSiguiente = False  
    btnUltimo = False  
    btnSalir = False  
  
    tbCodigo.Enabled = True  
    tbNombre.Enabled = True  
End Sub  
  
Sub pon_btn_anadir()  
    btnNuevo.Enabled = False  
    btnEditar.Enabled = False  
    btnCancelar.Enabled = True  
    btnBuscar.Enabled = False  
    btnGrabar.Enabled = True  
    btnBorrar.Enabled = False  
    btnPrimero = False  
    btnAnterior = False  
    btnSiguiente = False  
    btnUltimo = False  
    btnSalir = False  
  
    tbCodigo.Enabled = True  
    tbNombre.Enabled = True  
End Sub  
  
Private Sub btnBuscar_Click()  
    frmBuscador.Show()  
    If respuesta <> "" Then  
        Dim f As Integer  
        f = existe(respuesta)  
        If f <> -1 Then  
            Cells(f, 1).Select()  
            mostrar_registro(fNumReg)  
            mostrar_posicion()  
        End If  
    End If  
End Sub  
  
Private Sub btnCancelar_Click()  
    If fNumReg > fNumRegs Then  
        Cells(fNumRegs + 1, 1).Select()  
    End If  
    mostrar_registro(fNumReg)  
    mostrar_posicion()
```



```
pon_btn_inicio()
nuevo = False
End Sub

Private Sub btnEditar_Click()
    mostrar_registro(fNumReg)
    mostrar_posicion()
    pon_btn_normal()
    nuevo = False
    tbCodigo.SetFocus()
End Sub

Function fNumRegs() As Integer
    Dim nfilas As Integer
    nfilas = Cells.Find("*", SearchOrder:=xlByRows, _
        LookIn:=xlValues, SearchDirection:=xlPrevious).Row
    fNumRegs = nfilas - 1
End Function

Function fNumReg() As Integer
    fNumReg = ActiveCell.Row - 1
End Function

Private Sub btnSalir_Click()
    Unload(Me)
End Sub

Sub mostrar_posicion()
    indicadorSB.Text = fNumReg & " de " & fNumRegs
End Sub

Sub mostrar_registro(nReg As Integer)
    tbCodigo.Text = Cells(nReg + 1, 1).Value
    tbNombre.Text = Cells(nReg + 1, 2).Value
End Sub

Private Sub btnPrimero_Click()
    Range("A2").Select()
    mostrar_registro(fNumReg)
    mostrar_posicion()
End Sub

Private Sub btnSiguiente_Click()
    Dim fila As Integer
    If fNumReg < fNumRegs Then
        fila = fNumReg + 2
    Else
        fila = fNumRegs + 1
    End If
    Cells(fila, 1).Select()
    mostrar_registro(fNumReg)
    mostrar_posicion()
End Sub
```

```
Private Sub btnAnterior_Click()  
    Dim fila As Integer  
    If fNumReg > 1 Then  
        fila = fNumReg  
    Else  
        fila = 2  
    End If  
    Cells(fila, 1).Select()  
    mostrar_registro(fNumReg)  
    mostrar_posicion()  
End Sub  
  
Private Sub btnUltimo_Click()  
    Cells(fNumRegs + 1, 1).Select()  
    mostrar_registro(fNumReg)  
    mostrar_posicion()  
End Sub  
  
Private Sub btnNuevo_Click()  
    pon_btn_anadir()  
    Cells(fNumRegs + 2, 1).Select()  
    tbCodigo.Text = ""  
    tbNombre.Text = ""  
    tbCodigo.SetFocus()  
    nuevo = True  
End Sub  
  
Private Sub btnGrabar_Click()  
    Dim codigo As String  
    codigo = tbCodigo.Text  
    If existe(codigo) <> -1 And nuevo Then  
        MsgBox("Código (" & codigo & ") ya existe.")  
        tbCodigo.SetFocus()  
        Exit Sub  
    End If  
    ActiveCell = tbCodigo.Text  
    ActiveCell.Offset(0, 1).Select()  
    ActiveCell = tbNombre.Text  
    ordenar()  
    Cells(fNumReg + 1, 1).Select()  
    pon_btn_inicio()  
    mostrar_posicion()  
End Sub  
  
Private Sub btnBorrar_Click()  
    If MsgBox("Está Vd. seguro de la anulación de (" & tbCodigo.Text  
& ") ? ", vbYesNo + vbExclamation, "Eliminar registro") = vbYes Then  
        Range(Cells(fNumReg + 1, 1), Cells(fNumReg + 1,  
2)).Delete(xlShiftUp)  
        If fNumReg > fNumRegs Then  
            Cells(fNumRegs + 1, 1).Select()  
        End If  
    End If  
End If
```

```
pon_btn_inicio()
mostrar_registro(fNumReg)
mostrar_posicion()
End Sub

Private Sub rbCodigo_Click()
ordenar()
End Sub

Private Sub rbNombre_Click()
ordenar()
End Sub

Private Sub UserForm_Initialize()
Range("A2").Select()
pon_btn_inicio()
mostrar_registro(fNumReg)
mostrar_posicion()
nuevo = False
rbCodigo.Value = True
rbNombre = False
End Sub

Sub ordenar()
Dim r As Range
r = Range("A1:B11")
r = Range(Cells(1, 1), Cells(fNumRegs + 1, 2))
If rbCodigo.Value = True Then
r.Sort(Key1:="Teléfono", order1:=xlAscending, Header:=xlYes)
Else
r.Sort(Key1:="Nombre", order1:=xlAscending, Header:=xlYes)
End If
End Sub

Function existe(codigo As String) As Integer
Dim s, r As Range
r = Range(Cells(1, 1), Cells(fNumRegs + 1, 1))
s = r.Find(What:=codigo, LookIn:=xlValues)
If Not s Is Nothing Then
existe = s.Row
Else
existe = -1
End If
End Function
```

Ra-Ma®

El código correspondiente al buscador (**frmBuscador**) es el siguiente:

```
Option Explicit

Sub buscar()
    Dim guardar As Boolean
    Dim fr As Integer
    fr = 0
    Dim partícula As String
    partícula = tbPartícula.Text
    Dim f, filas As Integer
    filas = frmContactos.fNumRegs
    lbRes.Clear()
    For f = 2 To filas + 1
        Dim código, nombre As String
        código = Cells(f, 1)
        nombre = Cells(f, 2)
        If partícula <> "" Then
            If InStr(UCase(nombre), UCase(partícula)) <> 0 Or
InStr(UCase(código), UCase(partícula)) <> 0 Then
                guardar = True
            Else
                guardar = False
            End If
        Else
            guardar = True
        End If
        If guardar Then
            lbRes.AddItem()
            lbRes.List(fr, 0) = código
            lbRes.List(fr, 1) = nombre
            fr = fr + 1
        End If
    Next f
    If lbRes.ListCount > 0 Then
        lbRes.ListIndex = 0
        btnSeleccionar.Enabled = True
    Else
        btnSeleccionar.Enabled = False
    End If
End Sub

Private Sub btnCancelar_Click()
    frmContactos.respuesta = ""
    Unload(Me)
End Sub

Private Sub btnSeleccionar_Click()
    seleccionar()
    Unload(Me)
End Sub
```

```
Sub seleccionar()  
    Dim index As Integer  
    index = lbRes.ListIndex  
    If index <> -1 Then  
        frmContactos.respuesta = lbRes.List(index, 0)  
    End If  
End Sub  
  
Private Sub lbRes_DblClick(ByVal Cancel As MSForms.ReturnBoolean)  
    seleccionar()  
    Unload(Me)  
End Sub  
  
Private Sub tbParticula_Change()  
    buscar()  
End Sub  
  
Public Sub UserForm_Initialize()  
    frmContactos.respuesta = ""  
    btnSeleccionar.Default = True  
    lbRes.ColumnCount = 2  
    btnSeleccionar.Enabled = False  
    buscar()  
    tbParticula.SetFocus()  
End Sub
```

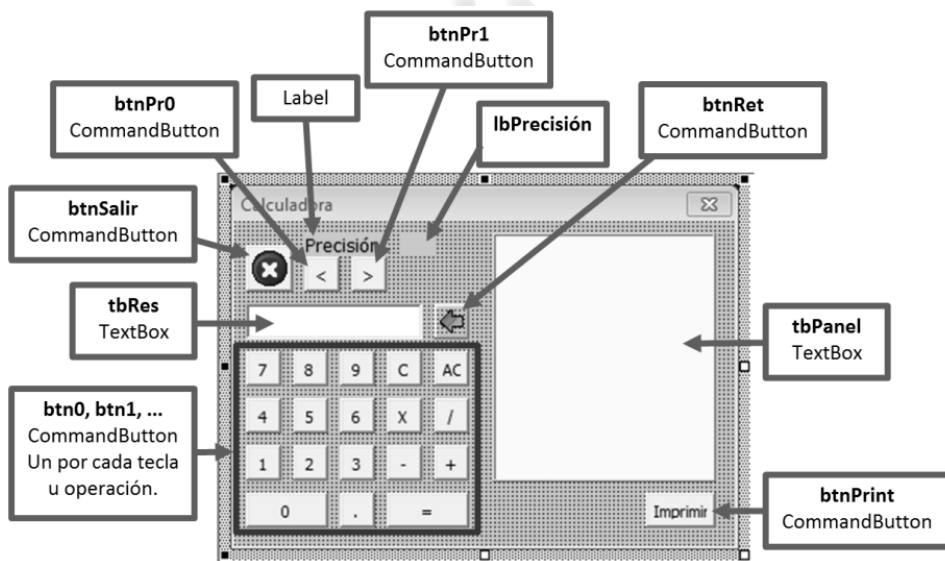
### 6.5.3 Calculadora

En este ejemplo, se muestra como confeccionar una calculadora simple en la que como funcionalidades a destacar indicaremos la posibilidad de utilizar el teclado numérico y también la posibilidad de imprimir la lista de operaciones realizadas por la impresora del sistema.

Ra-Ma®

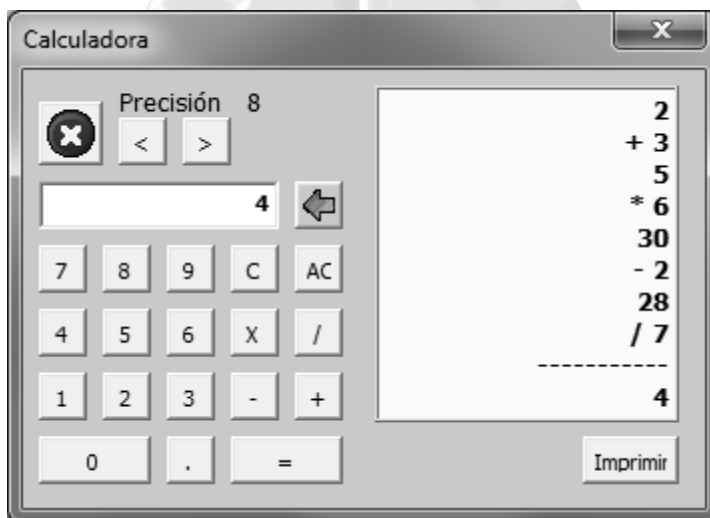
### 6.5.3.1 ELEMENTOS DEL FORMULARIO

El formulario (denominado **frmCal**) contiene los siguientes elementos:



### 6.5.3.2 EJECUCIÓN DEL FORMULARIO

En ejecución posee el siguiente aspecto:



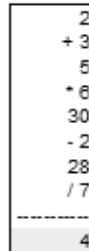
La funcionalidad prevista para cada elemento es la siguiente:

Objeto	Comentario
Botones <b>0,1,2...9</b>	Permite introducir dígito numérico.
Botón <b>Salir</b> ✕	Permite cerrar la calculadora.
Botones <b>&lt; &gt;</b>	Permiten disminuir o aumentar la precisión (número de decimales a considerar para los cálculos).
Caja de texto <b>Resultado</b>	Muestra el número que se está introduciendo o el resultado del cálculo.
Botón <b>Retroceso</b> ←	Permite eliminar el último dígito introducido.
Botón <b>C</b>	Anula el último valor entrado.
Botón <b>AC</b>	Anula todo el cálculo y permite volver a empezar.
Botón <b>.</b>	Introduce el separador decimal.
Botón <b>=</b>	Ejecuta el cálculo
Botón <b>Imprimir</b>	Imprime por impresora el conjunto de operaciones que se hallan reflejadas en el cuadro de texto de operaciones.
Caja de texto <b>Operaciones</b>	Es el cuadro de la derecha que contiene cada una de las operaciones que se van realizando.

Para realizar cualquier operación, basta con hacer clic con el mouse sobre el botón asociado al número que se desea introducir y a continuación, la operación a realizar con el mismo seguido de otro valor y así sucesivamente hasta que se pulse el signo igual para obtener el resultado final. También es posible introducir los valores y las operaciones con el teclado numérico. En este caso, al pulsar la tecla **INTRO** equivale a pulsar la tecla igual (=). Observaremos que las operaciones se reflejan en el panel de la derecha además de ir visualizando el resultado en el cuadro de texto situado sobre los botones numéricos de la pantalla.

Las operaciones se realizan con una precisión 4 (número de decimales) pero se puede aumentar hasta 8.

Por último, se puede imprimir por la impresora del sistema el conjunto de operaciones realizadas simplemente pulsando sobre el botón **Imprimir**. La impresión de este ejemplo sería tal y como se muestra a continuación:



```

2
+ 3
5
* 6
30
- 2
28
/ 7
-----
4

```

### 6.5.3.3 CÓDIGO FUENTE DEL FORMULARIO

El código correspondiente al buscador (**frmCal**) es el siguiente:

```

Public memoria As String
Public borrar As Boolean
Public total As Boolean
Public haypunto As Boolean
Public coma As String
Public precision As Integer
Const maxPrecision As Integer = 8

Dim op, opant As Integer

Private Sub b0_KeyDown(ByVal KeyCode As MSForms.ReturnInteger, ByVal Shift As Integer)
    mostrar(0)
End Sub

Private Sub b1_Click()
    mostrar(1)
End Sub

Private Sub b2_Click()
    mostrar(2)
End Sub

Private Sub b3_Click()
    mostrar(3)
End Sub

```



```
Private Sub b4_Click()  
    mostrar(4)  
End Sub  
  
Private Sub b5_Click()  
    mostrar(5)  
End Sub  
  
Private Sub b6_Click()  
    mostrar(6)  
End Sub  
  
Private Sub b7_Click()  
    mostrar(7)  
End Sub  
Private Sub b8_Click()  
    mostrar(8)  
End Sub  
  
Private Sub b9_Click()  
    mostrar(9)  
End Sub  
  
Private Sub b0_Click()  
    mostrar(0)  
End Sub  
  
Private Sub bp_Click()  
    If InStr(tbRes.Text, coma) <> 0 And borrar = False Then haypunto  
= True Else haypunto = False  
    If Not (haypunto) Then  
        mostrar(coma)  
    End If  
End Sub  
  
Private Sub bsum_Click()  
    sumar()  
End Sub  
  
Private Sub bres_Click()  
    restar()  
End Sub  
  
Private Sub bmul_Click()  
    multiplicar()  
End Sub  
  
Private Sub bdiv_Click()  
    dividir()  
End Sub
```

```
Private Sub bigual_Click()  
    igual()  
End Sub  
  
Sub igual()  
    If memoria <> "" Then total = True  
    Call opera(memoria, tbRes.Text)  
    memoria = ""  
    op = 0  
    btRet.Enabled = False  
    agregar_Panel(memoria)  
End Sub  
  
Sub sumar()  
    Call opera(memoria, tbRes.Text)  
    op = 1  
End Sub  
  
Sub restar()  
    Call opera(memoria, tbRes.Text)  
    op = 2  
End Sub  
  
Sub multiplicar()  
    Call opera(memoria, tbRes.Text)  
    op = 3  
End Sub  
  
Sub dividir()  
    Call opera(memoria, tbRes.Text)  
    op = 4  
End Sub  
  
Sub limpiar_ultimo()  
    tbRes.Text = "0"  
End Sub  
  
Sub limpiar_todo()  
    tbRes.Text = "0"  
    memoria = ""  
    borrar = True  
    tbPanel.Text = ""  
End Sub  
  
Sub opera(val1 As String, val2 As String)  
    opant = op  
    If memoria = "" Then  
        memoria = val2  
    Else  
        Select Case opant  
            Case 1 'Suma  
                If memoria = "" Then  
                    memoria = val2
```

```

        Else
            memoria = Round(CDbl(val1) + CDbl(val2),
precision)
            agregar_Panel("+ " & val2)
        End If
    Case 2 ' Resta
        If memoria = "" Then
            memoria = val2
        Else
            memoria = Round(CDbl(val1) - CDbl(val2),
precision)
            agregar_Panel("- " & val2)
        End If
    Case 3 ' Multiplicación
        If memoria = "" Then
            memoria = val2
        Else
            memoria = Round(CDbl(val1) * CDbl(val2),
precision)
            agregar_Panel("* " & val2)
        End If
    Case 4 ' División
        If memoria = "" Then
            memoria = val2
        Else
            If CDbl(val2) <> 0 Then
                memoria = Round(CDbl(val1) / CDbl(val2),
precision)
            Else
                memoria = 0
                agregar_Panel("ERROR")
            End If
            agregar_Panel("/ " & val2)
        End If
    End Select
End If
tbRes.Text = memoria
If total Then
    agregar_Panel("-----")
End If
agregar_Panel(memoria)
borrar = True
btRet.Enabled = True
total = False
End Sub

Sub agregar_Panel(s As String)
    If tbPanel.Text <> "" And s <> "" Then
        tbPanel.Text = tbPanel.Text & vbCrLf
    End If
    tbPanel.Text = tbPanel.Text & s
End Sub

```

```
Private Sub btnC_Click()
    limpiar_ultimo()
End Sub

Private Sub btnAC_Click()
    limpiar_todo()
End Sub

Sub retroceso()
    Dim s As String
    s = tbRes.Text
    If s <> "" And s <> "0" And Len(s) > 1 Then
        s = Left(s, Len(s) - 1)
    Else
        s = "0"
        btRet.Enabled = False
    End If
    tbRes.Text = s
End Sub

Sub mas_precision()
    precision = precision + 1
    If precision > maxPrecision Then precision = maxPrecision
    lbPrecision.Caption = precision
    tbPanel.SetFocus()
End Sub

Sub menos_precision()
    precision = precision - 1
    If precision < 0 Then precision = 0
    lbPrecision.Caption = precision
    tbPanel.SetFocus()
End Sub

Private Sub btnOff_Click()
    Unload(Me)
End Sub

Sub mostrar(c As String)
    If borrar Then
        borrar = False
        tbRes.Text = ""
    End If
    If tbRes.Text = "0" And c <> coma Then tbRes.Text = ""
    If tbRes.Text = coma Then tbRes.Text = "0" & coma
    tbRes.Text = tbRes.Text & c
    btRet.Enabled = True
End Sub

Private Sub btnPr0_Click()
    menos_precision()
End Sub
```

```
Private Sub btnPr1_Click()  
    mas_precision()  
End Sub  
  
Private Sub btnPrint_Click()  
    imprimir(tbPanel.Text)  
    tbRes.SetFocus()  
End Sub  
  
Private Sub btnSalir_Click()  
    Unload(Me)  
End Sub  
  
Private Sub btRet_Click()  
    retroceso()  
End Sub  
  
Private Sub Image1_Click()  
    Unload(Me)  
End Sub  
  
Private Sub tbPanel_KeyDown(ByVal KeyCode As MSForms.ReturnInteger,  
ByVal Shift As Integer)  
    If KeyCode >= 96 And KeyCode <= 105 Then  
        mostrar(KeyCode - 96)  
    End If  
    If KeyCode = 110 Then  
        mostrar(coma)  
    End If  
    Select Case KeyCode  
        Case 107 ' Sumar  
            sumar()  
        Case 109 ' Restar  
            restar()  
        Case 106 ' Multiplicar  
            multiplicar()  
        Case 111 ' Dividir  
            dividir()  
        Case 48 ' Igual  
            igual()  
        Case 13  
            igual()  
        Case 65 ' Limpiar todo (A)  
            limpiar_todo()  
        Case 67 ' limpiar ultimo (C)  
            limpiar_ultimo()  
        Case 8 ' Retroceso  
            retroceso()  
        Case 226 ' Menos precision  
            If Shift = 0 Then  
                menos_precision()  
            Else  
                mas_precision()  
            End If  
        End Select  
End Sub
```

```

        End If
        Case 70 ' Fin (F)
            Unload(Me)
        Case 80 ' Print (P)
            imprimir(tbPanel.Text)
    End Select

End Sub

Sub imprimir(ByVal texto As String)
    Dim existe As Boolean
    Dim h As Worksheet
    For Each h In ActiveWorkbook.Worksheets
        If h.Name = "work" Then existe = True
    Next
    If Not (existe) Then

ActiveWorkbook.Sheets.Add(Before:=Worksheets(Worksheets.Count))
        ActiveWorkbook.Sheets(Worksheets.Count).Name = "work"
    End If
    Worksheets("work").Select()
    Worksheets("work").UsedRange.Delete()
    Dim lineas As Object
    lineas = Split(texto, Constants.vbNewLine)
    Dim i As Integer
    For i = 0 To UBound(lineas)
        Debug.Print(lineas(i))
        Cells(i + 1, 1) = "" & lineas(i)
    Next
    Worksheets("work").Columns("A").AutoFit()
    Worksheets("work").Columns("A").HorizontalAlignment = xlRight
    Range(Cells(1, 1), Cells(i, 1),
1)) .BorderAround(Linestyle:=xlContinuous, Weight:=xlThin, Color:=5)
    Range(Cells(i, 1), Cells(i, 1)).Interior.ColorIndex = 6
    Range("A1:A6").Select()
    Range(Cells(1, 1), Cells(i, 1)).NumberFormat = "@"
    Range(Cells(1, 1), Cells(i, 1)).PrintOut()
    Worksheets("work").UsedRange.Delete()
    Worksheets("work").Range("A1").Select()
End Sub

Private Sub UserForm_Initialize()
    memoria = ""
    borrar = False
    tbRes.Text = "0"
    btRet.Enabled = False
    total = False
    tbPanel.ScrollBars = fmScrollBarsBoth
    tbPanel.SetFocus()
    Dim textoNumero As Double
    TestNumero = Format("0.1", "#. #")
    If InStr(TestNumero, ",") Then coma = "," Else coma = "."
    precision = 4
    lbPrecision.Caption = precision
End Sub

```

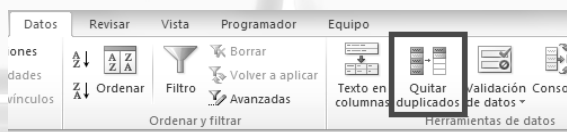
Para relanzar la llamada desde la opción **Macros** de **MS Excel**, bastará con insertar un módulo y crear el siguiente método:

```
Sub lanza_cal()
    frmCal.Show()
End Sub
```

## 6.5.4 Deduplicar filas

En ocasiones, nos encontramos con una hoja que posee información duplicada y necesitamos poder ‘deduplicarla’ para obtener una serie de filas únicas.

Esta función, podemos encontrarla en el grupo de Herramientas de datos que se halla en la ficha Datos y concretamente, se llama Quitar duplicados:

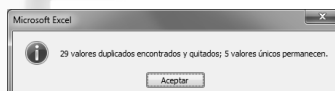
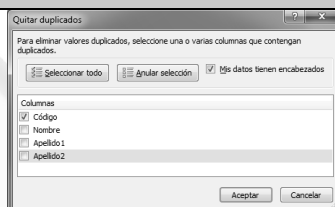


Por ejemplo, supongamos que disponemos de una hoja con la siguiente información:

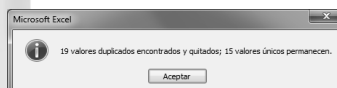
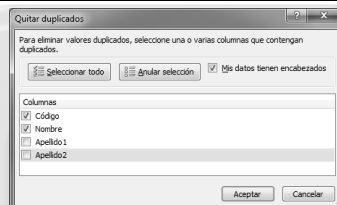
Original				
	A	B	C	D
	Código	Nombre	Apellido1	Apellido2
1				
2		1 Paco	Roca	Sanz
3		2 Juan	Sánchez	Roca
4		3 Xavi	Gómez	Sánchez
5		4 Jordi	Sanz	Gómez
6		1 Paco	Roca	Sanz
7		2 Eduardo	Roca	Sanz
8		3 Paco	Sánchez	Roca
9		4 Juan	Gómez	Sánchez
10		1 Xavi	Sanz	Gómez
11		2 Jordi	Sanz	Roca
12		3 Eduardo	Sánchez	Sánchez
13		1 Paco	Gómez	Gómez
14		2 Paco	Roca	Sanz
15		3 Juan	Sánchez	Roca
16		4 Xavi	Gómez	Sánchez
17		1 Jordi	Sanz	Gómez
18		2 Eduardo	Roca	Sanz
19		3 Paco	Sánchez	Roca
20		4 Juan	Gómez	Sánchez
21		1 Xavi	Sanz	Gómez
22		2 Jordi	Sanz	Roca
23		3 Eduardo	Sánchez	Sánchez
24		1 Paco	Gómez	Gómez
25		2 Paco	Roca	Sanz
26		3 Juan	Sánchez	Roca
27		4 Xavi	Gómez	Sánchez
28		1 Jordi	Sanz	Gómez
29		2 Eduardo	Roca	Sanz
30		3 Paco	Sánchez	Roca
31		4 Juan	Gómez	Sánchez
32		1 Xavi	Sanz	Gómez
33		2 Jordi	Sanz	Roca
34		3 Eduardo	Sánchez	Sánchez
35		5 Paco	Gómez	Gómez

**Deduplicar en base al Código**

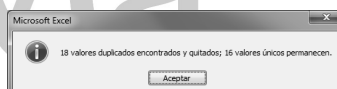
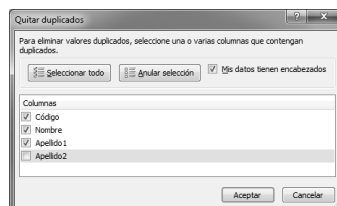
	A	B	C	D
1	Código	Nombre	Apellido1	Apellido2
2		1 Paco	Roca	Sanz
3		2 Juan	Sánchez	Roca
4		3 Xavi	Gómez	Sánchez
5		4 Jordi	Sanz	Gómez
6		5 Paco	Gómez	Gómez

**Acciones / Mensajes****Deduplicar en base al Código y Nombre**

	A	B	C	D
1	Código	Nombre	Apellido1	Apellido2
2		1 Paco	Roca	Sanz
3		2 Juan	Sánchez	Roca
4		3 Xavi	Gómez	Sánchez
5		4 Jordi	Sanz	Gómez
6		2 Eduardo	Roca	Sanz
7		3 Paco	Sánchez	Roca
8		4 Juan	Gómez	Sánchez
9		1 Xavi	Sanz	Gómez
10		2 Jordi	Sanz	Roca
11		3 Eduardo	Sánchez	Sánchez
12		2 Paco	Roca	Sanz
13		3 Juan	Sánchez	Roca
14		4 Xavi	Gómez	Sánchez
15		1 Jordi	Sanz	Gómez
16		5 Paco	Gómez	Gómez

**Acciones / Mensajes****Deduplicar en base al Código, Nombre y Apellido1**

	A	B	C	D
1	Código	Nombre	Apellido1	Apellido2
2		1 Paco	Roca	Sanz
3		2 Juan	Sánchez	Roca
4		3 Xavi	Gómez	Sánchez
5		4 Jordi	Sanz	Gómez
6		2 Eduardo	Roca	Sanz
7		3 Paco	Sánchez	Roca
8		4 Juan	Gómez	Sánchez
9		1 Xavi	Sanz	Gómez
10		2 Jordi	Sanz	Roca
11		3 Eduardo	Sánchez	Sánchez
12		1 Paco	Gómez	Gómez
13		2 Paco	Roca	Sanz
14		3 Juan	Sánchez	Roca
15		4 Xavi	Gómez	Sánchez
16		1 Jordi	Sanz	Gómez
17		5 Paco	Gómez	Gómez

**Acciones / Mensajes**

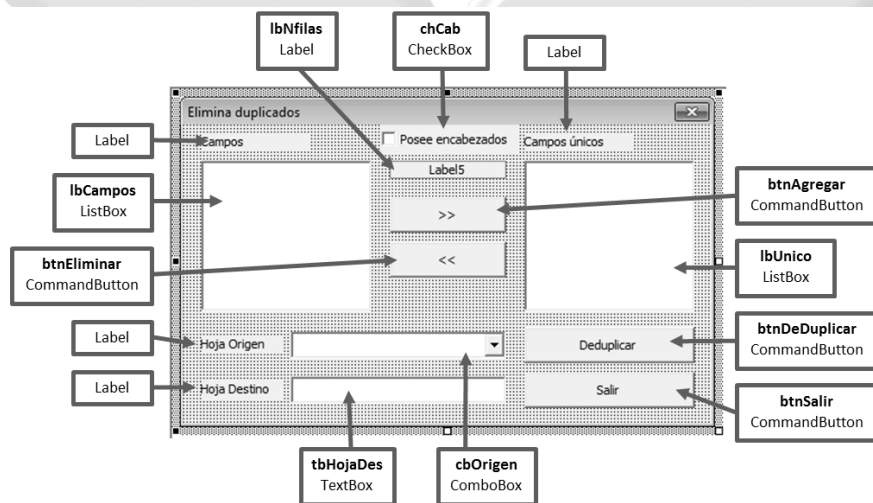


Si nos fijamos sólo en la **columna A (Código)** observaremos que existen códigos que se repiten y cuyos valores únicos son el **1, 2, 3, 4 y 5**. Por tanto, si quisiéramos **‘deduplicar’** atendiendo sólo al código, obtendríamos un resultado en el que sólo existirían 5 filas de datos (una por cada código). Sin embargo, si consideramos que la repetición existe cuando el código y el nombre son iguales pero no en cualquier otro caso, el resultado contendrá 15 filas de datos ya que **‘1 – Paco’** es diferente de **‘1 – Xavi’**.

El siguiente ejemplo realiza el mismo cometido pero nos sirve para probar algunos de los elementos que podemos utilizar en los formularios. Como característica que diferencia a la función **Quitar duplicados**, podemos destacar que podemos elegir la hoja origen en la que se hallan los datos a **‘deduplicar’** y también, seleccionar la hoja destino en la que queremos depositar el resultado.

#### 6.5.4.1 ELEMENTOS DEL FORMULARIO

El formulario (denominado **frmDeduplicar**) contiene los siguientes elementos:



#### 6.5.4.2 EJECUCIÓN DEL FORMULARIO

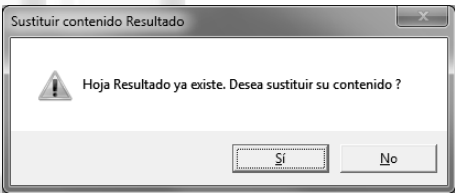
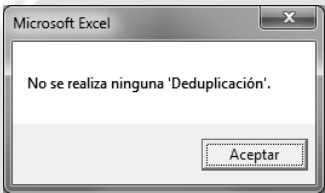
En ejecución, posee el siguiente aspecto:

Ra-Ma®

Posee encabezados	No posee encabezados

La funcionalidad prevista para cada elemento es la siguiente:

Objeto	Comentario
Lista <b>Campos</b>	Muestra los campos que podemos seleccionar para tener en cuenta la deduplicación.
Lista <b>Campos únicos</b>	Muestra los campos que se tendrán en cuenta para la deduplicación.
Etiqueta <b>filas</b>	Indica el número de filas de la hoja origen a tratar.
Casilla de verificación <b>Posee encabezados</b>	Permite indicar si la hoja origen posee encabezados a tener en cuenta como nombres de campos y por tanto, no considerar en la deduplicación o no.
Botón <b>Agregar (&gt;&gt;)</b>	Permite añadir un campo a la lista de campos únicos a tener en cuenta para la deduplicación. Sólo se activa si hay campos que se pueden agregar.
Botón <b>Eliminar (&lt;&lt;)</b>	Permite eliminar un campo de la lista de campos únicos a tener en cuenta para la deduplicación. Sólo se activa si hay campos que se pueden eliminar.

Lista desplegable <b>Hoja Origen</b>	Muestra las hojas existentes en el libro sobre las que se puede realizar una deduplicación.
Caja de texto <b>Hoja Destino</b>	<p>Permite indicar el nombre de la hoja sobre la que colocar el resultado. Por defecto propone la hoja <b>Resultado</b> y, en caso de que ya exista se solicita confirmación para sustituir el contenido de la misma mediante la siguiente pregunta:</p>  <p>Si se indica <b>No</b>, se recibe el siguiente mensaje:</p> 
Botón <b>Deduplicar</b>	Realiza la ‘deduplicación’.
Botón <b>Salir</b>	Cierra el cuadro de diálogo de ‘deduplicar’.

### 6.5.4.3 CÓDIGO FUENTE DEL FORMULARIO

El código correspondiente al formulario (**frmDeDuplicar**) es el siguiente:

```
Option Explicit On
Public nfres As Integer

Private Sub btnAgregar_Click()
    agregar()
End Sub

Private Sub btnEliminar_Click()
    eliminar()
End Sub
```

```

Sub agregar()
    If lbCampos.ListCount <> 0 And lbCampos.ListIndex <> -1 Then
        Dim s As String
        s = lbCampos.List(lbCampos.ListIndex)
        lbUnicos.AddItem(s)
        lbCampos.RemoveItem(lbCampos.ListIndex)
    End If
    If lbCampos.ListCount = 0 Then btnAgregar.Enabled = False
    btnEliminar.Enabled = True
    btnDeDuplicar.Enabled = True
End Sub

Sub eliminar()
    If lbUnicos.ListCount <> 0 And lbUnicos.ListIndex <> -1 Then
        Dim s As String
        s = lbUnicos.List(lbUnicos.ListIndex)
        lbCampos.AddItem(s)
        lbUnicos.RemoveItem(lbUnicos.ListIndex)
    End If
    If lbUnicos.ListCount = 0 Then
        btnEliminar.Enabled = False
        btnDeDuplicar.Enabled = False
    End If
    btnAgregar.Enabled = True
End Sub

Private Sub btnDeDuplicar_Click()
    Dim hojades As String
    hojades = tbHojaDes.Text
    Dim cnhojades As String
    cnhojades = ""
    Dim existe As Boolean
    existe = False
    Dim h As Worksheet
    For Each h In Worksheets
        If h.Name = tbHojaDes.Text Then
            existe = True
        End If
    Next
    If existe Then
        If MsgBox("Hoja " & hojades & " ya existe. Desea sustituir su contenido ? ", vbYesNo + vbExclamation, "Sustituir contenido " & hojades) = vbYes Then
            Worksheets(hojades).Select()
            Worksheets(hojades).Range("A1").Select()
            Worksheets(hojades).UsedRange.Select()
            Worksheets(hojades).UsedRange.Delete()
            Worksheets(hojades).Range("A1").Select()
        Else
            MsgBox("No se realiza ninguna 'Deduplicación'.")
            Exit Sub
        End If
    Else

```

```
        Dim hay As Integer
        hay = Worksheets.Count
        Dim nueva As Worksheet
        nueva = Worksheets.Add(after:=Worksheets(hay),
Type:=xlWorksheet)
        Worksheets(nueva.Index).Name = tbHojaDes.Text
    End If
    deduplicar()
End Sub

Private Sub btnSalir_Click()
    Unload(Me)
End Sub

Private Sub cborigen_Change()
    inicializar()
End Sub

Private Sub chCab_Click()
    If tbHojaDes.Text <> "" Then
        inicializar()
    End If
End Sub

Private Sub lbCampos_Click()
    If lbCampos.ListCount > 0 Then btnAgregar.Enabled = True
End Sub

Private Sub lbCampos_DblClick(ByVal Cancel As MSForms.ReturnBoolean)
    agregar()
End Sub

Private Sub lbUnicos_Click()
    If lbUnicos.ListCount > 0 Then btnEliminar.Enabled = True
End Sub

Private Sub lbUnicos_DblClick(ByVal Cancel As MSForms.ReturnBoolean)
    eliminar()
End Sub

Private Sub UserForm_Click()

End Sub

Private Sub UserForm_Initialize()
    chCab.Value = True
    carga_origen()
    inicializar()
End Sub

Sub inicializar()
    Worksheets(cborigen.Text).Activate()
    fabrica_lista()
```

```

    btnDeDuplicar.Default = True
    tbHojaDes.Text = "Resultado"
    btnAgregar.Enabled = False
    btnEliminar.Enabled = False
    If lbUnicos.ListCount = 0 Then btnDeDuplicar.Enabled = False
    lbNfilas.Caption = numFilas & " filas"
    Worksheets(cborigen.Text).Activate()
    nfres = 0
End Sub

Sub carga_origen()
    Dim h As Worksheet
    For Each h In Worksheets
        If h.Name <> "Resultado" Then
            cborigen.AddItem(h.Name)
        End If
    Next
    If cborigen.ListCount > 0 Then
        cborigen.ListIndex = 0
    End If
End Sub

Sub deduplicar()
    nfres = 0
    Dim h, r As Worksheet
    h = Worksheets(cborigen.Text)
    r = Worksheets(tbHojaDes.Text)
    r.Select()
    r.UsedRange.Delete()
    Dim nfila, ncolumna, nfilas, ncolumnas, filaini As Integer
    h.Select()
    If chCab.Value = True Then
        filaini = 2
        agregar_resultado(h.Rows(1))
    Else
        filaini = 1
    End If
    nfilas = numFilas
    ncolumnas = numColumnas
    For nfila = filaini To nfilas
        ' columnas a tener en cuenta
        Dim cs() As String
        ReDim cs(1 To lbUnicos.ListCount)
        For ncolumna = 1 To lbUnicos.ListCount
            Dim colCmp As String
            Dim partes As Object
            partes = Split(lbUnicos.List(ncolumna - 1))
            colCmp = partes(0)
            cs(ncolumna) = Cells(nfila, CInt(colCmp))
        Next ncolumna
        If Not (busca_En_Resultado(cs)) Then
            agregar_resultado(h.Rows(nfila))
        End If
    Next nfila
End Sub

```

```
Next nfila
r.Select()
End Sub

Sub agregar_resultado(ByVal r As Range)
    Dim h As Worksheet
    h = Worksheets(tbHojaDes.Text)
    Dim nf As Integer
    nfres = nfres + 1
    Dim ncolumna As Integer
    For ncolumna = 1 To numColumnas
        h.Cells(nfres, ncolumna) = r.Cells(1, ncolumna)
    Next ncolumna
End Sub

Function busca_En_Resultado(ByVal cs As Object) As Boolean
    Dim r As Worksheet
    r = Worksheets(tbHojaDes.Text)
    Dim n, nf, nfs, ncolumna, filaini As Integer
    nfs = nfres
    n = UBound(cs)
    Dim existe As Boolean
    existe = False
    If chCab.Value = True Then
        filaini = 2
    Else
        filaini = 1
    End If
    For nf = filaini To nfs
        Dim igual As Boolean
        igual = True
        For ncolumna = 1 To n
            Dim colCmp As String
            Dim partes As Object
            partes = Split(lbUnicos.List(ncolumna - 1))
            colCmp = partes(0)
            Dim compara, contenido As String
            compara = cs(ncolumna)
            contenido = r.Cells(nf, CInt(colCmp))
            If compara <> contenido Then igual = False
        Next ncolumna
        If igual = True Then
            busca_En_Resultado = True
            Exit Function
        End If
    Next nf
    busca_En_Resultado = existe
End Function

Function numFilas() As Integer
    numFilas = Worksheets(cborigen.Text).UsedRange.Cells.Rows.Count
End Function
```

```
Function numColumnas() As Integer
    numColumnas =
Worksheets(cborigen.Text).UsedRange.Cells.Columns.Count
End Function

Sub fabrica_lista()
    Dim l As Object
    Dim inicio As Boolean
    inicio = True
    lbCampos.Clear()
    lbUnicos.Clear()
    Dim r As Range
    r = Worksheets(cborigen.Text).UsedRange.Rows(1)
    Dim c As Range
    For Each c In r.Columns
        Dim n As String
        If chCab.Value = True Then
            n = c.Value
            If n = "" Then
                n = c.Cells.Column & " - <???"
            Else
                n = c.Cells.Column & " - " & c.Value
            End If
        Else
            n = c.Cells.Column & " - C" & c.Cells.Column
        End If
        If inicio Then
            lbUnicos.AddItem(n)
            inicio = False
        Else
            lbCampos.AddItem(n)
        End If
    Next
End Sub
```

Para relanzar la llamada desde la opción **Macros de MS Excel**, bastará con insertar un módulo y crear el siguiente método:

```
Sub lanza_deduplicar()
    frmDeDuplicar.Show()
End Sub
```

Ra-Ma®



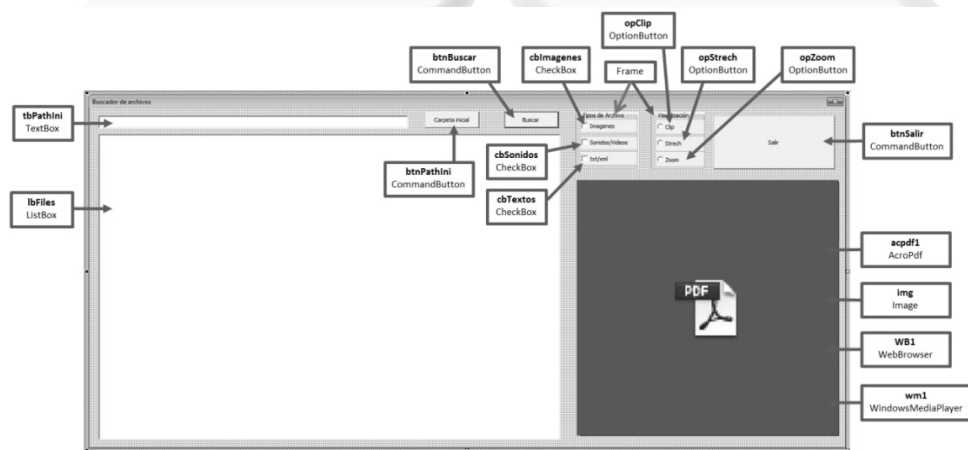
## 6.5.5 Buscador de archivos

El siguiente ejemplo, permite realizar la búsqueda de todos los archivos que se hallan contenidos en un determinado directorio y subdirectorios incluidos en el mismo que poseen unas determinadas extensiones. En definitiva, lo que pretende es encontrar y pre-visualizar dichos archivos para poder observar de forma rápida que contiene cada archivo sin necesidad de tener que abrirlos uno a uno.

Una vez mas, hay que insistir que la finalidad del ejemplo es la de mostrar parte de la funcionalidad de los elementos y objetos que participan en el mismo y no pretende fabricar un producto con todos los acabados que podemos imaginar.

### 6.5.5.1 ELEMENTOS DEL FORMULARIO

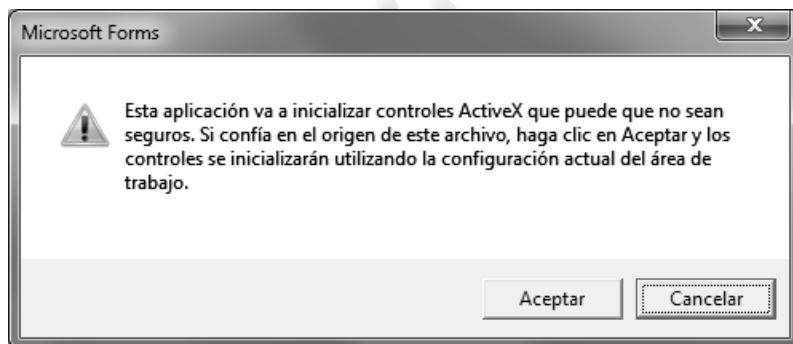
La pantalla del ejemplo contiene los siguientes elementos:



Observamos que los controles `acpdf1` (AcroPdf), `img` (Image), `WB1` (WebBrowser) y `wm1` (WindowsMediaPlayer) están superpuestos con la intención de compartir el mismo espacio en la pantalla. La idea es que, cada vez que se seleccione un archivo en la lista de archivos encontrados, su contenido se visualice en esta zona y se active el control asociado al tipo de archivo (o extensión del mismo). Es posible que alguno de los controles no se hallen disponibles en el equipo sobre el que se ejecute este ejemplo. En este caso, como primera solución, bastaría con comentar en el código cualquier referencia asociado al mismo y ya está. En general, en todos los equipos deberían funcionar sin problemas la visualización de archivos de tipo imagen, sonido y de texto (txt y html).

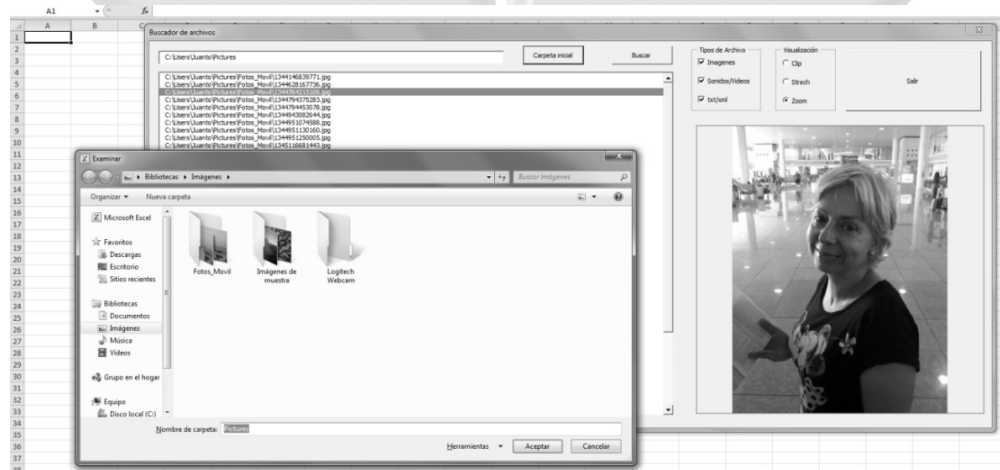
### 1.1.1.4 EJECUCIÓN DEL FORMULARIO

Al ejecutar el formulario, puede aparecer la siguiente advertencia:



Simplemente es para advertir del uso de controles de ActiveX y verificar la confianza sobre el origen del archivo.

El aspecto del formulario es el siguiente:



# Ra-Ma<sup>®</sup>

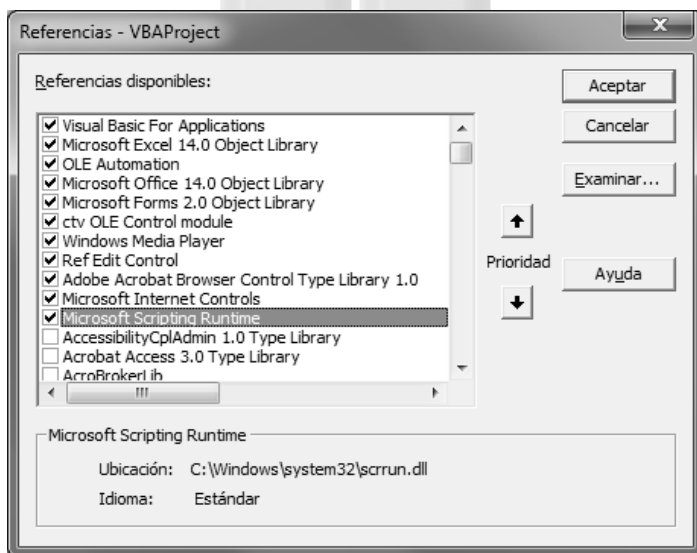
La funcionalidad prevista para cada elemento es la siguiente:

Objeto	Comentario
Cuadro de textos <b>Carpeta Inicial</b>	Muestra el directorio a partir del cual se buscarán los archivos a visualizar.
Botón <b>Carpeta inicial</b>	Permite seleccionar la carpeta a utilizar para buscar los archivos contenidos en la misma y sus subdirectorios.
Botón <b>Buscar</b>	Inicia la búsqueda a partir de la carpeta indicada y de los tipos de archivo seleccionados en las casillas de verificación.
Lista de <b>Archivos</b>	Muestra los archivos que se han hallados.
Panel <b>Tipos de Archivo</b>	Permite indicar qué tipos de archivo se desean buscar.
Panel <b>Visualización</b>	Sólo tiene utilidad para las imágenes y permite indicar el modo en el que deben mostrarse las imágenes (clip, strech, zoom)
Botón <b>Salir</b>	Permite cerrar el formulario.
Panel <b>Previsualización</b>	En realidad se trata de una zona en la que se superponen los diferentes controles que se necesitan para visualizar, en función del tipo de archivo, el contenido del mismo.

Si hacemos clic sobre el botón **Carpeta inicial**, se abre un cuadro de diálogo mediante el cual podemos navegar hasta el directorio a partir del cual deseamos iniciar la búsqueda. Los archivos que se buscan, se encasillan según su extensión, en alguna de las siguientes categorías:

Objeto	Comentario
<b>Imágenes</b>	Trata como imágenes las extensiones: <b>jpg, gif y bmp.</b>
<b>Sonidos / Videos</b>	Trata como sonidos o videos las extensiones: <b>mp3, wav, avi y mpg.</b>
<b>Txt / html</b>	Trata como textos las extensiones: <b>html, htm, txt y xml.</b> También considera como texto los archivos con extensión <b>pdf</b> aunque en función de la instalación, es posible que el control no esté disponible.

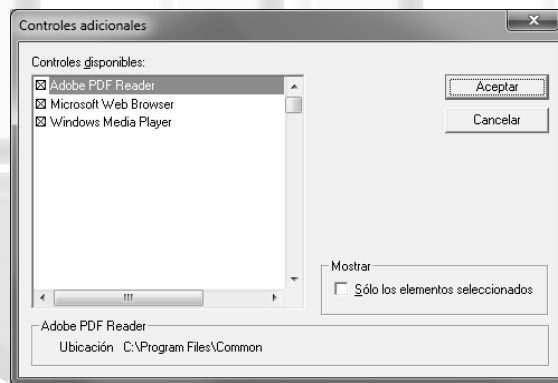
A continuación se muestran las referencias que deberíamos tener marcadas para que el ejemplo funcione correctamente:



Especialmente, las relativas a:

- Adobe Acrobat Browser Control Type Library 1.0
- Microsoft Internet Controls
- Microsoft Scripting Runtime

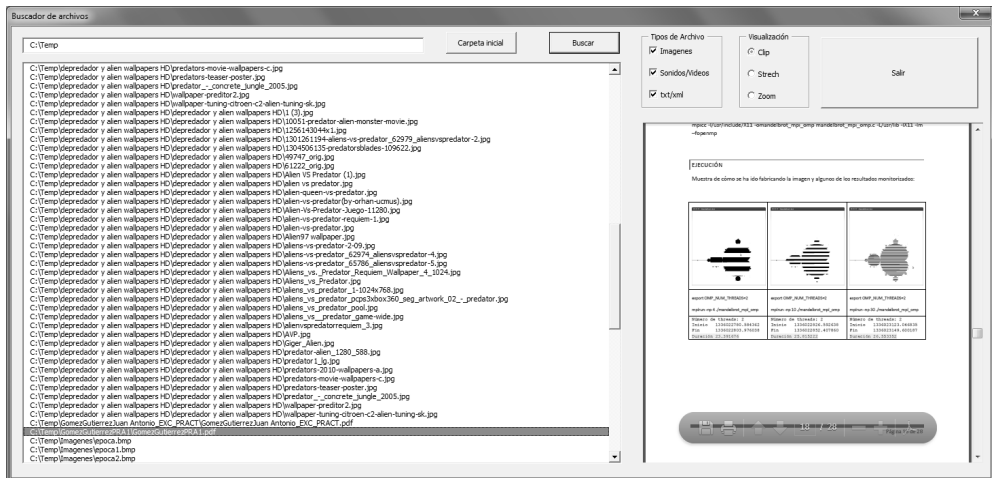
En nuestro **Cuadro de controles**, tendremos que agregar (si es preciso) algunos controles adicionales asegurándonos que, además de los controles que ya están seleccionados por defecto, tenemos agregados los siguientes:



Una vez seleccionados estos controles, nuestro **Cuadro de controles** debería mostrar los siguientes:



En el siguiente ejemplo, puede apreciarse el resultado de una búsqueda en la que se han recolectado todos los tipos de archivos previstos y se visualiza un archivo de tipo 'pdf':



### 6.5.5.2 CÓDIGO FUENTE DEL FORMULARIO

El código correspondiente al formulario (**frmBuscaArchivos**) es el siguiente:

```
Const ALTO As Integer = 370
Const ANCHO As Integer = 370

Private Sub btnPathIni_Click()
    With Application.FileDialog(msoFileDialogFolderPicker)
        .InitialFileName = tbPathIni.Text
        .Show()
    End With
    If .SelectedItems.Count > 0 Then
        tbPathIni.Text = .SelectedItems(1)
    End If
End Sub

Private Sub btnSalir_Click()
    Unload(Me)
End Sub

Private Sub btnBuscar_Click()
    Dim Fso As FileSystemObject
    Dim directorio As Folder

    DoEvents()
    lbFiles.Clear()

    Fso = New FileSystemObject
    directorio = Fso.GetFolder(tbPathIni.Text)
    Call expande(directorio)
```

```
' Seleccionamos el primer elemento
If lbFiles.ListCount > 0 Then
    lbFiles.Selected(0) = True
End If
End Sub

Private Sub expande(ByVal directorio As Folder)
    On Error GoTo salir
    Dim agregar As Boolean

    Dim fichero As file
    For Each fichero In directorio.Files
        Dim tipo As Integer
        tipo = Xtension(fichero)
        Dim partes As Object
        partes = Split(fichero, ".")
        Dim extension As String
        extension = partes(UBound(partes))
        agregar = False
        If cbImagenes.Value = True And tipo = 1 Then
            agregar = True
        ElseIf cbSonidos.Value = True And tipo = 2 Then
            agregar = True
        ElseIf cbTextos.Value = True And tipo = 3 Then
            agregar = True
        ElseIf cbTextos.Value = True And tipo = 4 Then
            agregar = True
        End If
        If agregar Then
            lbFiles.AddItem(directorio.path & "\" & fichero.Name)
        End If
    Next
    Dim Subdirectorio As Folder
    For Each Subdirectorio In directorio.SubFolders
        Call expande(Subdirectorio)
    Next
    Exit Sub
salir:
End Sub

Private Sub lbFiles_Click()
    On Error GoTo salir

    Dim ruta As String
    ruta = lbFiles.List(lbFiles.ListIndex)
    img.Visible = False
    acpdf1.Visible = False
    WB1.Visible = False
    wml.Visible = False

    wml.URL = ""
    wml.Controls.stop()
    wml.Visible = False
```

```

Dim tipo As Integer
tipo = Xtension(ruta)
Select Case tipo
    Case 1 ' Imagen
        img.Visible = True
        img.Picture = LoadPicture("")
        img.Picture = LoadPicture(ruta)
    Case 2 ' Audio / Video
        wml.URL = ruta
        wml.Height = ALTO
        wml.Width = ANCHO
        wml.Visible = True
    Case 3 ' pdf
        acpdf1.Visible = True
        acpdf1.src = ruta
        acpdf1.gotoFirstPage()
        acpdf1.SetFocus()
    Case 4 ' html / txt / xml
        WB1.Visible = True
        WB1.Navigate(ruta)
    Case Else ' Otros
End Select
lbFiles.SetFocus()
Exit Sub

salir:
End Sub

Function Xtension(ByVal ruta As String) As Integer

    Dim partes As Object
    partes = Split(ruta, ".")
    Dim extension As String
    extension = partes(UBound(partes))
    If extension = "jpg" Or extension = "gif" Or extension = "bmp"
Then
        Xtension = 1 ' Imagen
    ElseIf extension = "mp3" Or extension = "wav" Or extension =
"avi" Or extension = "mpg" Then
        Xtension = 2 ' Audio/Video
    ElseIf extension = "pdf" Then
        Xtension = 3 ' pdf
    ElseIf extension = "html" Or extension = "htm" Or extension =
"txt" Or extension = "xml" Then
        Xtension = 4 ' html / txt / xml
    Else
        Xtension = 0 'Otras
    End If
End Function

Private Sub opClip_Click()
    If lbFiles.ListIndex <> -1 Then
        Select Case Xtension(lbFiles.List(lbFiles.ListIndex))
            Case 1 ' Imagen

```



```
        img.PictureSizeMode = fmPictureSizeModeClip
    Case 2 ' Audio / Video
        wml.stretchToFit = True
    Case Else ' Otros
    End Select
End If
End Sub

Private Sub opStrech_Click()
    If lbFiles.ListIndex <> -1 Then
        Select Case Xtension(lbFiles.List(lbFiles.ListIndex))
            Case 1 ' Imagen
                img.PictureSizeMode = fmPictureSizeModeStretch
            Case 2 ' Audio / Video
                wml.stretchToFit = True
            Case Else ' Otros
        End Select
    End If
End Sub

Private Sub opZoom_Click()
    If lbFiles.ListIndex <> -1 Then
        Select Case Xtension(lbFiles.List(lbFiles.ListIndex))
            Case 1 ' Imagen
                img.PictureSizeMode = fmPictureSizeModeZoom
            Case 2 ' Audio / Video
                wml.stretchToFit = True
            Case Else ' Otros
        End Select
    End If
End Sub

Private Sub UserForm_Initialize()
    tbPathIni.Text = "C:\Users\Juanto\Pictures"
    tbPathIni.Text = "C:\Temp"
    cbImagenes.Value = True
    cbSonidos.Value = True
    cbTextos.Value = False
    btnBuscar.Default = True
    img.Visible = False
    wml.Visible = False
    acpdf1.Visible = False
    opClip.Value = True
End Sub
```

Ra-Ma®

Para relanzar la llamada desde la opción **Macros** de **MS Excel**, bastará con insertar un módulo y crear el siguiente método:

```
Sub lanza_buscador()  
    frmBuscaArchivos.Show()  
End Sub
```

### 6.5.6 Valoración equipo de trabajo

El siguiente ejemplo pretende ser una muestra de como calcular una valoración general de cada uno de los recursos definidos en la hoja **Recursos\_AAAAMM** siendo **AAAA** y **MM** el año y mes respectivamente que se desea valorar. Por una parte, el formulario se apoya en una hoja denominada '**Tablas**' la cual, posee cada uno de los posibles valores que pueden seleccionarse para cada una de las características valoradas en cada recurso. Por otra parte, cada valoración mensual se almacenará en una hoja para ir engrosando el libro mes a mes.

La idea es que, cada mes, hemos de valorar para cada recurso cada una de las siguientes características: puntualidad, eficiencia en horas extras, calidad del trabajo, velocidad de ejecución del trabajo, autonomía, trato a los usuarios y solidaridad con el equipo.

Cada aspecto a valorar posee un juego de posibles valores los cuales, tienen un peso determinado que, al multiplicarlo por un coeficiente asociado a cada característica, configura un valor que sumado al resto de valores del resto de características, aportará un total que permitirá encasillar la valoración para determinar si el recurso es alguno de los siguientes valores: Poco eficiente, Normal, Eficiente y Muy Eficiente.

Este supuesto es arbitrario y el único propósito que persigue es el de tener una excusa para implementar listas de valores y otros objetos para seguir abundando en el uso de controles para la confección de formularios.

La hoja Tablas posee los siguientes datos:

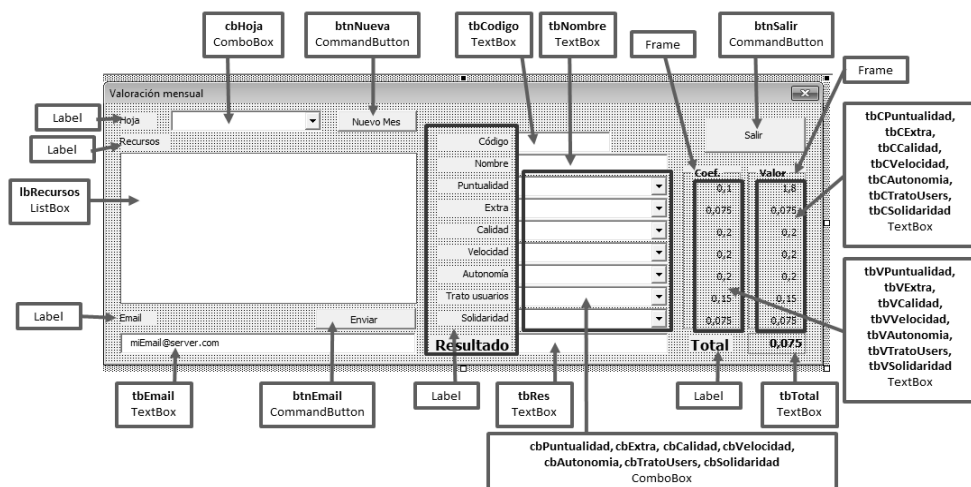
	A	B	C	D	E	F	G
1	Puntualidad	Extra	Calidad	Velocidad	Autonomía	Trato_usuarios	Solidaridad
2	Impuntual	Poco eficiente	Poca calidad	Lento	Poco autónomo	Malo	No es solidario
3	Generalmente puntual	Normal	Normal	Normal	Generalmente autónomo	Normal	Normal
4	Siempre puntual	Eficiente	Mucha calidad	Rápido	Autónomo	Bueno	Solidario
5		Muy eficiente	Perfectos	Muy rápido		Muy bueno	Muy solidario
6							
7							
8							
9	<b>Coefficientes</b>						
10	Puntualidad	Extra	Calidad	Velocidad	Autonomía	Trato_usuarios	Solidaridad
11	0,1	0,075	0,2	0,2	0,2	0,15	0,075
12							
13							
14							
15	Poco eficiente	Normal	Eficiente	Muy eficiente			
16	0	2,1	2,65	3,15			
17							
18	Email envío	miEmail@server.com					

Las hojas de resultados, almacenarán las valoraciones de cada mes y tendrán la siguiente configuración:

	A	B	C	D	E	F	G	H	I	J
1	Código	Recurso	Valoracion	Puntualidad	Extra	Calidad	Velocidad	Autonomía	Trato_usuarios	Solidaridad
2	12345678	Juan Antonio Gómez	Muy eficiente	Siempre puntual	Muy eficiente	Perfectos	Muy rápido	Autónomo	Muy bueno	Muy solidario
3	01010101	Carmen Nolosé	Muy eficiente	Siempre puntual	Muy eficiente	Perfectos	Muy rápido	Autónomo	Muy bueno	Muy solidario
4	02020202	Jordi Tintoré	Poco eficiente	Impuntual	Normal	Poca calidad	Rápido	Poco autónomo	Malo	Solidario
5	03030303	Xavi Barrés	Eficiente	Siempre puntual	Muy eficiente	Normal	Muy rápido	Generalmente autónomo	Bueno	Muy solidario

### 6.5.6.1 ELEMENTOS DEL FORMULARIO

El formulario (denominado **frmVal**) contiene los siguientes elementos:



### 6.5.6.2 EJECUCIÓN DEL FORMULARIO

En ejecución, el formulario (**frmVal**) tiene el siguiente aspecto:


Código	Recurso	Valoración
12345678	Juan Antonio Gómez	Normal
01010101	Carmen Nolosé	Muy eficiente
02020202	Jordi Tintoré	Eficiente
03030303	Xavi Barrés	Eficiente

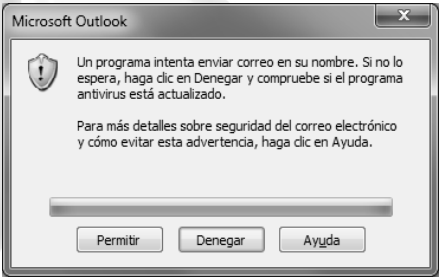

Coef.	Valor
0,1	0,3
0,075	0,3
0,2	0,8
0,2	0,8
0,2	0,6
0,15	0,6
0,075	0,3

**Resultado** *Muy eficiente* **Total** *3,7*

En principio, se muestra la última hoja existente en el libro suponiendo que es la que hemos de tratar por ser la última que se ha creado. Cada vez que seleccionemos una hoja, los datos se actualizarán en la ventana y, cada vez que seleccionemos un recurso, se mostrarán los valores de cada aspecto a valorar y su resultado total. En función del resultado total, se mostrará un color u otro de manera que el verde representaría un **Muy eficiente** y el rojo para **Poco eficiente** dejando el amarillo para **Normal** y el azul para **Eficiente**. Permite crear un libro con la hoja en curso y enviarlo por email a la dirección indicada en la caja de texto **Email**.

La funcionalidad prevista para cada elemento es la siguiente:

Objeto	Comentario
Lista desplegable <b>Hoja</b>	Permite seleccionar la hoja de recursos que se desea tratar. 
Botón <b>Nuevo Mes</b>	Permite crear una nueva hoja de recursos usando el año de la anterior y sumando 1 al mes. Si se trata del mes 12, se suma 1 al año y se empieza con el mes 1.

Caja de texto <b>Email</b>	Propone la dirección email
Botón <b>Enviar</b>	<p>Fabrica un libro nuevo con el mismo nombre que la hoja de resultados y lo envía por email. Es posible que reciba un mensaje de advertencia indicando que se va a enviar un email y tendrá que escoger entre <b>Permitir</b> o <b>Denegar</b> su envío.</p> <div></div> <p>No se depura el formato de la dirección de email pero si se verifica que éste no esté vacío. Si usamos nuestra dirección de correo y hacemos clic sobre este botón, al consultar el correo veremos un mensaje con el libro adjunto.</p> <div><p>Recursos_201212.xlsx</p><p>Tamaño: 8 KB</p><div></div></div> <p>Una vez visualizado el correo, se solicitará confirmación de recepción del mismo.</p>
Cajas de texto <b>Código y Nombre</b>	Muestra el código y el nombre del recurso seleccionado en la lista de recursos.

Lista de <b>Recursos</b>	Muestra el código, el nombre y la valoración de cada uno de los recursos existentes en la hoja Recursos que se está tratando.
Listas desplegables <b>Características</b>	Se muestra el valor que el recurso seleccionado posee para cada una de las características previstas.
Cajas de texto de <b>Coefficientes</b>	Muestra el coeficiente asignado a cada característica.
Cajas de texto de <b>Valores</b>	Muestra el valor que supone multiplicar el coeficiente de la característica, por el valor seleccionado para la misma.
Caja de texto <b>Total</b>	Valor numérico que representa la suma de los valores de todas las características consideradas.
Caja de texto <b>Resultado</b>	Muestra el resultado asociado al valor total resultante de la suma de todos los valores.  El fondo de esta caja variará en función del resultado tal y como se ha comentado anteriormente (verde, azul, amarillo y rojo según la eficiencia calculada).
Botón <b>Salir</b>	Cierra el formulario.

### 6.5.6.3 CÓDIGO FUENTE DEL FORMULARIO

Hay que marcar la referencia **Microsoft Scripting Runtime** a través de la opción **Herramientas->Referencias** del menú del **Editor de Visual Basic**.

El código correspondiente al formulario (**frmVal**) es el siguiente:

```
Dim nreg As Integer
Dim hRes As Worksheet

Private Sub btnEmail_Click()
    On Error GoTo error
    If tbEmail.Text = "" Then
        MsgBox("Ha de introducir una dirección de email válida.")
        Exit Sub
    End If
```

```

Dim appExcel As Excel.Application
appExcel = New Excel.Application

Dim fichero As File
Dim Fso As FileSystemObject
Dim Destino As Workbook
Application.DisplayAlerts = False
Application.ScreenUpdating = False
Application.EnableEvents = False
Dim strHoja, strRuta, filename As String
strHoja = ActiveCell.Worksheet.Name
strRuta = ActiveWorkbook.path
filename = strRuta & "\" & strHoja & ".xlsx"
Fso = New FileSystemObject
If Fso.FileExists(filename) Then
    Destino = appExcel.Workbooks.Open(filename)
Else
    Dim nhogas As Integer
    nhogas = appExcel.SheetsInNewWorkbook
    appExcel.SheetsInNewWorkbook = 1
    Destino = appExcel.Workbooks.Add()
    Destino.Saved = True
    Destino.SaveAs(filename)
    appExcel.SheetsInNewWorkbook = nhogas
End If
Destino.Activate()
Application.EnableEvents = True
ThisWorkbook.Activate()
Dim hd As Worksheet
hd = Destino.Worksheets(1)
Dim i, j As Integer
For i = 1 To hRes.UsedRange.Cells.Rows.Count
    For j = 1 To 3
        hd.Cells(i, j).Value = hRes.Cells(i, j).Value
        If i = 1 Then
            hd.Cells(i, j).Style = "Énfasis1"
        End If
    Next j
Next i
hd.Columns.AutoFit()
Destino.Worksheets(1).Name = strHoja
Destino.Saved = True
Destino.Save()
Destino.SendMail(Recipients:=tbEmail.Text, Subject:="Valoracion
mensual (" & strHoja & ")", returnreceipt:=True)
Destino.Close()
Destino = Nothing
Application.DisplayAlerts = True
Application.ScreenUpdating = True
appExcel = Nothing
Exit Sub
error:
MsgBox (Err.Description)

```

```
appExcel = Nothing
End Sub

Private Sub btnNueva_Click()
    Dim nueva As Worksheet

    hRes.Copy(after:=ActiveWorkbook.Worksheets(ActiveWorkbook.Worksheets.Count))
    nueva = ActiveWorkbook.Worksheets(ActiveWorkbook.Worksheets.Count)
    nueva.Name = busca_nombre
    nueva.Activate()
    carga_cbHoja()
    hRes = Worksheets(cbHoja.List(cbHoja.ListIndex))
End Sub

Function busca_nombre() As String
    Dim existe As Boolean
    existe = True
    Dim nombre, nombre_nuevo As String
    Dim partes As Object
    partes = Split(hRes.Name, "_")
    nombre = partes(UBound(partes))
    Dim i, anyo, mes As Integer
    While existe = True
        existe = False
        anyo = Mid(nombre, 1, 4)
        mes = Mid(nombre, 5, 2)
        mes = mes + 1
        If mes > 12 Then
            anyo = anyo + 1
            mes = 1
        End If
        nombre_nuevo = "Recursos_" & anyo & Format(mes, "00")
        For i = 1 To ActiveWorkbook.Worksheets.Count
            If ActiveWorkbook.Worksheets(i).Name = nombre_nuevo Then
                existe = True
            End If
        Next i
        If existe = True Then
            nombre = nombre_nuevo
        End If
    End While
    busca_nombre = nombre_nuevo
End Function

Private Sub btnSalir_Click()
    Unload(Me)
End Sub

Private Sub cbHoja_Change()
    If cbHoja.ListIndex <> -1 Then
        hRes = Worksheets(cbHoja.List(cbHoja.ListIndex))
        hRes.Activate()
    End If
End Sub
```



```
        If nreg > hRes.UsedRange.Cells.Rows.Count Then
            nreg = hRes.UsedRange.Cells.Rows.Count
        End If
        lbRecursos.RowSource = hRes.Range("A2:C" &
hRes.UsedRange.Cells.Rows.Count).Address
        If nreg > 0 Then
            mostrar_registro(nreg)
        End If
        lbRecursos.SetFocus()
    End If
End Sub

Private Sub cbPuntualidad_Change()
    hRes.Cells(nreg + 1, 4).Value = cbPuntualidad.Value
    tbVpuntualidad.Text = tbCpuntualidad.Text *
(cbPuntualidad.ListIndex + 1)
    suma()
End Sub

Private Sub cbExtra_Change()
    hRes.Cells(nreg + 1, 5).Value = cbExtra.Value
    tbVextra.Text = tbCextra.Text * (cbExtra.ListIndex + 1)
    suma()
End Sub

Private Sub cbCalidad_Change()
    hRes.Cells(nreg + 1, 6).Value = cbCalidad.Value
    tbVcalidad.Text = tbCcalidad.Text * (cbCalidad.ListIndex + 1)
    suma()
End Sub

Private Sub cbVelocidad_Change()
    hRes.Cells(nreg + 1, 7).Value = cbVelocidad.Value
    tbVvelocidad.Text = tbCvelocidad.Text * (cbVelocidad.ListIndex +
1)
    suma()
End Sub

Private Sub cbAutonomia_Change()
    hRes.Cells(nreg + 1, 8).Value = cbAutonomia.Value
    tbVautonomia.Text = tbCautonomia.Text * (cbAutonomia.ListIndex +
1)
    suma()
End Sub

Private Sub cbTratoUsers_Change()
    hRes.Cells(nreg + 1, 9).Value = cbTratoUsers.Value
    tbVtrato.Text = tbCtrato.Text * (cbTratoUsers.ListIndex + 1)
    suma()
End Sub

Private Sub cbSolidaridad_Change()
    hRes.Cells(nreg + 1, 10).Value = cbSolidaridad.Value
```

```
tbVsolidaridad.Text = tbCsolidaridad.Text *  
(cbSolidaridad.ListIndex + 1)  
    suma()  
End Sub  
  
Private Sub lbRecursos_Click()  
    nreg = lbRecursos.ListIndex + 1  
    mostrar_registro(nreg)  
End Sub  
  
Private Sub UserForm_Initialize()  
    carga_cbHoja()  
    hRes = Worksheets(cbHoja.List(cbHoja.ListIndex))  
    cbPuntualidad.RowSource = "Tablas!A2:A4"  
    cbExtra.RowSource = "Tablas!B2:B5"  
    cbCalidad.RowSource = "Tablas!C2:C5"  
    cbVelocidad.RowSource = "Tablas!D2:D5"  
    cbAutonomia.RowSource = "Tablas!E2:E4"  
    cbTratoUsers.RowSource = "Tablas!F2:F5"  
    cbSolidaridad.RowSource = "Tablas!G2:G5"  
    Dim f As Integer  
    lbRecursos.RowSource = hRes.Range("A2:C" &  
hRes.UsedRange.Cells.Rows.Count).Address  
    lbRecursos.ColumnWidths = "50;100;90"  
    lbRecursos.Selected(0) = True  
    nreg = 1  
    mostrar_registro(nreg)  
    lbRecursos.SetFocus()  
End Sub  
  
Sub carga_cbHoja()  
    cbHoja.Clear()  
    Dim h As Worksheet  
    For Each h In ActiveWorkbook.Worksheets  
        If h.Name <> "Tablas" Then  
            cbHoja.AddItem(h.Name)  
        End If  
    Next  
    cbHoja.ListIndex = cbHoja.ListCount - 1  
End Sub  
  
Sub mostrar_registro(nreg As Integer)  
    tbCodigo.Text = hRes.UsedRange.Cells(nreg + 1, 1).Value  
    tbNombre.Text = hRes.UsedRange.Cells(nreg + 1, 2).Value  
    cbPuntualidad.Value = hRes.UsedRange.Cells(nreg + 1, 4).Value  
    cbExtra.Value = hRes.UsedRange.Cells(nreg + 1, 5).Value  
    cbCalidad.Value = hRes.UsedRange.Cells(nreg + 1, 6).Value  
    cbVelocidad.Value = hRes.UsedRange.Cells(nreg + 1, 7).Value  
    cbAutonomia.Value = hRes.UsedRange.Cells(nreg + 1, 8).Value  
    cbTratoUsers.Value = hRes.UsedRange.Cells(nreg + 1, 9).Value  
    cbSolidaridad.Value = hRes.UsedRange.Cells(nreg + 1, 10).Value  
    suma()  
End Sub
```

```
Sub suma()  
    tbTotal.Text = CDb1(tbVpuntualidad.Text) + CDb1(tbVextra.Text) +  
    _  
    CDb1(tbVcalidad.Text) + CDb1(tbVvelocidad.Text) + _  
    CDb1(tbVautonomia.Text) + CDb1(tbVtrato.Text) + _  
    CDb1(tbVsolidaridad.Text)  
    clasificar()  
End Sub  
  
Sub clasificar()  
    Dim res As Integer  
    Dim color As Long  
    Dim valor As Double  
    valor = CDb1(tbTotal.Text)  
    If valor < Worksheets("Tablas").Cells(16, 2).Value Then  
        res = 1  
        color = vbRed  
    ElseIf valor < Worksheets("Tablas").Cells(16, 3).Value Then  
        res = 2  
        color = vbYellow  
    ElseIf valor < Worksheets("Tablas").Cells(16, 4).Value Then  
        res = 3  
        color = vbHighlight  
    Else  
        res = 4  
        color = vbGreen  
    End If  
    tbRes.Text = Worksheets("Tablas").Cells(15, res).Value  
    hRes.UsedRange.Cells(nreg + 1, 3).Value = tbRes.Text  
    tbRes.BackColor = color  
End Sub
```

Para relanzar la llamada desde la opción **Macros de MS Excel**, bastará con insertar un módulo y crear el siguiente método:

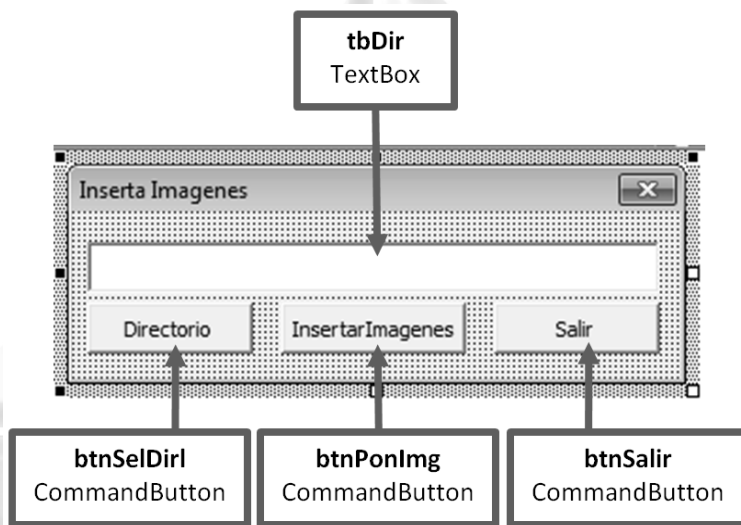
```
Sub lanza_val()  
    frmVal.Show()  
End Sub
```

### 6.5.7 Insertar imágenes desde directorio

Este ejemplo permite insertar en una hoja de cálculo todas las fotos que se hallen en un determinado directorio acompañadas de su nombre.

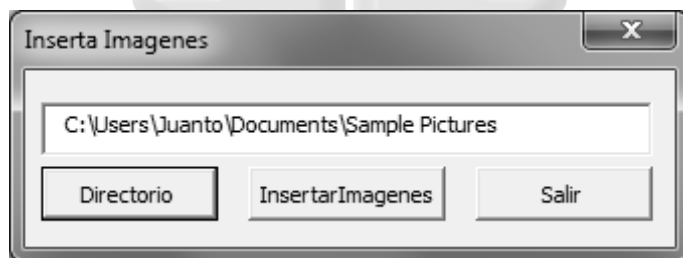
### 6.5.7.1 ELEMENTOS DEL FORMULARIO

El formulario (denominado **frmInsImg**) contiene los siguientes elementos:

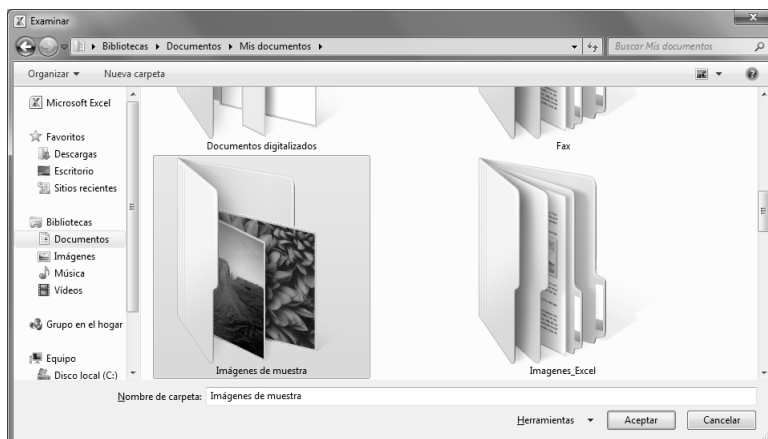


### 6.5.7.2 EJECUCIÓN DEL FORMULARIO

Al ejecutar el formulario, vemos lo siguiente:

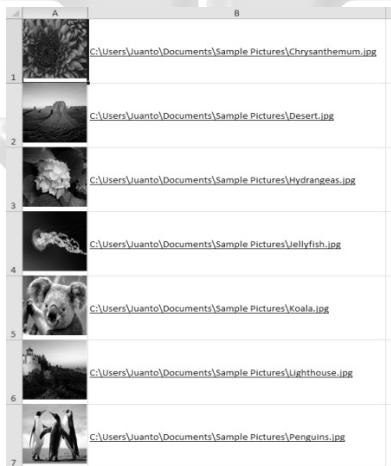


En primer lugar, hemos de seleccionar el directorio que contiene las imágenes que queremos insertar en la hoja que se halle activa en este momento. Para ello, al pulsar sobre el botón directorio, se abrirá el cuadro de diálogo **Examinar** para que podamos navegar hasta la carpeta que nos interese:



Una vez localizada la imagen, pulsaremos en **Aceptar** para cerrar el cuadro de diálogo **Examinar** y regresar al formulario. En este punto, sólo queda pulsar en el botón **Insertar Imagen** para empezar a insertar las imágenes del directorio en la hoja activa.

En este ejemplo, el resultado es el siguiente:



El tamaño con el que se insertan las imágenes está determinado por las constantes **ALTO** y **ANCHO** para facilitar un posible cambio del mismo.

Hay que marcar la referencia **Microsoft Scripting Runtime** a través de la opción **Herramientas->Referencias** del menú del **Editor de Visual Basic**.

La funcionalidad prevista para cada elemento es la siguiente:

Objeto	Comentario
Botón <b>Directorio</b>	Permite seleccionar el directorio que contiene las imágenes a insertar.
Botón <b>Insertar Imágenes</b>	Inserta las imágenes en la hoja activa.
Botón <b>Salir</b>	Cierra el formulario.

### 6.5.7.3 CÓDIGO FUENTE DEL FORMULARIO

El código correspondiente al formulario (**frmInsImg**) es el siguiente:

```
Option Explicit On
```

```
Const ALTO As Integer = 70
Const ANCHO As Integer = 70
Const MARGEN As Integer = 5
```

```
Private Sub btnPonImg_Click()
    On Error GoTo error
    Dim hd As Worksheet
    hd = ActiveSheet
    Dim agregar As Boolean
    Dim f, top_ant As Integer
    f = 0
    top_ant = 0
    Dim Fso As FileSystemObject
    Dim directorio As Folder
    Fso = New FileSystemObject
    directorio = Fso.GetFolder(tbDir.Text)
    Dim fichero As file
    Dim ruta As String
    For Each fichero In directorio.Files
        Dim partes As Object
        partes = Split(fichero, ".")
        Dim extension As String
        extension = LCase(partes(UBound(partes)))
        agregar = False
        If extension = "jpg" Or extension = "jpeg" Or extension =
"gif" Or extension = "tiff" Or extension = "bmp" Or extension =
"ico" Or extension = "png" Then
            agregar = True
        End If
        If agregar Then
            ruta = directorio.Path & "\" & fichero.Name
            f = f + 1
        End If
    Next fichero
End Sub
```

```

        hd.Cells(f, 1).Select()
        hd.Rows(f & ":" & f).RowHeight = ALTO + MARGEN
        hd.Columns("A").ColumnWidth = ANCHO / 5.5
        hd.Shapes.AddPicture(Filename:=ruta,
LinkToFile:=msoTrue, SaveWithDocument:=msoTrue, Left:=0,
Top:=top_ant, Width:=ANCHO, Height:=ALTO)
        top_ant = top_ant + ALTO + MARGEN
        hd.Cells(f, 2).Select()
        hd.Hyperlinks.Add(Anchor:=Selection, Address:=ruta, _
TextToDisplay:=ruta)
        With Selection
            .HorizontalAlignment = xlGeneral
            .VerticalAlignment = xlCenter
            .WrapText = False
            .Orientation = 0
            .AddIndent = False
            .IndentLevel = 0
            .ShrinkToFit = False
            .ReadingOrder = xlContext
            .MergeCells = False
        End With
    End If
Next
hd.Range("A:B").Columns.AutoFit()
hd.Cells(1, 1).Select()
Exit Sub
error:
    MsgBox(Err.Description)
End Sub

Private Sub btnSalir_Click()
    Unload(Me)
End Sub

Private Sub btnSelDir_Click()
    With Application.FileDialog(msoFileDialogFolderPicker)
        .InitialFileName = tbDir.Text
        .Show()
        If .SelectedItems.Count > 0 Then
            tbDir.Text = .SelectedItems(1)
        End If
        If .SelectedItems.Count > 0 Then
            Debug.Print.SelectedItems(1)
        End If
    End With
End Sub

Private Sub UserForm_Initialize()
    Dim WS As Object
    WS = CreateObject("WScript.Shell")
    tbDir.Text = WS.SpecialFolders("MyDocuments")
    WS = Nothing
End Sub

```

Para relanzar la llamada desde la opción **Macros** de **MS Excel**, bastará con insertar un módulo y crear el siguiente método:

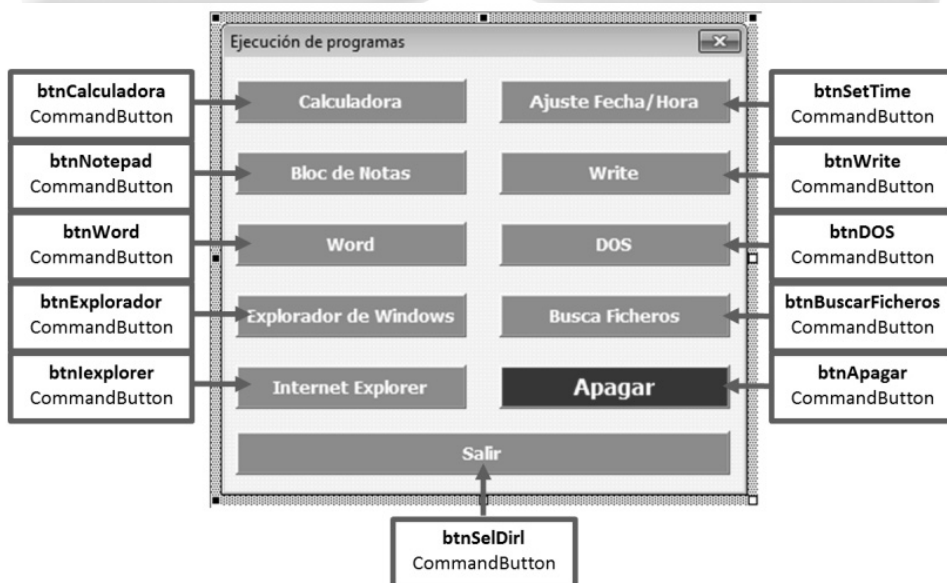
```
Sub lanza_form()  
    frmInsImg.Show()  
End Sub
```

## 6.5.8 Ejecutar aplicaciones o programas

El siguiente ejemplo muestra cómo se pueden ejecutar aplicaciones Windows desde un formulario. Fundamentalmente utilizaremos las instrucciones `Shell` y `CreateObject` (comentadas en el capítulo Funciones VBA).

### 6.5.8.1 ELEMENTOS DEL FORMULARIO

El formulario (denominado **frmInsImg**) contiene los siguientes elementos:

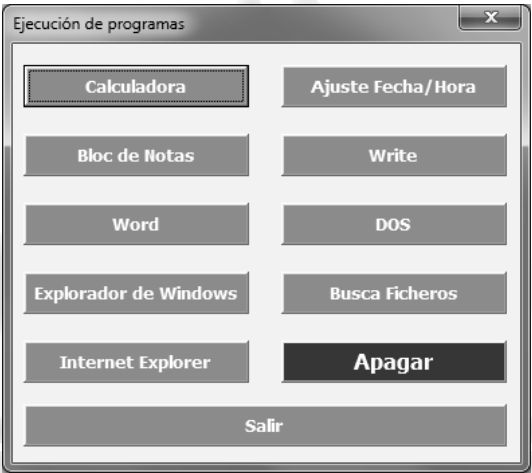


Ra-Ma<sup>®</sup>


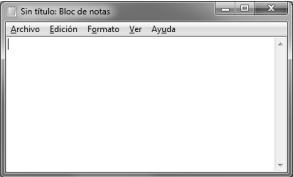


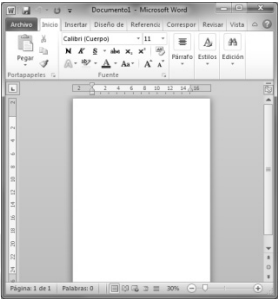
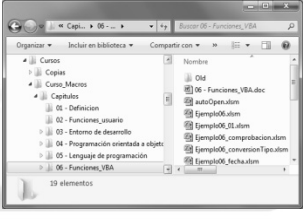
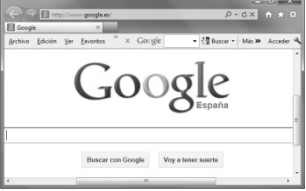


6.5.8.2 EJECUCIÓN DEL FORMULARIO

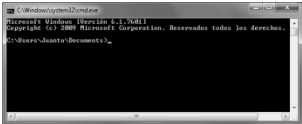
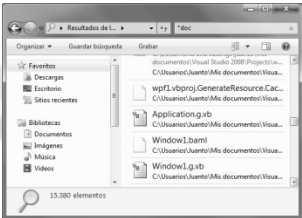
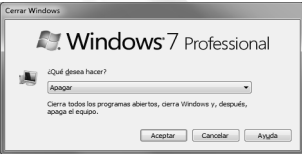
En ejecución, el formulario (**frmEjecutar**) tiene el siguiente aspecto:



La funcionalidad prevista para cada elemento es la siguiente:

Objeto	Comentario	
Botón <b>Calculadora</b>	Abre la calculadora.	
Botón <b>Bloc de Notas</b>	Abre el Notepad.	

<p><b>Botón Word</b></p>	<p>Abre <b>Microsoft Windows</b>.</p>	
<p><b>Botón Explorador de Windows</b></p>	<p>Abre el <b>explorador de Windows</b>.</p>	
<p><b>Botón Internet Explorer</b></p>	<p>Abre el navegador <b>Internet Explorer</b>.</p>	
<p><b>Botón Ajuste Fecha/Hora</b></p>	<p>Abre el cuadro de diálogo que nos permite cambiar la <b>fecha y hora</b> del sistema.</p>	
<p><b>Botón Write</b></p>	<p>Abre el editor de texto <b>Write</b>.</p>	

Botón <b>DOS</b>	Abre Ventana DOS.	
Botón <b>Busca Ficheros</b>	Abre ventana que permite buscar ficheros	
Botón <b>Apagar</b>	Invoca al cuadro de diálogo que permite apagar el sistema.	

### 6.5.8.3 CÓDIGO FUENTE DEL FORMULARIO

El código correspondiente al formulario (**frmEjecutar**) es el siguiente:

```

Private Sub btnApagar_Click()
    With CreateObject("shell.application")
        .ShutDownWindows()
    End With
End Sub

Private Sub btnBuscarFicheros_Click()
    Dim s As Object
    s = CreateObject("shell.application")
    s.FindFiles()
End Sub

Private Sub btnCalculadora_Click()
    Shell(pathname:="Calc.exe",
    windowstyle:=VbAppWinStyle.vbNormalFocus)
End Sub

Private Sub btnDOS_Click()
    Shell(pathname:="cmd.exe",
    windowstyle:=VbAppWinStyle.vbNormalFocus)
End Sub

Private Sub btnExplorador_Click()
    Shell(pathname:="explorer.exe C:\Temp",
    windowstyle:=VbAppWinStyle.vbMaximizedFocus)

```

```
End Sub

Private Sub btnIexplorer_Click()
    Shell(pathname:="C:\Archivos de programa\Internet
Explorer\iexplore.exe",
windowstyle:=VbAppWinStyle.vbMinimizedFocus)
End Sub

Private Sub btnNotepad_Click()
    Shell(pathname:="Notepad.exe",
windowstyle:=VbAppWinStyle.vbNormalFocus)
End Sub

Private Sub btnSetTime_Click()
    With CreateObject("shell.application")
        .SetTime()
    End With
End Sub

Private Sub btnWord_Click()
    Shell(pathname:="Winword.exe",
windowstyle:=VbAppWinStyle.vbMaximizedFocus)
End Sub

Private Sub btnSalir_Click()
    Unload(Me)
End Sub

Private Sub btnWrite_Click()
    Shell(pathname:="Write.exe",
windowstyle:=VbAppWinStyle.vbNormalFocus)
End Sub
```

Para relanzar la llamada desde la opción **Macros de MS Excel**, bastará con insertar un módulo y crear el siguiente método:

```
Sub lanza_ejecutar()
    frmEjecutar.Show()
End Sub
```

## 6.5.9 Menús

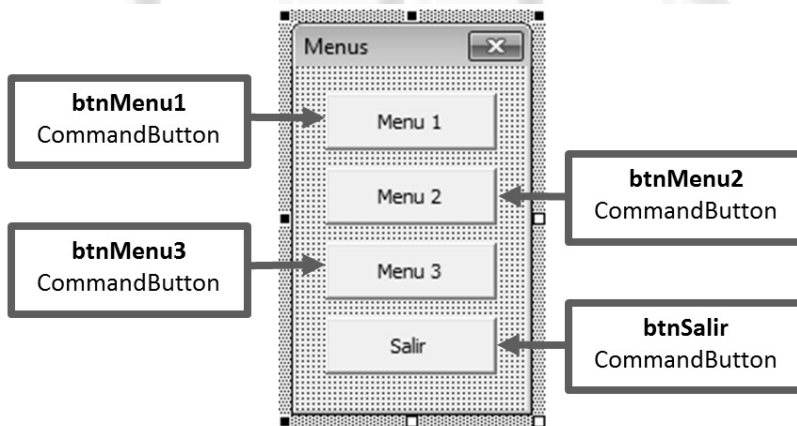
Los menús no existen como objetos que podamos incorporar en los formularios simplemente arrastrando un control y ubicándolo sobre el mismo.

El tratamiento de menús en VBA es algo más complejo o se tiene que simular de alguna manera.

En el siguiente ejemplo se utiliza un formulario que muestra 3 posibles formas de implementar un menú. El primer método, fabrica un menú flotante el cual se crea y destruye cada vez que se invoca. El método 2 simula el funcionamiento de un menú el cual visualiza y oculta opciones según se van necesitando. El tercer método, simplemente muestra un panel que se puede recorrer con una barra de desplazamiento y que contiene diversas opciones de un menú.

### 6.5.9.1 ELEMENTOS DEL FORMULARIO

El formulario principal (denominado **frmMenuPral**) contiene los siguientes elementos:



### 6.5.9.2 EJECUCIÓN DEL FORMULARIO

En ejecución, el formulario (**frmMenuPral**) tiene el siguiente aspecto:



La funcionalidad prevista para cada elemento es la siguiente:

Objeto	Comentario
Botón <b>Menu 1</b>	Crea menú flotante.
Botón <b>Menu 2</b>	Invoca al primer ejemplo de menús.
Botón <b>Menu 3</b>	Invoca al segundo ejemplo de menús.

### 6.5.9.3 CÓDIGO FUENTE DEL FORMULARIO

El código correspondiente al formulario (**frmMenuPral**) es el siguiente:

```
Option Explicit
```

```
Private Sub btnMenu1_Click()  
    Call mod01.CrearMenu()  
End Sub
```

```
Private Sub btnMenu2_Click()  
    frmMenus2.Show()  
End Sub
```

```
Private Sub btnMenu3_Click()  
    frmMenus3.Show()  
End Sub
```

```
Private Sub btnSalir_Click()  
    Unload(Me)  
End Sub
```

A continuación explicaremos cada uno de los ejemplos que son invocados desde este menú principal.

### 6.5.9.4 CÓDIGO FUENTE DEL MENÚ 1. MENÚ FLOTANTE

En primer lugar, hemos de agregar un módulo a través del menú **Insertar->Módulo**. Dicho módulo, lo denominaremos **mod01**.

En este módulo, crearemos los métodos necesarios para crear, mostrar y borrar el menú flotante.

El código a situar en el módulo será el siguiente:

```
Option Explicit On

Public Const nomMenu As String = "MiMenu"

Sub BorrarMenu()
    ' Borra el menú por si éste existe previamente
    On Error Resume Next
    Application.CommandBars(nomMenu).Delete()
    On Error GoTo 0
End Sub

Sub CrearMenu()
    ' Borra el menú.
    Call BorrarMenu()

    ' Crea el menú.
    Call MenuPersonalizado()

    ' Muestra el menú.
    On Error Resume Next
    Application.CommandBars(nomMenu).ShowPopup()
    On Error GoTo 0
End Sub

Sub MenuPersonalizado()
    Dim MenuItem As CommandBarPopup
    With Application.CommandBars.Add(Name:=nomMenu,
    Position:=msoBarPopup, _
        MenuBar:=False, Temporary:=True)

        With .Controls.Add(Type:=msoControlButton)
            .Caption = "Calculadora"
            .FaceId = 71
            .OnAction = "" & ThisWorkbook.Name & "!" &
"lanza_calculadora"
        End With

        With .Controls.Add(Type:=msoControlButton)
            .Caption = "Saludo"
            .FaceId = 72
            .OnAction = "" & ThisWorkbook.Name & "!" & "saludo"
        End With

        MenuItem = .Controls.Add(Type:=msoControlPopup)
        With MenuItem
            .Caption = "Formularios"

            With .Controls.Add(Type:=msoControlButton)
                .Caption = "Menús de tipo 2"
                .FaceId = 71
            End With
        End With
    End With
End Sub
```

```

        .OnAction = "" & ThisWorkbook.Name & "!" &
"lanza_menus2"
        End With

        With .Controls.Add(Type:=msoControlButton)
            .Caption = "Menús de tipo 3"
            .FaceId = 72
            .OnAction = "" & ThisWorkbook.Name & "!" &
"lanza_menus3"
        End With
        MenuItem = .Controls.Add(Type:=msoControlPopup)
        With MenuItem
            .Caption = "Mas submenus"

            With .Controls.Add(Type:=msoControlButton)
                .Caption = "Calculadora"
                .FaceId = 71
                .OnAction = "" & ThisWorkbook.Name & "!" &
"lanza_calculadora"
            End With

            With .Controls.Add(Type:=msoControlButton)
                .Caption = "Notepad"
                .FaceId = 72
                .OnAction = "" & ThisWorkbook.Name & "!" &
"lanza_notepad"
            End With
            With .Controls.Add(Type:=msoControlButton)
                .Caption = "Explorador de Windows"
                .FaceId = 73
                .OnAction = "" & ThisWorkbook.Name & "!" &
"lanza_explorer"
            End With
        End With

        End With
        With .Controls.Add(Type:=msoControlButton)
            .Caption = "Salir"
            .FaceId = 73
            .OnAction = "" & ThisWorkbook.Name & "!" &
"lanza_salida"
        End With
    End With
End Sub

Sub TestMacro()
    MsgBox("Hi there! Greetings from the Netherlands.")
End Sub

Sub saludo()
    MsgBox("Hola Mundo!!")
End Sub

```

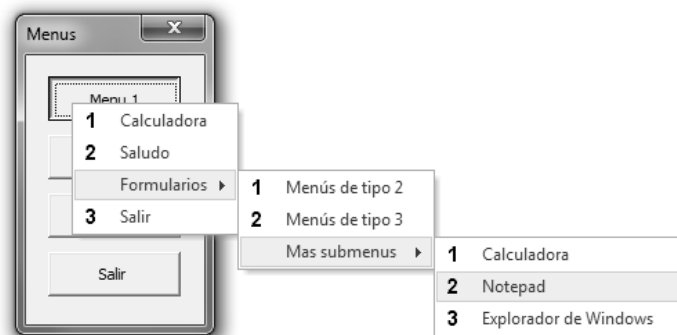


```
Sub lanza_calculadora()  
    Shell("calc.exe")  
End Sub  
  
Sub lanza_explorer()  
    Shell("explorer.exe")  
End Sub  
  
Sub lanza_notepad()  
    Shell("Notepad.exe", windowstyle:=VbAppWinStyle.vbNormalFocus)  
End Sub  
  
Sub lanza_menus2()  
    frmMenus2.Show()  
End Sub  
  
Sub lanza_menus3()  
    frmMenus3.Show()  
End Sub  
  
Sub lanza_salida()  
    Unload(frmMenuPral)  
End Sub
```

Para la invocación de este menú, hemos visto que en el evento clic del botón **btnMenu1** del formulario principal (**frmMenuPral**), hemos colocado el siguiente código:

```
Private Sub btnMenu1_Click()  
    Call mod01.CrearMenu()  
End Sub
```

Al hacer clic sobre el botón cuya etiqueta indica 'Menu 1', vemos que aparece un menú flotante el cual podemos ir expandiendo tal y como se muestra en la siguiente imagen:

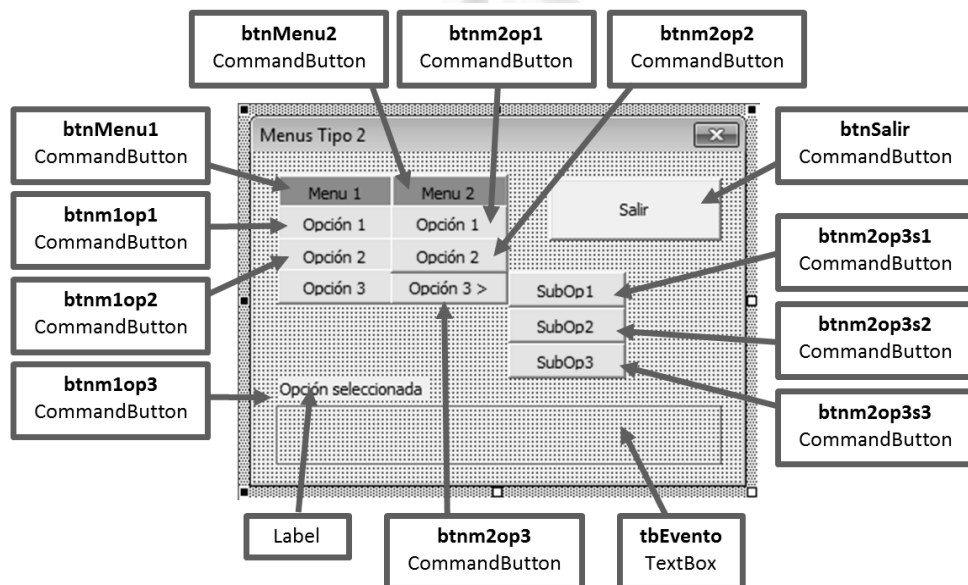


En función de la opción que seleccionemos, se ejecutará una cosa u otra. A modo de resumen, las acciones asociadas a cada opción son:

Opción	Acción
1 Calculadora	Ejecuta la calculadora.
2 Saludo	Muestra un cuadro de mensaje con un saludo.
Formularios	Despliega un menú de opciones.
Formularios – Menús de tipo 1	Invoca al segundo ejemplo de menús.
Formularios – Menús de tipo 2	Invoca al tercer ejemplo de menús.
Formularios – Mas submenús	Despliega un submenú de opciones.
Formularios – Mas submenús – 1 Calculadora	Ejecuta la calculadora.
Formularios – Mas submenús – 2 Notepad	Ejecuta el Bloc de notas.
Formularios – Mas submenús – 3 Explorador de Windows	Ejecuta el Explorador de Windows.
3 Salir	Cierra el formulario <sup>®</sup> principal ( <b>frmMenuPral</b> ).

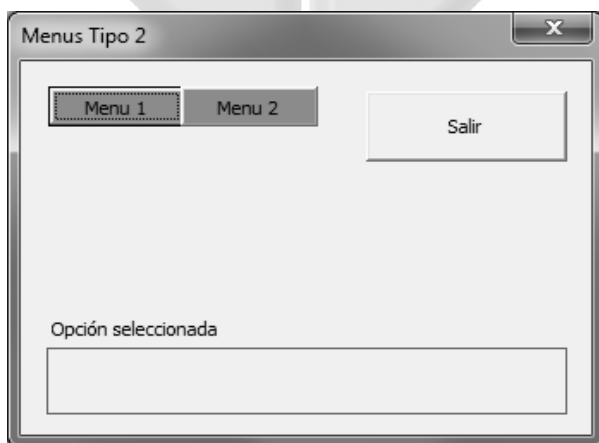
### 6.5.9.5 ELEMENTOS DEL FORMULARIO PARA MENU 2

El formulario para el menú 2 (denominado **frmMenus2**) contiene los siguientes elementos:



### 6.5.9.6 EJECUCIÓN DEL FORMULARIO PARA MENU 2

Al ejecutar el formulario, vemos lo siguiente:



En función del menú que seleccionemos, se despliegan unas opciones u otras. La idea es mantener los controles no visibles e ir haciendolos visibles según las acciones que se realicen. De esta forma, podemos conseguir el siguiente efecto:



#### 6.5.9.7 CÓDIGO FUENTE DEL FORMULARIO PARA MENU 2

El código asociado a este formulario es el siguiente:

```
Option Explicit On

Public ver_menu1, ver_menu2, ver_submenu23 As Boolean

Private Sub btnm1op1_Click()
    llena_tb(ActiveControl.Name)
    ocultar_submenu23()
    ver_menu1 = True
End Sub

Private Sub btnm1op2_Click()
    llena_tb(ActiveControl.Name)
    ocultar_submenu23()
    ver_menu1 = True
End Sub

Private Sub btnm1op3_Click()
    llena_tb(ActiveControl.Name)
    ocultar_submenu23()
    ver_menu1 = True
End Sub

Private Sub btnm2op1_Click()
    llena_tb(ActiveControl.Name)
    ocultar_submenu23()
    ver_menu2 = True
End Sub
```

```
Private Sub btnm2op2_Click()  
    llena_tb(ActiveControl.Name)  
    ocultar_submenu2()  
    ver_menu2 = False  
End Sub  
  
Private Sub btnm2op3_Click()  
    ver_submenu23 = Not (ver_submenu23)  
    btnm2op3s1.Visible = ver_submenu23  
    btnm2op3s2.Visible = ver_submenu23  
    btnm2op3s3.Visible = ver_submenu23  
End Sub  
  
Private Sub btnm2op3s1_Click()  
    llena_tb(ActiveControl.Name)  
    ocultar_submenu2()  
    ver_menu2 = False  
    ver_submenu23 = False  
End Sub  
Private Sub btnm2op3s2_Click()  
    llena_tb(ActiveControl.Name)  
    ocultar_submenu2()  
    ver_menu2 = False  
    ver_submenu23 = False  
End Sub  
  
Private Sub btnm2op3s3_Click()  
    llena_tb(ActiveControl.Name)  
    ocultar_submenu2()  
    ver_menu2 = False  
    ver_submenu23 = False  
End Sub  
  
Private Sub btnMenu1_Click()  
    ocultar_submenu2()  
    ver_menu1 = Not (ver_menu1)  
    ver_menu2 = False  
    btnm1op1.Visible = ver_menu1  
    btnm1op2.Visible = ver_menu1  
    btnm1op3.Visible = ver_menu1  
End Sub  
  
Private Sub btnMenu2_Click()  
    ocultar_submenu2()  
    ver_menu2 = Not (ver_menu2)  
    ver_menu1 = False  
    btnm2op1.Visible = ver_menu2  
    btnm2op2.Visible = ver_menu2  
    btnm2op3.Visible = ver_menu2  
End Sub
```

```

Sub ocultar_submenu()
    btnm1op1.Visible = False
    btnm1op2.Visible = False
    btnm1op3.Visible = False
    btnm2op1.Visible = False
    btnm2op2.Visible = False
    btnm2op3.Visible = False
    btnm2op3s1.Visible = False
    btnm2op3s2.Visible = False
    btnm2op3s3.Visible = False
End Sub

Private Sub btnSalir_Click()
    Unload(Me)
End Sub

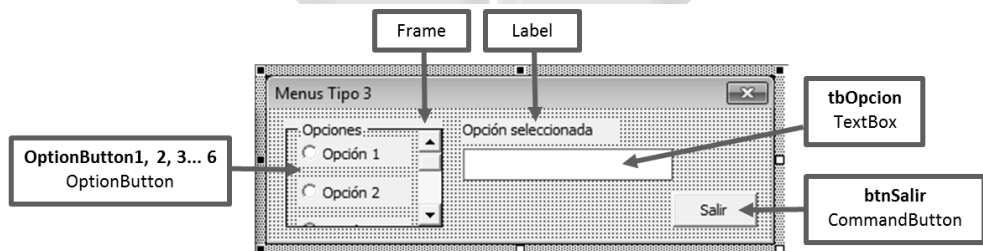
Sub llena_tb(n As String)
    tbEvento.Text = n
End Sub

Private Sub UserForm_Initialize()
    ver_menu1 = False
    ver_menu2 = False
    ver_submenu23 = False
End Sub

```

### 6.5.9.8 ELEMENTOS DEL FORMULARIO PARA MENU 3

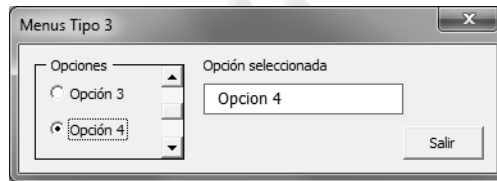
El formulario para el menú 3 (denominado **frmMenu3**) contiene los siguientes elementos:



Ra-Ma®

### 6.5.9.9 EJECUCIÓN DEL FORMULARIO PARA MENU 3

Este formulario, en ejecución presenta el siguiente aspecto:



Este ejemplo es muy simple y lo que se propone para organizar las opciones es un panel que muestre una zona determinada del mismo y disponga de una barra de desplazamiento que permita mostrar las opciones que contiene. En este caso, definimos la propiedad Height del Frame a 60 y su propiedad ScrollHeight a 150. Cada vez que se selecciona una opción, en el cuadro de texto se muestra la opción escogida.

### 6.5.9.10 CÓDIGO FUENTE DEL FORMULARIO PARA MENU 3

El código asociado a este formulario es el siguiente:

```
Private Sub btnSalir_Click()  
    Unload(Me)  
End Sub  
  
Private Sub OptionButton1_Click()  
    tbOpcion.Text = "Opcion 1"  
End Sub  
  
Private Sub OptionButton2_Click()  
    tbOpcion.Text = "Opcion 2"  
End Sub  
  
Private Sub OptionButton3_Click()  
    tbOpcion.Text = "Opcion 3"  
End Sub  
  
Private Sub OptionButton4_Click()  
    tbOpcion.Text = "Opción 4"  
End Sub  
  
Private Sub OptionButton5_Click()  
    tbOpcion.Text = "Opción 5"  
End Sub  
  
Private Sub OptionButton6_Click()  
    tbOpcion.Text = "Opcion 6"  
End Sub
```

Para relanzar la llamada desde la opción **Macros** de **MS Excel**, bastará con insertar un módulo y crear el siguiente método:

```
Sub lanza_ejecutar()  
    frmMenuPral.Show()  
End Sub
```

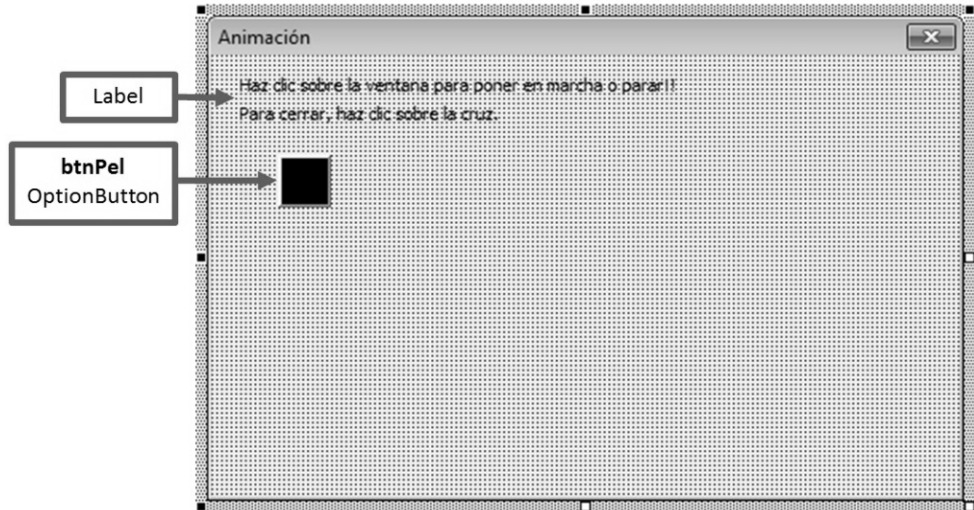
## 6.5.10 Animación

Este ejemplo muestra un botón que va cambiando de posición y va rebotando con los laterales de la ventana en la que se halla.

Se apoya en una ventana de control que permite cambiar algunas características como son el color del botón, la velocidad con la que se mueve y el tamaño de la ventana que lo contiene.

### 6.5.10.1 ELEMENTOS DEL FORMULARIO

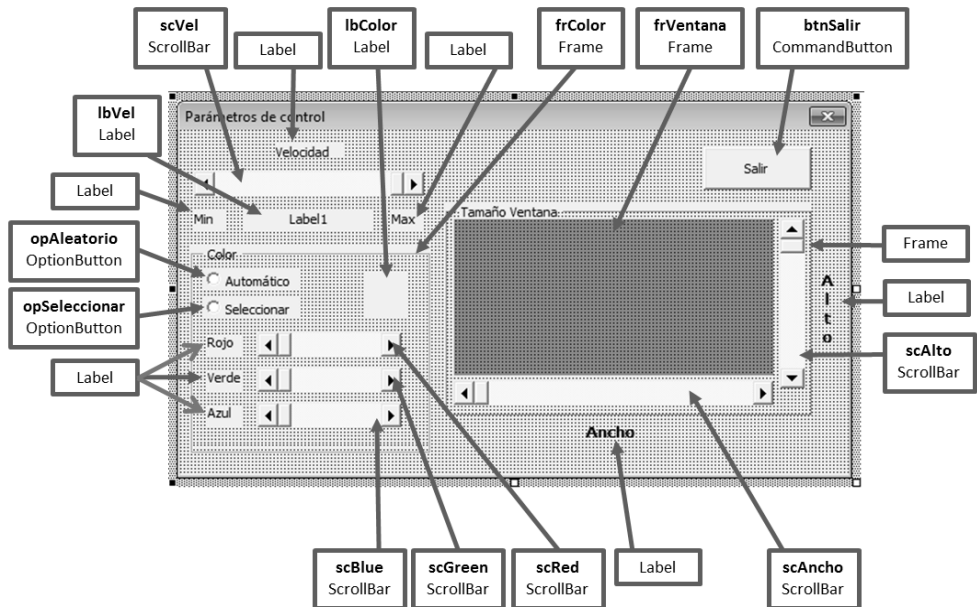
El formulario principal (denominado **frmAnimacion**) contiene los siguientes elementos:



Ra-Ma®

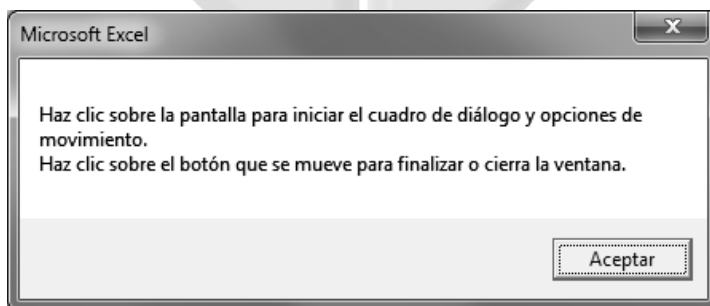


Al hacer clic sobre la pantalla, se muestra una ventana que permite cambiar ciertas características como hemos comentado anteriormente. Dicha ventana (**frmControl**) , posee los siguientes elementos:



### 6.5.10.2 EJECUCIÓN DEL FORMULARIO

Al ejecutar el formulario principal (**frmAnimacion**) el siguiente mensaje:



Ra-Ma<sup>®</sup>

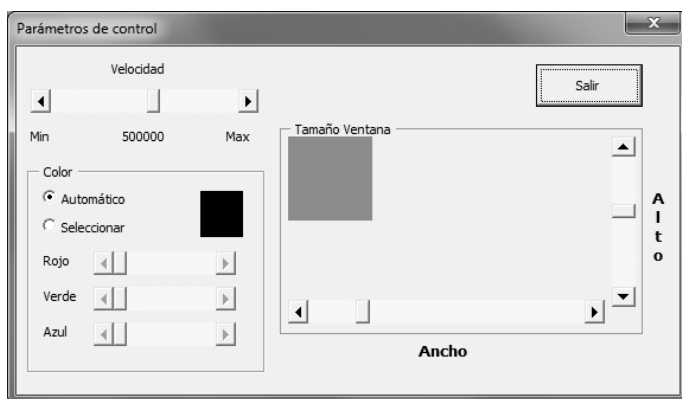
Una vez aceptado, aparece la siguiente ventana:



Si hacemos clic sobre el botón, aparece un mensaje despidiéndose del usuario y se cierra el formulario:



Si hacemos clic sobre cualquier parte del formulario, aparece la siguiente ventana de control:

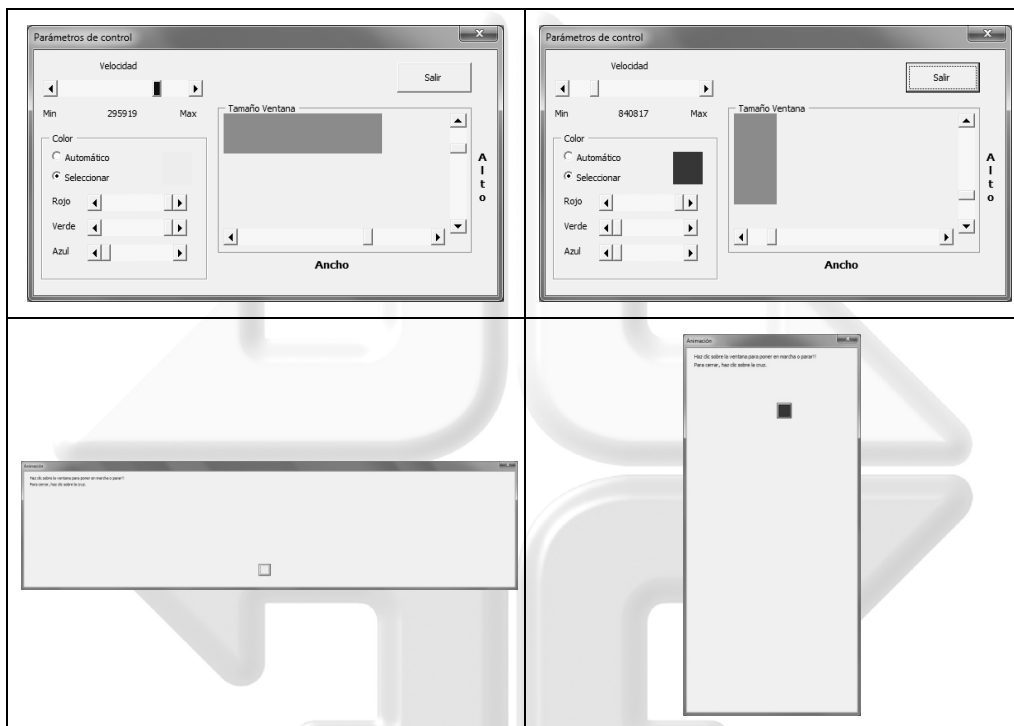


La funcionalidad prevista para cada elemento es la siguiente:

Objeto	Comentario
Barra desplazamiento <b>Velocidad</b>	Permite variar la velocidad de movimiento del botón.
Marco <b>Color</b>	Permite agrupar los botones de opción <b>Automático</b> y <b>Seleccionar</b> permitiendo que el color se calcule sólo por el propio proceso o bien, utilizando uno indicado por el usuario.
Opción <b>Automático</b>	El color lo decide el propio programa según va transcurriendo el tiempo.
Opción <b>Seleccionar</b>	Habilita las barras de desplazamiento para poder seleccionar la cantidad de color deseado en cda caso.
Barras de desplazamiento <b>Rojo, Verde y Azul.</b>	Permiten indicar el nivel de cada color en la mezcla del color resultante.
Marco <b>Ventana</b>	Muestra en azul el tamaño proporcional que tendrá la ventana sobre la que rebotará el botón. Observaremos que al cambiar este marco, cambia también el tamaño de la ventana principal.
Barras de desplazamiento <b>Alto y Ancho</b>	Al modificar su valor, modificamos el tamaño de la ventana principal.

Al cerrar esta ventana, el botón iniciará su movimiento e irá rebotando contra cada uno de los laterales de la ventana que lo contiene.

A continuación se muestran algunos ejemplos de color del botón y del tamaño de la ventana:



En previsión de bucles infinitos, se establece un tiempo de seguridad para que, al estar en marcha durante 3 minutos sin interrupción alguna, el botón se pare.

Para relanzar la llamada desde la opción Macros de MS Excel, bastará con insertar un módulo y crear el siguiente método:

```
Sub lanza_animacion()
    frmAnimacion.Show()
End Sub
```

Ra-Ma®

### 6.5.10.3 CÓDIGO FUENTE DEL FORMULARIO

El código correspondiente al formulario (**frmAnimación**) es el siguiente:

```
Option Explicit On
Public velocidad As Double
Dim x, y As Double
Dim incx, incy As Integer
Dim mover As Boolean
Dim hini, hfin As Date
Public gRed As Integer
Public gGreen As Integer
Public gBlue As Integer
Public tipo_color As Integer
Private Sub UserForm_Click()
    If mover = True Then
        mover = False
    Else
        frmControl.Show()
        DoEvents()
        btnPel.BackColor = RGB(gRed, gGreen, gBlue)
        If x > Me.Height - btnPel.Height Then x = 1
        If y > Me.Width - btnPel.Width Then y = 1
        hini = Time
        hfin = DateAdd("n", 3, hini)
        mover = True
        mueve()
    End If
End Sub

Private Sub UserForm_Initialize()
    MsgBox("Haz clic sobre la pantalla para iniciar el cuadro de diálogo y opciones de movimiento." & vbCrLf & "Haz clic sobre el botón que se mueve para finalizar o cierra la ventana.")
    x = 1
    y = 1
    incx = 1
    incy = 1
    tipo_color = 1 ' 1 - Automático, 2 - Seleccionado por usuario
    velocidad = 500000
    gRed = 0
    gGreen = 0
    gBlue = 0
    btnPel.BackColor = RGB(gRed, gGreen, gBlue)
    Me.Height = 400
    Me.Width = 400
End Sub

Private Sub btnPel_Click()
    MsgBox("Adios!!")
    Unload(Me)
End Sub
```

```
Sub mueve()  
    hini = Time  
    hfin = DateAdd("n", 3, hini) ' Tiempo de seguridad  
    Dim topex As Double  
    Dim topey As Double  
    topex = Me.Height - 40  
    topey = Me.Width - 25  
    Do While mover = True  
        Dim i As Double  
        For i = 1 To velocidad  
            Next i  
            If x > topex Or x < 1 Then  
                incx = -incx  
            End If  
            x = x + incx  
            If y > topey Or y < 1 Then  
                incy = -incy  
            End If  
            y = y + incy  
            btnPel.Move(Left:=y, Top:=x, Layout:=True)  
            If tipo_color = 1 Then  
                gBlue = gBlue + 10  
                If gBlue > 255 Then  
                    gBlue = 0  
                    gGreen = gGreen + 10  
                    If gGreen > 255 Then  
                        gGreen = 0  
                        gRed = gRed + 10  
                        If gRed > 255 Then gRed = 0  
                    End If  
                End If  
            End If  
            btnPel.BackColor = RGB(gRed, gGreen, gBlue)  
            frmAnimacion.Repaint()  
            DoEvents()  
            If Time > hfin Then mover = False  
        Loop  
    End Sub  
  
Private Sub UserForm_Terminate()  
    mover = False  
    DoEvents()  
End Sub
```

Ra-Ma®

El código correspondiente al formulario (**frmControl**) es el siguiente:

```
Option Explicit On

Dim inicio As Boolean

Private Sub btnSalir_Click()
    Unload(Me)
End Sub

Private Sub opAleatorio_Click()
    scRed.Enabled = False
    scGreen.Enabled = False
    scBlue.Enabled = False
    frmAnimacion.tipo_color = 1
End Sub

Private Sub opSeleccionar_Click()
    scRed.Enabled = True
    scGreen.Enabled = True
    scBlue.Enabled = True
    frmAnimacion.tipo_color = 2
End Sub

Private Sub scAlto_Change()
    frVentana.Height = scAlto.Value
    If Not (inicio) Then conversion_medidas_out()
End Sub

Private Sub scAncho_Change()
    frVentana.Width = scAncho.Value
    If Not (inicio) Then conversion_medidas_out()
End Sub

Private Sub scVel_Change()
    lbVel.Caption = scVel.Value
    frmAnimacion.velocidad = scVel.Value
End Sub

Private Sub scRed_Change()
    frmAnimacion.gRed = scRed.Value
    mostrar_color()
End Sub

Private Sub scGreen_Change()
    frmAnimacion.gGreen = scGreen.Value
    mostrar_color()
End Sub

Private Sub scBlue_Change()
    frmAnimacion.gBlue = scBlue.Value
```

```

    mostrar_color()
End Sub

Private Sub UserForm_Initialize()
    inicio = True
    scVel.Value = frmAnimacion.velocidad
    scRed.Value = frmAnimacion.gRed
    scGreen.Value = frmAnimacion.gGreen
    scBlue.Value = frmAnimacion.gBlue
    mostrar_color()
    conversion_medidas_in()
    If frmAnimacion.tipo_color = 1 Then
        opAleatorio.Value = True
        opSeleccionar.Value = False
        scRed.Enabled = False
        scGreen.Enabled = False
        scBlue.Enabled = False
    Else
        opAleatorio.Value = False
        opSeleccionar.Value = True
        scRed.Enabled = True
        scGreen.Enabled = True
        scBlue.Enabled = True
    End If
    inicio = False
End Sub

Sub mostrar_color()
    lbColor.BackColor = RGB(scRed.Value, scGreen.Value,
scBlue.Value)
End Sub

Sub conversion_medidas_in()
    scAlto.Value = frmAnimacion.Height / 7.5
    scAncho.Value = frmAnimacion.Width / 7.5
    frVentana.Height = scAlto.Value
    frVentana.Width = scAncho.Value
End Sub

Sub conversion_medidas_out()
    frmAnimacion.Height = scAlto.Value * 7.5
    frmAnimacion.Width = scAncho.Value * 7.5
End Sub

```

Para relanzar la llamada desde la opción **Macros de MS Excel**, bastará con insertar un módulo y crear el siguiente método:

```

Sub lanza_animacion()
    frmAnimacion.Show()
End Sub

```



## APÉNDICES

---

Excel posee muchas combinaciones de teclas que permiten realizar ciertas tareas sólo con el uso del teclado. A continuación, se muestra una relación de las mismas.

### A. ERRORES

Error	Descripción
3	Return sin Gosub
5	El argumento o la llamada al procedimiento no son válidos
6	Desbordamiento
7	Sin memoria
9	El subíndice está fuera del intervalo
10	La matriz está bloqueada de manera fija o temporal
11	División por cero
13	No coinciden los tipos

14	No hay suficiente espacio de cadena
16	La expresión es demasiado compleja
17	No se pudo realizar la operación solicitada
18	Se produjo una interrupción por parte del usuario
20	Resume sin Error
28	No hay suficiente espacio de pila
35	No se ha definido Sub, Function o Property
47	Hay demasiados recursos de código o clientes de la aplicación DLL
48	Error al cargar recursos de código o DLL
49	La convención de llamada a la DLL no es correcta
51	Error interno
52	Nombre o número de archivo incorrecto
53	Archivo no encontrado
54	Modo de archivo incorrecto
55	El archivo ya está abierto
57	Error de E/S de dispositivo
58	El archivo ya existe
59	La longitud de registro es incorrecta
61	Disco lleno
62	La entrada de datos ha sobrepasado el final del archivo
63	El número de registro es incorrecto

67	Hay demasiados archivos
68	El dispositivo no está disponible
70	Se ha denegado el permiso
71	El disco no está listo
74	No se pudo cambiar el nombre con una unidad de disco diferente
75	Error de acceso a la ruta o al archivo
76	No se ha encontrado la ruta de acceso
91	La variable de objeto o la variable de bloque With no está establecida
92	El bucle For no está inicializado
93	La cadena modelo no es válida
94	El uso de Null no es válido
97	No se pudo llamar a un procedimiento Friend de un objeto que no tiene una instancia de la clase definida
98	Una llamada a una propiedad o método no puede incluir una referencia a un objeto privado, como un argumento o un valor de retorno
298	No se puede cargar la DLL especificada del sistema
320	No se pueden utilizar nombres de dispositivos de caracteres en nombres de archivos
321	Formato de archivo no válido
322	No se pudo crear un archivo temporal necesario
325	Formato no válido en archivo de recursos
327	No se encontró el valor del dato
328	Parámetro no válido. No se puede escribir matrices

335	No se puede tener acceso al registro del sistema
336	El componente no está registrado correctamente
337	No se encontró el componente
338	El componente no se ejecutó correctamente
360	Objeto ya cargado
361	No se puede cargar ni descargar este objeto
363	No se ha encontrado el control especificado
3364	El objeto se ha descargado
365	No se puede descargar dentro de este contexto
368	El archivo especificado no está actualizado. Este programa necesita una versión más reciente
371	El objeto especificado no se puede utilizar como un formulario propietario para Show()
380	Valor de propiedad no válido
381	El índice de la matriz de propiedades no es válido
382	Set no se admite en tiempo de ejecución
383	Set no se admite (propiedad de sólo lectura)
385	Se necesita un índice de matriz de propiedad
387	Set no está permitido
393	La propiedad no se puede leer en tiempo de ejecución
394	La propiedad 'l' es de sólo escritura
400	El formulario ya está mostrado. No se pudo mostrar en forma modal

402	Primero debe cerrar u ocultar el formulario modal superior
419	Permiso denegado para utilizar el objeto
422	No se encontró la propiedad
423	No se ha encontrado la propiedad o el método
424	Se requiere un objeto
425	Uso no válido del objeto
429	El componente no puede crear el objeto o devolver la referencia a ese objeto
430	Esta clase no acepta Automatización
432	No se encontró el nombre del archivo o de la clase durante la operación de Automatización
438	El objeto no acepta esta propiedad o método
440	Error de Automatización
442	Se ha perdido la conexión con la biblioteca de tipos o con la biblioteca de objetos para procesos remotos
443	El objeto de Automatización no tiene un valor predeterminado
445	El objeto no permite esta acción
446	El objeto no acepta argumentos con nombre
447	El objeto no acepta la configuración regional actual
448	No se ha encontrado el argumento con nombre
449	El argumento no es opcional
450	Número de argumentos erróneo o asignación de propiedad no válida
451	La propiedad let procedure no se ha definido y la propiedad get

	procedure no ha devuelto un objeto
452	El número ordinal no es válido
453	No se encuentra la función de biblioteca especificada
454	No se encuentra el recurso de código
455	Error en el bloqueo de los recursos de código
457	Esta clave ya está asociada con un elemento de esta colección
458	La variable usa un tipo de Automatización no admitido en Visual Basic
459	Este componente no acepta este conjunto de eventos eventos
460	El formato del Portapapeles no es válido
461	No se encontró el método o el dato miembro
462	El servidor remoto no existe o no se encuentra disponible
463	La clase no está registrada en la máquina local
480	No se puede crear una imagen AutoRedraw
481	Imagen no válida
482	Error de impresora
483	El controlador de impresora no admite la propiedad especificada
484	Problemas al obtener información de la impresora desde el sistema; asegúrese de que la impresora está configurada correctamente
485	El tipo de imagen no es válido
486	No se puede imprimir la imagen de formulario en este tipo de impresora
520	No se puede vaciar el Portapapeles
521	No se puede abrir el Portapapeles

735	No se pudo guardar el archivo en TEMP
744	No se encontró el texto de búsqueda
746	Sustituciones demasiado largas
31001	Memoria insuficiente
31004	Ningún objeto
31018	La clase no está establecida
31027	No se puede activar el objeto
31032	No se puede crear el objeto incrustado
31036	Error al guardar en archivo
31037	Error al cargar de archivo

## B. TIPOS DE ARCHIVO

Valor	Nombre	Descripción
-4158	xlCurrentPlatformText	Texto de la plataforma actual
-4143	xlWorkbookNormal	Libro normal
2	xlSYLK	SYLK (vínculo simbólico)
4	xlWKS	Hoja de cálculo
5	xlWK1	WK1
6	xlCSV	CSV
7	xlDBF2	DBF2
8	xlDBF3	DBF3

9	xlDIF	DIF
11	xlDBF4	DBF4
14	xlWJ2WD1	WJ2WD1
15	xlWK3	WK3
16	xlExcel2	Excel2
17	xlTemplate	Plantilla
17	xlTemplate8	Plantilla 8
18	xlAddIn	Complemento de Microsoft Excel 97-2003
18	xlAddIn8	Complemento de Microsoft Excel 97-2003
19	xlTextMac	Texto de Macintosh
20	xlTextWindows	Texto de Windows
21	xlTextMSDOS	Texto de MSDOS
22	xlCSVMac	CSV de Macintosh
23	xlCSVWindows	CSV de Windows
24	xlCSVMSDOS	CSV de MSDOS
25	xlIntlMacro	Macro internacional
26	xlIntlAddIn	Complemento internacional
27	xlExcel2FarEast	Excel2 Este asiático®
28	xlWorks2FarEast	Works2 Este asiático
29	xlExcel3	Excel3



30	xlWK1FMT	WK1FMT
31	xlWK1ALL	WK1ALL
32	xlWK3FM3	WK3FM3
33	xlExcel4	Excel4
34	xlWQ1	WQ1
35	xlExcel4Workbook	Libro de Excel4
36	xlTextPrinter	Texto de impresora
38	xlWK4	WK4
39	xlExcel5	Excel5
39	xlExcel7	Excel7
40	xlWJ3	WJ3
41	xlWJ3FJ3	WJ3FJ3
42	xlUnicodeText	Texto Unicode
43	xlExcel9795	Excel9795
44	xlHtml	Formato HTML
45	xlWebArchive	Archivo de almacenamiento Web
46	xlXMLSpreadsheet	Hoja de cálculo XML
50	xlExcel12	Excel12
51	xlOpenXMLWorkbook	Libro de Open XML
51	xlWorkbookDefault	Predeterminado de hoja de cálculo

52	xlOpenXMLWorkbookMacroEnabled	Libro de Open XML con macros habilitadas
53	xlOpenXMLTemplateMacroEnabled	Plantilla XML abierta con macros habilitadas
54	xlOpenXMLTemplate	Plantilla XML abierta
55	xlOpenXMLAddIn	Complemento de Open XML
56	xlExcel8	Excel8
60	xlOpenDocumentSpreadsheet	Hoja de cálculo de OpenDocument

### C. EQUIVALENCIA DE FUNCIONES DE ESPAÑOL A INGLÉS

Español	Inglés
ABRIR.ARCHIVO	OPEN.DIALOG
ABRIR	FOPEN
ABS	ABS
ACOS	ACOS
ACOSH	ACOSH
AGREGAR.BARRA	ADD.BAR
AGREGAR.BARRA.HERRAMIENTAS	ADD.TOOLBAR
AGREGAR.COMANDO	ADD.COMMAND
AGREGAR.MENU	ADD.MENU
AGRUPAR	GROUP
AHORA	NOW
ALEATORIO	RAND
AÑO	YEAR
APL.TITULO	APP.TITLE
ARCHIVOS	FILES
AREAS	AREAS
ARGUMENTO	ARGUMENT
ASC	ASC
ASENO	ASIN
ASENOH	ASINH
ATAN	ATAN

ATAN2	ATAN2
ATANH	ATANH
AYUDA	HELP
BDCONTAR	DCOUNT
BDCONTARA	DCOUNTA
BDDVEST	DSTDEV
BDDVESTP	DSTDEVP
BDEXTRAER	DGET
BDMAX	DMAX
BDMIN	DMIN
BDPRODUCTO	DPRODUCT
BDPROMEDIO	DAVERAGE
BDSUMA	DSUM
BDVAR	DVAR
BDVARP	DVARP
BINOM.CRIT	CRITBINOM
BUSCAR	LOOKUP
BUSCARH	HLOOKUP
BUSCARV	VLOOKUP
CADENA.FECHA	DATESTRING
CADENA.NUMERO	NUMBERSTRING
CAMBIAR.NOMBRE.COMANDO	RENAME.COMMAND
CANCELAR.TECLA	CANCEL.KEY
CARACTER	CHAR
CELDA	CELL
CELDA.ACTIVA	ACTIVE.CELL
CERRARA	FCLOSE
CODIGO	CODE
COEF.DE.CORREL	CORREL
COEFICIENTE.ASIMETRIA	SKEW
COEFICIENTE.R2	RSQ
COINCIDIR	MATCH
COLUMNA	COLUMN
COLUMNAS	COLUMNS
COMBINAT	COMBIN
CONCATENAR	CONCATENATE <sup>®</sup>
CONTAR	COUNT
CONTAR.BLANCO	COUNTBLANK
CONTAR.SI	COUNTIF
CONTARA	COUNTA
COS	COS

COSH	COSH
COVAR	COVAR
CREAR.OBJETO	CREATE.OBJECT
CRECIMIENTO	GROWTH
CUADRO.DE.DIALOGO	DIALOG.BOX
CUADRO.DE.TEXTO	TEXT.BOX
CUARTIL	QUARTILE
CURTOSIS	KURT
DB	DB
DBCS	DBCS
DDB	DDB
DECIMAL	FIXED
DERECHA	RIGHT
DERECHAB	RIGHTB
DESREF	OFFSET
DESREGISTRAR	UNREGISTER
DESVEST	STDEV
DESVESTP	STDEVP
DESVIA2	DEVSQ
DESVPROM	AVEDEV
DETENER	HALT
DIA	DAY
DIAS360	DAYS360
DIASEM	WEEKDAY
DIRECCION	ADDRESS
DIRECTORIO	DIRECTORY
DISTR.WEIBULL	WEIBULL
DISTR.BETA	BETADIST
DISTR.BETA.INV	BETAINV
DISTR.BINOM	BINOMDIST
DISTR.CHI	CHIDIST
DISTR.EXP	EXPONDIST
DISTR.F	FDIST
DISTR.F.INV	FINV
DISTR.GAMMA	GAMMADIST
DISTR.GAMMA.INV	GAMMAINV
DISTR.HIPERGEOM	HYPGEOMDIST
DISTR.LOG.INV	LOGINV
DISTR.LOG.NORM	LOGNORMDIST
DISTR.NORM	NORMDIST
DISTR.NORM.ESTAND	NORMSDIST

DISTR.NORM.ESTAND.INV	NORMSINV
DISTR.NORM.INV	NORMINV
DISTR.T	TDIST
DISTR.T.INV	TINV
DOCUMENTOS	DOCUMENTS
DVS	VDB
ECHO	ECHO
EJEC	EXEC
EJECUTA	EXECUTE
ELEGIR	CHOOSE
ELIMINAR.BARRA	DELETE.BAR
ELIMINAR.BARRA.HERRAMIENTAS	DELETE.TOOLBAR
ELIMINAR.COMANDO	DELETE.COMMAND
ELIMINAR.MENU	DELETE.MENU
ENCONTRAR	FIND
ENCONTRARB	FINDB
ENTERO	INT
ERROR	ERROR
ERROR.TIPICO.XY	STEYX
ESBLANCO	ISBLANK
ESCENARIO.INDICAR	SCENARIO.GET
ESCRIBIRA	FWRITE
ESCRIBIRALN	FWRITELN
ESERR	ISERR
ESERROR	ISERROR
ESLOGICO	ISLOGICAL
ESNOD	ISNA
ESNOTEXTO	ISNONTEXT
ESNUMERO	ISNUMBER
ESPACIOS	TRIM
ESREF	ISREF
ESTABLECER.NOMBRE	SET.NAME
ESTABLECER.VALOR	SET.VALUE
ESTEXTO	ISTEXT
ESTIMACION.LINEAL	LINEST
ESTIMACION.LOGARITMICA	LOGEST
EVALUAR	EVALUATE
EXP	EXP
EXTRAE	MIDB
EXTRAE	MID
FACT	FACT

FECHA	DATE
FECHANUMERO	DATEVALUE
FILA	ROW
FILAS	ROWS
FISHER	FISHER
FORMULA.CONVERTIR	FORMULA.CONVERT
FRECUENCIA	FREQUENCY
GAMMA.LN	GAMMALN
GET.MOVIE	GET.MOVIE
GRADOS	DEGREES
GUARDAR.ARCHIVO	SAVE.DIALOG
GUARDAR.BARRA.HERRAMIENTAS	SAVE.TOOLBAR
HABILITAR.COMANDO	ENABLE.COMMAND
HABILITAR.HERRAMIENTA	ENABLE.TOOL
HALLAR	SEARCH
HALLARB	SEARCHB
HORA	HOUR
HORANUMERO	TIMEVALUE
HOY	TODAY
ID.REGISTRO	REGISTER.ID
IGUAL	EXACT
INDICAR.AREA.DE.TRABAJO	GET.WORKSPACE
INDICAR.BARRA	GET.BAR
INDICAR.BARRA.HERRAMIENTAS	GET.TOOLBAR
INDICAR.CAMPO.TABLA.DI	GET.PIVOT.FIELD
INDICAR.CELDA	GET.CELL
INDICAR.DEF	GET.DEF
INDICAR.DOCUMENTO	GET.DOCUMENT
INDICAR.ELEMENTO.GRAFICO	GET.CHART.ITEM
INDICAR.ELEMENTO.TABLA.DI	GET.PIVOT.ITEM
INDICAR.FORMULA	GET.FORMULA
INDICAR.HERRAMIENTA	GET.TOOL
INDICAR.INFO.VINCULO	GET.LINK.INFO
INDICAR.LIBRO	GET.WORKBOOK
INDICAR.NOMBRE	GET.NAME
INDICAR.NOTAS	GET.NOTE
INDICAR.OBJETO	GET.OBJECT
INDICAR.TABLA.DINAMICA	GET.PIVOT.TABLE
INDICAR.VENTANA	GET.WINDOW
INDICE	INDEX
INDIRECTO	INDIRECT

INFO	INFO
INICIAR	INITIATE
INT.PAGO.DIR	ISPMT
INTERSECCION	INTERCEPT
INTERVALO.CONFIANZA	CONFIDENCE
INTRODUCIR	INPUT
IR.A	GOTO
IZQUIERDA	LEFT
IZQUIERDAB	LEFTB
JERARQUIA	RANK
K.ESIMO.MAYOR	LARGE
K.ESIMO.MENOR	SMALL
LARGO	LEN
LARGOB	LENB
LEERA	FREAD
LEERALN	FREADLN
LIMPIAR	CLEAN
LN	LN
LOG	LOG
LOG10	LOG10
LLAMADOR	CALLER
LLAMAR	CALL
MARCAR.COMANDO	CHECK.COMMAND
MAX	MAX
MAYUSC	UPPER
MDETERM	MDETERM
MEDIA.ACOTADA	TRIMMEAN
MEDIA.ARMO	HARMEAN
MEDIA.GEOM	GEOMEAN
MEDIANA	MEDIAN
MES	MONTH
MIN	MIN
MINUSC	LOWER
MINUTO	MINUTE
MINVERSA	MINVERSE
MMULT	MMULT
MODA	MODE
MONEDA	DOLLAR
MOSTRAR.BARRA	SHOW.BAR
MOVIE.COMMAND	MOVIE.COMMAND
MULTIPLO.INFERIOR	FLOOR

MULTIPLO.SUPERIOR	CEILING
N	N
NEGBINOMDIST	NEGBINOMDIST
NO	NOT
NOD	NA
NOMBRES	NAMES
NOMPROPIO	PROPER
NORMALIZACION	STANDARDIZE
NOTA	NOTE
NPER	NPER
NSHORA	TIME
NUMERO.ROMANO	ROMAN
O	OR
OPCIONES.INDICAR.LISTAS	OPTIONS.LISTS.GET
PAGO	PMT
PAGOINT	IPMT
PAGOPRIN	PPMT
PASO.A.PASO	STEP
PAUSA	PAUSE
PEARSON	PEARSON
PENDIENTE	SLOPE
PERCENTIL	PERCENTILE
PERMUTACIONES	PERMUT
PERSONALIZAR.DESHACER	CUSTOM.UNDO
PERSONALIZAR.REPETIR	CUSTOM.REPEAT
PI	PI
POISSON	POISSON
POSICIONA	FPOS
POTENCIA	POWER
PRESIONAR.HERRAMIENTA	PRESS.TOOL
PROBABILIDAD	PROB
PRODUCTO	PRODUCT
PROMEDIO	AVERAGE
PRONOSTICO	FORECAST
PRUEBA.CHI	CHITEST
PRUEBA.CHI.INV	CHIINV
PRUEBA.F	FTEST
PRUEBA.FISHER.INV	FISHERINV
PRUEBA.T	TTEST
PRUEBA.Z	ZTEST
RADIANES	RADIANS



RAIZ	SQRT
RANGO.PERCENTIL	PERCENTRANK
REANUDAR	RESUME
REDONDEA.IMPARG	ODD
REDONDEA.PARG	EVEN
REDONDEAR	ROUND
REDONDEAR.MAS	ROUNDUP
REDONDEAR.MENOS	ROUNDDOWN
REEMPLAZAR	REPLACE
REEMPLAZARB	REPLACEB
REFABS	ABSREF
REFREL	RELREF
REFTEXTO	REFTEXT
REGISTRAR	REGISTER
REINICIAR	RESTART
REPETIR	REPT
RESIDUO	MOD
RESTABLECER.BARRA	RESET.TOOLBAR
RESULTADO	RESULT
REVISION.ORTOGRAFICA	SPELLING.CHECK
SEGUNDO	SECOND
SELECCION	SELECTION
SENO	SIN
SENOH	SINH
SERIES	SERIES
SI	IF
SIFECHA	DATEDIF
SIGNO	SIGN
SLN	SLN
SOLICITAR	REQUEST
SUBTOTALES	SUBTOTAL
SUMA	SUM
SUMA.CUADRADOS	SUMSQ
SUMAPRODUCTO	SUMPRODUCT
SUMAR.SI	SUMIF
SUMAX2MASY2	SUMX2PY2
SUMAX2MENOSY2	SUMX2MY2
SUMAXMENOSY2	SUMXMY2
SUSTITUIR	SUBSTITUTE
SYD	SYD
T	T

TABLA.DI.ADICIONAR.DATOS	PIVOT.ADD.DATA
TAMAÑO	FSIZE
TAN	TAN
TANH	TANH
TASA	RATE
TENDENCIA	TREND
TERMINAR	TERMINATE
TEXTO	TEXT
TEXTOREF	TEXTREF
TIPO	TYPE
TIPO.DE.ERROR	ERROR.TYPE
TIR	IRR
TIRM	MIRR
TRANSFERIR	POKE
TRANSPONER	TRANSPOSE
TRUNCAR	TRUNC
ULTIMO.ERROR	LAST.ERROR
USDOLLAR	USDOLLAR
VA	PV
VALOR	VALUE
VALREF	DEREF
VAR	VAR
VARP	VARP
VENTANA.TITULO	WINDOW.TITLE
VENTANAS	WINDOWS
VF	FV
VINCULOS	LINKS
VNA	NPV
VOLATIL	VOLATILE
Y	AND
FALSO	FALSE
VERDADERO	TRUE

Ra-Ma®

## D. CONSTANTES DE USO FRECUENTE

Constante	Descripción
<b>vbBack</b>	Retroceso
<b>vbCr</b>	Retorno de carro.
<b>vbCrLf</b>	Retorno de carro y salto de línea.
<b>vbFormFeed</b>	Salto de página.
<b>vbLf</b>	Salto de línea.
<b>vbNewLine</b>	Carácter de nueva línea.
<b>vbNullChar</b>	Carácter null.
<b>vbNullString</b>	Cadena de longitud cero.
<b>vbObjectError</b>	Número de error del objeto.
<b>vbTab</b>	Carácter tabulación.
<b>vbVerticalTab</b>	Retorno de carro.

## E. TECLAS PARA SENDKEYS

**Ejemplo:**

Sendkey "{ESC}" o SendKey "{F1}"

Acción / Valor
RETROCESO {BACKSPACE}, {BS} o {BKSP}
INTER {BREAK}
BLOQ MAYÚS {CAPSLOCK}
SUPR {DELETE} o {DEL}
FLECHA ABAJO {DOWN}

FIN {END}
ENTRAR {ENTER} o ~
ESC {ESC}
AYUDA {HELP}
INICIO {HOME}
INSERT {INSERT} o {INS}
FLECHA IZQUIERDA {LEFT}
BLOQ NÚM {NUMLOCK}
RE PÁG {PGDN}
AV PÁG {PGUP}
IMPR PANT {PRTSC}
FLECHA DERECHA {RIGHT}
BLOQ DESPL {SCROLLLOCK}
TAB {TAB}
FLECHA ARRIBA {UP}
F1 {F1}
F2 {F2}
F3 {F3}
F4 {F4}
F5 {F5}
F6 {F6}

F7 {F7}
F8 {F8}
F9 {F9}
F10 {F10}
F11 {F11}
F12 {F12}
F13 {F13}
F14 {F14}
F15 {F15}
F16 {F16}

**F. VARIABLES A UTILIZAR PARA ‘ENVIRON’**

**Environ("PATH")** ‘ Devuelve valor de la variable PATH

Variable
ALLUSERSPROFILE
APPDATA
classpath
CommonProgramFiles
CommonProgramFilesx86)
CommonProgramW6432
COMPUTERNAME

ComSpec
FP_NO_HOST_CHECK
HOMEDRIVE
HOMEPATH
LOCALAPPDATA
LOGONSERVER
NUMBER_OF_PROCESSORS
OS
Path
PATHEXT
PROCESSOR_ARCHITECTURE
PROCESSOR_IDENTIFIER
PROCESSOR_LEVEL
PROCESSOR_REVISION
ProgramData
ProgramFiles
ProgramFilesx86)
ProgramW6432
PSModulePath
PUBLIC
SESSIONNAME

SystemDrive
SystemRoot
TEMP
TMP
UNIADMIN_HOME
USERDOMAIN
USERNAME
USERPROFILE
VS100COMNTOOLS
VS90COMNTOOLS
WecVersionForRosebud.14EC
Windir

Ra-Ma®





# ÍNDICE ALFABÉTICO

---

## A

Asignar macro .....631, 632, 656  
Ayuda..... 4, 140, 179, 207, 210, 567, 606

## B

Barra de herramientas .....654  
Barras de herramientas .....177  
Biblioteca .....139, 739

## C

Call .....749  
Cells .....637

## D

DLL.....139, 176

## E

Ejecutar .....124, 150  
Ejecutar .....627  
Explorador.....625, 637  
Exportar.....308, 328, 448

## F

Formulario ..... 631, 653, 656  
Funciones de usuario..... 179  
Function .....435, 498

## G

Gosub .....735  
Goto .....167, 206, 749

## I

Importar .....236, 308, 309

## M

Módulo140, 223, 624, 625, 627, 652, 676,  
685, 695, 704, 709, 713, 715, 716, 725,  
729, 733  
Mostrar.... 43, 45, 46, 165, 254, 258, 319, 648

## O

On Error .....167  
Option ..... 18, 19, 24, 27, 50, 53, 537, 627,  
660, 665, 680, 707, 715, 716, 721, 730,  
732

**P**

Personalizar .....628  
Portapapeles .....437, 442, 524, 543, 563,  
582, 603, 620  
Programador..... 177, 624, 627, 631

**R**

Range ..... 157, 162, 175, 176, 312, 318,  
319, 336, 360, 362, 363, 367, 378, 393,  
396, 643

**T**

Thisworkbook ..... 178, 637, 638, 641, 652,  
699, 716

**V**

Ventana de código.....631, 638  
Ventana de código.....626  
Vista..... 157, 171, 234  
Visual Basic ..... 161, 180, 223, 291, 294,  
344, 624, 699, 706, 740



Ra-Ma®