

MANUAL DE VBA

Felipe Ordenes Odi

CONTENIDO

1	INTRODUCCION	11
1.1	ELEMENTOS	12
1.1.1	Ventana Proyecto	12
1.1.2	Ventana Propiedades	13
1.1.3	Ventana Código	14
1.1.4	Ventana Inmediato	14
1.1.5	Ventana Locales.....	15
1.1.6	Ventana Inspección	15
2	ESTRUCTURA	16
2.1	VARIABLES.....	16
2.1.1	Tipos de Variables.....	16
2.1.2	Variables Locales	17
2.1.3	Variables Públicas	17
2.1.4	Variables Estáticas	17
2.1.5	Convertir Variables	17
2.2	SENTENCIAS	18
2.2.1	Do Loop.....	18
2.2.2	For Next	18
2.2.3	For Each Next.....	19
2.2.4	While Wend	19
2.2.5	GoTo	19
2.2.6	With End With	20
2.2.7	If Then	20
2.2.8	Select Case.....	21
2.2.9	DoEvents.....	21
2.2.10	On Error GoTo.....	22
2.2.11	On Error Resume Next.....	22
3	ARRAYS	23

3.1	ARRAYS FIJOS	23
3.2	ARRAYS VARIABLES	23
3.3	ARRAYS BIDIMENSIONALES	23
3.4	FUNCIONES DE ARRAY	24
3.4.1	Redim.....	24
3.4.2	Redim Preserve.....	24
3.4.3	LBound.....	24
3.4.4	UBound	24
3.4.5	Join.....	24
3.4.6	Split	25
3.4.7	Filter.....	25
4	USERFORM	26
4.1	CUADRO DE HERRAMIENTAS.....	26
4.2	FUNCIONES DE USERFORM	27
4.2.1	Load	27
5	DECLARACIONES.....	28
5.1	OPTION.....	28
5.1.1	Option Explicit	28
5.1.2	Option Private Module	28
5.1.3	Option Base 0	28
5.1.4	Option Base 1	28
5.1.5	Option Compare Binary	28
5.1.6	Option Compare Text	29
5.2	EVENTOS DE LIBROS.....	29
5.2.1	Activate.....	29
5.2.2	AddinInstall.....	29
5.2.3	AddinUninstall	29
5.2.4	AfterXmlExport	29
5.2.5	AfterXmlImport.....	30
5.2.6	BeforeClose.....	30

5.2.7	BeforePrint	30
5.2.8	BeforeSave.....	30
5.2.9	BeforeXmlExport	30
5.2.10	BeforeXmlImport.....	30
5.2.11	Deactivate.....	30
5.2.12	NewSheet	30
5.2.13	Open	31
5.2.14	PivoteTableCloseConnection.....	31
5.2.15	PivoteTableOpenConnection.....	31
5.2.16	RowsetComplete	31
5.2.17	SheetActivate.....	31
5.2.18	SheetBeforeDoubleClick.....	31
5.2.19	SheetBeforeRightClick	31
5.2.20	SheetCalculate	32
5.2.21	SheetChange.....	32
5.2.22	SheetDesactivate	32
5.2.23	SheetFolowHyperlink.....	32
5.2.24	SheetPivotTableUpdate.....	32
5.2.25	SheetSelectionChange	32
5.2.26	Sync.....	32
5.2.27	WindowActivate	33
5.2.28	WindowDesactivate.....	33
5.2.29	WindowResize	33
5.3	EVENTOS DE HOJAS	33
5.3.1	Activate.....	33
5.3.2	BeforeDoubleClick	33
5.3.3	BeforeRightClick	34
5.3.4	Calculate	34
5.3.5	Change	34
5.3.6	Deactivate.....	34

5.3.7	FollowHyperLink	34
5.3.8	PivotTableUpdate	34
5.3.9	SelectionChange	34
6	OPM.....	35
6.1	Application	35
6.1.1	.Caption.....	35
6.1.2	.Path.....	35
6.1.3	.ScreenUpdating	35
6.1.4	.WindowState	35
6.1.5	.DisplayStatusBar	36
6.1.6	.DisplayFormulaBar	36
6.1.7	.Calculate	36
6.1.8	.Help.....	36
6.1.9	.Run.....	36
6.1.10	.EnableEvents	37
6.1.11	.DisplayAlerts	37
6.1.12	.Quit	37
6.1.13	.Calculation	37
6.1.14	.CutCopyMode.....	37
6.2	WorkBooks.....	38
6.2.1	.Name	38
6.2.2	.Path.....	38
6.2.3	.Author.....	38
6.2.4	.Saved	39
6.2.5	.FullName.....	39
6.2.6	.Open	39
6.2.7	.Activate.....	39
6.2.8	.Close	39
6.2.9	.Add.....	40
6.2.10	.Protect	40

6.2.11	.Unprotect.....	40
6.2.12	.Save.....	40
6.2.13	.SaveAs.....	40
6.2.14	.SaveCopyAs.....	41
6.2.15	.Delete	41
6.2.16	.Calculate	41
6.2.17	Kill	41
6.3	WorkSheets.....	42
6.3.1	.UsedRange.....	42
6.3.2	.Tab	42
6.3.3	.Name	42
6.3.4	.Index	43
6.3.5	.Count	43
6.3.6	.Next.....	43
6.3.7	.Previous	43
6.3.8	.Activate.....	43
6.3.9	.Calculate	44
6.3.10	.Delete	44
6.3.11	.Protect	44
6.3.12	.Unprotect.....	44
6.3.13	.Paste	44
6.3.14	.Add.....	45
6.3.15	.Select	45
6.3.16	.Move.....	45
6.3.17	.Copy	45
6.3.18	.Visible	45
6.3.19	.PrintOut	46
6.3.20	.PageSetup.....	46
6.3.21	.PageSetup.PrintArea	46
6.3.22	.PageSetup.LeftHeader.....	46

6.3.23	.PageSetup.LeftFooter	46
6.4	Range	47
6.4.1	.EntireRow	47
6.4.2	.EntireColumn	47
6.4.3	.Dependents	48
6.4.4	.SpecialCells	48
6.4.5	.End	48
6.4.6	.CurrentRegion.....	48
6.4.7	.Count	49
6.4.8	.FormulaR1C1	49
6.4.9	.Name	49
6.4.10	.Value	49
6.4.11	.Formula.....	49
6.4.12	.Text	50
6.4.13	.Insert.....	50
6.4.14	.Delete	50
6.4.15	.Address	50
6.4.16	.HasFormula.....	50
6.4.17	.NumberFormat	51
6.4.18	.Offset	51
6.4.19	.Cells.....	51
6.4.20	.HorizontalAlignment.....	51
6.4.21	.Select	51
6.4.22	.Activate.....	52
6.4.23	.Calculate	52
6.4.24	.Copy	52
6.4.25	.Delete	52
6.4.26	.Clear.....	52
6.4.27	.ClearComments	53
6.4.28	.ClearFormats	53

6.4.29	.ClearContents	53
6.4.30	.ClearNotes	53
6.4.31	.MergeCells	53
6.4.32	.Hidden	53
6.4.33	.Locked	54
6.4.34	.PasteSpecial	54
6.4.35	.Cut	54
6.4.36	.Sort	54
6.4.37	.ScrollArea	54
6.4.38	.AutoFilter	55
6.4.39	.Font	55
6.4.40	.Font.Bold	55
6.4.41	.Font.Size	55
6.4.42	.Interior	55
6.4.43	.Interior.ColorIndex	56
6.5	ACTIVIEWINDOW	56
6.5.1	.ScrollRow	56
6.5.2	.ScrollColumn	56
6.6	NAMES	56
6.6.1	.Add	56
6.7	MSGBOX	57
6.8	INPUTBOX	57

1 INTRODUCCION

VBA (Visual Basic for Applications) es un lenguaje de programación que permite crear macros, es decir, un conjunto de instrucciones de código que permiten realizar una tarea determinada. Excel 2007 trae incorporado un editor de VBA, por lo que las macros se programan dentro del mismo programa Excel.

Las macros sirven para:

- Automatizar procesos.
- Crear funciones.
- Crear comandos, complementos y menús.
- Crear aplicaciones.

Cuando se habla de macros se habla de Objetos, Propiedades y Métodos (OPM). Estos son los 3 conceptos generales más importantes que se deben conocer a la hora de programar macros.

Objetos: Corresponden a los elementos mismos, tales como una hoja, un libro, el Excel mismo, una tabla, un rango, un gráfico, una columna, una fila, etc.

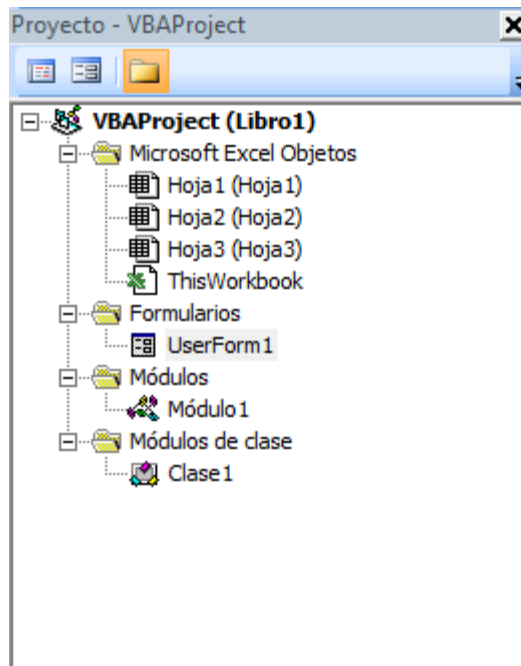
Propiedades: Corresponden a las características de cada objeto, tales como el ancho, alto, color, etc.

Métodos: Corresponden a las acciones que se pueden ejercer sobre los objetos, tales como abrir, cerrar, ejecutar, activar, mover, copiar, borrar, etc.

1.1 ELEMENTOS

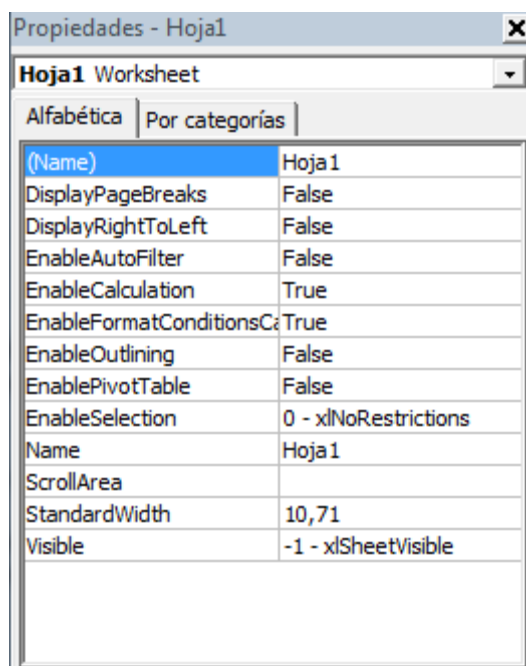
1.1.1 Ventana Proyecto

La ventana proyecto contiene las carpetas donde se guardan las macros. La ventana proyecto es la siguiente.



1.1.2 Ventana Propiedades

La ventana propiedades contiene las propiedades del objeto seleccionado. La ventana propiedades es la siguiente.



1.1.3 Ventana Código

La ventana código contiene el código de la macro. La ventana código es la siguiente.



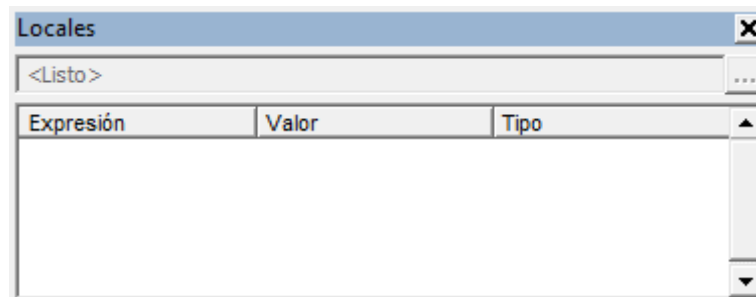
1.1.4 Ventana Inmediato

La ventana inmediato permite escribir, ejecutar y probar un código rápidamente, sin tener que armar toda la macro con la estructura, sino colocando un signo de interrogación previo a la línea de código. La ventana inmediato es la siguiente.



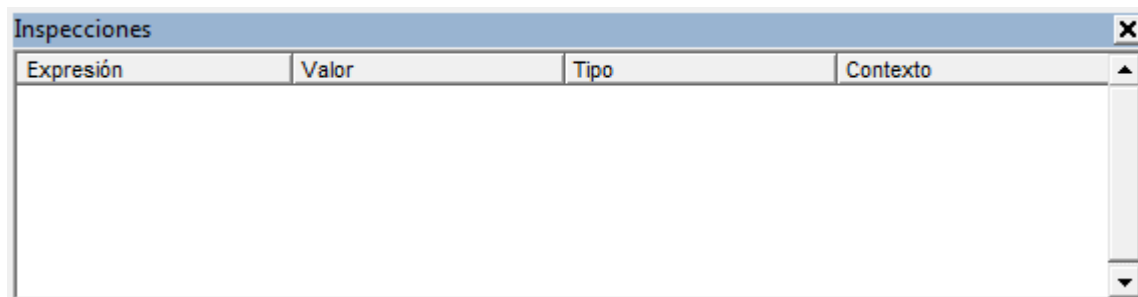
1.1.5 Ventana Locales

La ventana locales contiene todas las variables del objeto actual y los valores de estas cuando el código se esta ejecutando. La ventana locales es la siguiente.



1.1.6 Ventana Inspección

La ventana inspección contiene las expresiones a evaluar previamente definidas. La ventana inspección es la siguiente.



2 ESTRUCTURA

2.1 VARIABLES

2.1.1 Tipos de Variables

Variable	Tamaño	Tipo	Extremos	Intervalo	Valor Inicial
Byte	1 Byte	Número Entero	Menor +	0	0
			Mayor +	255	
			Menor –	-	
			Mayor –	-	
Integer	2 Bytes	Número Entero	Menor +	0	0
			Mayor +	32.767	
			Menor –	0	
			Mayor –	-32.768	
Long	4 Bytes	Número Entero	Menor +	0	0
			Mayor +	9.223.372.036.854.775.807	
			Menor –	0	
			Mayor –	-9.223.372.036.854.775.808	
Currency	8 Bytes	Número Entero	Menor +	0	0
			Mayor +	-922.337.203.685.477,5808	
			Menor –	0	
			Mayor –	922.337.203.685.477,5807	
Single	4 Bytes	Número Decimal	Menor +	1,401298 E-45	0
			Mayor +	3,4028235 E+38	
			Menor –	-1,401298 E-45	
			Mayor –	-3,4028235 E+38	
Double	8 Bytes	Número Decimal	Menor +	-1,79769313486231570 E+308	0
			Mayor +	-4,94065645841246544 E-324	
			Menor –	4,94065645841246544 E-324	
			Mayor –	1,79769313486231570 E+308	
Decimal	14 Bytes	Número Entero	Menor +	0	0
			Mayor +	79.228.162.514.264.337.593.543.950.335	
			Menor –	0	
			Mayor –	-79.228.162.514.264.337.593.543.950.335	
		Número Decimal	Menor +	0,00000000000000000000000000000001	
			Mayor +	7,9228162514264337593543950335	
			Menor –	-0,00000000000000000000000000000001	
			Mayor –	-7,9228162514264337593543950335	
Date	8 Bytes	Fecha		01/01/1900 a 31/12/9999	
Object	4 Bytes	Objeto		Cualquier referencia a tipo Object	Nothing
String (Largo Variable)	10 Bytes + Largo Cadena	Texto		Desde 0 a 2.000 millones	""
String (Largo Fija)	16 Bytes + Largo Cadena	Texto		Desde 1 a 65.400	""
Variant (Números)	16 Bytes	Todo		Intervalo de Double	Empty
Variant (Carácteres)	22 Bytes + Largo Cadena	Todo		Intervalo de String (Longitud Variable)	Empty
Boolean	2 Bytes	Boolean		True o False	False

2.1.2 Variables Locales

Corresponden a variables declaradas dentro de una macro, y que solo pueden ser usadas dentro de esta. Cuando la macro termina de ejecutarse, la variable desaparece y se libera la memoria. La declaración de una variable local es la siguiente.

Dim NombreVariable as TipoVariable

2.1.3 Variables Públicas

Corresponden a variables declaradas fuera de todas las macros, y que están disponibles para todas. La declaración de una variable pública es la siguiente.

Public NombreVariable as TipoVariable

2.1.4 Variables Estáticas

Corresponden a variables declaradas dentro de una macro, y que conservan su valor mientras el módulo, módulo de clase o userform estén en uso o abiertos. La declaración de una variable estática es la siguiente.

Static NombreVariable as TipoVariable

2.1.5 Convertir Variables

Función	Tipo Devuelto
Cbyte	Byte
CInt	Integer
CLong	Long
CCur	Currency
CSng	Single
CDBl	Double
CDec	Decimal
CDate	Date
CStr	String
CVar	Variant
CBool	Boolean

2.2 SENTENCIAS

2.2.1 Do Loop

Do Loop se utiliza para repetir las mismas instrucciones varias veces hasta que una condición tenga lugar.

```
Do  
[Instrucciones]  
Loop
```

2.2.2 For Next

For Next se utiliza para repetir un grupo de instrucciones un número especificado de veces.

```
For [Contador] = [Principio] To [Fin] Step [Incremento]
```

```
[Instrucciones]
```

```
If [Condicion] then
```

```
[Exit For]
```

```
End If
```

```
Next [Contador]
```

2.2.3 For Each Next

For Each Next se utiliza para repetir un grupo de instrucciones para cada elemento de una matriz o colección.

For Each [Elemento] In [Grupo]

[Instrucciones]

If [Condicion] then

[Exit For]

End If

Next [Elemento]

2.2.4 While Wend

While Wend se utiliza para repetir una serie de instrucciones mientras se cumpla una determinada condición, es decir, tenga el valor True.

While [Condicion]

[Instrucciones]

Wend

2.2.5 GoTo

GoTo se utiliza para variar el sentido que debería tomar una macro, ya que cuando una macro llega a una línea que contiene dicha instrucción "salta" al lugar especificado en GoTo.

GoTo [Linea]

Linea:

2.2.6 With End With

With End With se utiliza para ejecutar una serie de instrucciones sobre un único objeto o sobre un tipo definido por el usuario. Permite realizar múltiples operaciones sobre el mismo objeto.

```
With [Objeto]  
[Instrucciones]  
End With
```

2.2.7 If Then

If Then se utiliza para escoger entre 2 o más posibilidades, para decidir sobre que es lo que se quiere, entre una serie de alternativas.

```
If [Condicion] Then  
  
[Instrucciones]  
  
Elseif [CcondicionElseif] Then  
  
[InstruccionesElseif]  
  
Else  
  
[InstruccionesElse]  
  
End If
```

2.2.8 Select Case

Select Case se utiliza cuando las alternativas en una estructura de If Then son muchas. En tal caso conviene evitar el If Then y hacer uso de Select Case, que permite ejecutar uno de varios grupos de instrucciones, dependiendo del valor de una expresión.

```
Select Case [Expresion-m]
```

```
Case [Expresion-1]
```

```
[Instrucciones-1]
```

```
Case [Expresion-2]
```

```
[Instrucciones-2]
```

```
.....
```

```
Case [Expresion-n]
```

```
[Instrucciones-n]
```

```
Case Else
```

```
[InstruccionesElse]
```

```
End Select
```

2.2.9 DoEvents

DoEvents se utiliza para ceder el control de la ejecución al sistema operativo, para que éste pueda procesar otros eventos. El control no se devuelve hasta que el sistema operativo haya terminado de procesar los eventos en cola y se hayan enviado todas las teclas de la cola SendKeys.

```
DoEvents()
```

2.2.10 On Error GoTo

On Error GoTo se utiliza para controlar un error de macro en caso de existir. Permite indicar que en caso de existir un error haga un salto a otra línea donde se le indique cómo proceder.

On Error GoTo CasoError

CasoError:

2.2.11 On Error Resume Next

On Error Resume Next se utiliza para controlar un error de macro en caso de existir. Permite indicar que en caso de existir un error lo omita.

3 ARRAYS

Un Array corresponde a una cadena de elementos. Cada elemento de un array tiene un número de índice que lo identifica como único.

3.1 ARRAYS FIJOS

Un array fijo corresponde a aquel que posee un número fijo de elementos. La declaración de un Array fijo de 'n' elementos se hace de las siguientes formas.

Dim NombreArray(0 to n) as TipoVariable

Dim NombreArray(n) as TipoVariable

3.2 ARRAYS VARIABLES

Un array variable corresponde a aquel que puede modificarse el número de elementos. La declaración de un Array variable se hace de la siguiente forma.

Dim NombreArray() as TipoVariable

3.3 ARRAYS BIDIMENSIONALES

Un array bidimensional corresponde a aquel que posee 2 dimensiones, es decir, en vez de cadena corresponde a una matriz de elementos. La declaración de un Array bidimensional fijo de 'n' y 'm' elementos se hace de las siguientes formas.

Dim NombreArray(1 to n, 1 to m) as TipoVariable

Dim NombreArray(n, m) as TipoVariable

3.4 FUNCIONES DE ARRAY

3.4.1 Redim

Redimensiona el Array borrando la información que tenía guardada. La función Redim se usa de la siguiente forma.

Redim NombreArray(n)

3.4.2 Redim Preserve

Redimensiona el Array conservando la información que tenía guardada. La función Redim Preserve se usa de la siguiente forma.

Redim Preserve NombreArray(n)

3.4.3 LBound

Devuelve el índice inferior del array. La función LBound se usa de la siguiente forma.

LBound(NombreArray)

3.4.4 UBound

Devuelve el índice superior del array. La función UBound se usa de la siguiente forma.

UBound(NombreArray)

3.4.5 Join

Devuelve un String que contiene todos los elementos del Array separados por un elemento separador. La función Join se usa de la siguiente forma.

Join(NombreArray, "Separador")

3.4.6 Split

Devuelve un Array cuyos elementos corresponden a los que conforman un String separados por un elemento separador. La función Split se usa de la siguiente forma.

Split(NombreArray, "Separador")

3.4.7 Filter

Devuelve un array filtrado según algún criterio de búsqueda, permitiendo incluir o excluir el texto de búsqueda (True/False). La función Filter se usa de la siguiente forma.

Filter(NombreArray, "TextoBusqueda", True/False)

4 USERFORM

4.1 CUADRO DE HERRAMIENTAS



Las opciones y sus funcionalidades son las siguientes.



Seleccionar Objetos: Permite seleccionar controles insertos en el Userform.



Label: Permite insertar una etiqueta.



Textbox: Permite insertar un cuadro de texto, en el cual se introducen datos.



ComboBox: Sirve para que un usuario elija una opción de una lista.



ListBox: Sirve para que un usuario rellene o elija varias opciones de una lista.



CheckBox: Permite insertar una caja de chequeo.



OptionButton: Permite insertar botones de opciones.



ToggleButton: Sirve para activar o desactivar alguna funcionalidad. Este botón adopta el modo "Encendido" / "Apagado".



Frame: Sirve para agrupar elementos de un Userform (los elementos se deben ubicar dentro del Frame).



CommandButton: Es un simple botón que nos permite ejecutar acciones.



TabStrip: en un mismo Userform se pueden crear distintas secciones.



MultiPage: en un mismo Userform se pueden crear distintas páginas.



ScrollBar: si tenemos una lista con muchos elementos el scrollbar nos permite navegarlos.



SpinButton: permite aumentar o disminuir valores.



Image: permite introducir imágenes en el Userform.



RefEdit: permite hacer referencia a una celda de Excel.

4.2 FUNCIONES DE USERFORM

4.2.1 Load

Carga el Userform para ser mostrado en pantalla. La función Load se usa de la siguiente forma.

Load NombreUserForm

5 DECLARACIONES

5.1 OPTION

5.1.1 Option Explicit

Obliga a declarar explícitamente todas las variables mediante las instrucciones Dim, Private, Public, ReDim o Static. Cuando no se usa, todas las variables no declaradas son Variant.

5.1.2 Option Private Module

Impide que las variables declaradas en el modulo sean utilizadas en otros archivos. Las variables declaradas como publicas dentro del modulo si están disponibles en otros modulos del mismo archivo, pero no en modulos de otros archivos.

5.1.3 Option Base 0

Establece el 0 como el índice inferior de los Array declarados en el modulo. Esta opción viene por default, por lo que no es necesario declararla.

5.1.4 Option Base 1

Establece el 1 como el índice inferior de los Array declarados en el modulo.

5.1.5 Option Compare Binary

Especifica el método de comparación de Strings para un módulo. Esta opción viene por default, por lo que no es necesaria declararla. El ordenamiento de menor a mayor Binary es el siguiente.

A <B <E <Z <a <b <e <z

5.1.6 Option Compare Text

Especifica el método de comparación de Strings para un módulo. El ordenamiento de menor a mayor Text es el siguiente.

(A = a) <(=) <(B = b) <(E = e) <(=) <(Z = z) <(=)

5.2 EVENTOS DE LIBROS

Los eventos de libros corresponden a determinados sucesos propios de un libro, tales como abrirlo, guardarlo, cerrarlo, imprimirlo, etc. Las macros de eventos de libros se deben escribir en la carpeta ThisWorkbook de la ventana proyecto.

En la primera lista desplegable de la ventana código se debe seleccionar la opción Workbook.

En la segunda lista desplegable de la ventana código se debe seleccionar el evento de libro deseado.

5.2.1 Activate

Se ejecuta al activar el libro.

5.2.2 AddinInstall

Se ejecuta cuando el libro se instala como un complemento.

5.2.3 AddinUninstall

Se ejecuta cuando el libro se desinstala como un complemento.

5.2.4 AfterXmlExport

Se ejecuta después de guardar o exportar datos del libro a un archivo XML.

5.2.5 AfterXmlImport

Se ejecuta después de actualizar una conexión de datos XML o después de importar nuevos datos XML en el libro.

5.2.6 BeforeClose

Se ejecuta antes de cerrar el libro.

5.2.7 BeforePrint

Se ejecuta antes de imprimir el libro.

5.2.8 BeforeSave

Se ejecuta antes de guardar el libro.

5.2.9 BeforeXmlExport

Se ejecuta antes de guardar o exportar datos del libro a un archivo XML.

5.2.10 BeforeXmlImport

Se ejecuta antes de actualizar una conexión de datos XML o después de importar nuevos datos XML en el libro.

5.2.11 Deactivate

Se ejecuta cuando se desactiva el libro.

5.2.12 NewSheet

Se ejecuta cuando se crea una hoja en el libro.

5.2.13 Open

Se ejecuta al abrir el libro.

5.2.14 PivotTableCloseConnection

Se ejecuta después de que un informe de tabla dinámica cierre la conexión con su origen de datos.

5.2.15 PivotTableOpenConnection

Se ejecuta después de que informe de tabla dinámica abra la conexión con su origen de datos.

5.2.16 RowsetComplete

Se ejecuta cuando se navega por el conjunto de registros o se llama a la acción de conjunto de filas en una tabla dinámica OLAP.

5.2.17 SheetActivate

Se ejecuta cuando se activa una hoja del libro.

5.2.18 SheetBeforeDoubleClick

Se ejecuta cuando se hace doble click en una hoja, y antes de ejecutarse la acción de doble click predeterminada.

5.2.19 SheetBeforeRightClick

Se ejecuta cuando se hace click con el botón derecho en una hoja, y antes de ejecutarse la acción de click derecho predeterminada.

5.2.20 SheetCalculate

Se ejecuta después de recalcularse una hoja o después de que cualquier dato cambiado se trace en un gráfico.

5.2.21 SheetChange

Se ejecuta cuando cambian las celdas de una hoja.

5.2.22 SheetDesactivate

Se ejecuta cuando se desactiva una hoja.

5.2.23 SheetFolowHyperlink

Se ejecuta al hacer click en un hipervínculo del libro.

5.2.24 SheetPivotTableUpdate

Se ejecuta después de que se actualice la hoja de un informe de tabla dinámica.

5.2.25 SheetSelectionChange

Se ejecuta cuando cambia la selección en una hoja. No se ejecuta si la selección se encuentra en una hoja de gráficos.

5.2.26 Sync

Se ejecuta cuando la copia local de una hoja que forma parte de un área de documentos se sincroniza con la copia en el servidor.

5.2.27 WindowActivate

Se ejecuta cuando se activa una ventana del libro.

5.2.28 WindowDesactivate

Se ejecuta cuando se desactiva una ventana del libro.

5.2.29 WindowResize

Se ejecuta cuando cambia el tamaño de una ventana del libro.

5.3 EVENTOS DE HOJAS

Los eventos de hojas corresponden a determinados sucesos propios de una hoja, tales como activarla, desactivarla, calcularla, etc. Los macros de eventos de hojas se deben escribir en la carpeta del nombre de la hoja de la ventana proyecto.

En la primera lista desplegable de la ventana código se debe seleccionar la opción Worksheet.

En la segunda lista desplegable de la ventana código se debe seleccionar el evento de hoja deseado.

5.3.1 Activate

Se ejecuta al activar la hoja.

5.3.2 BeforeDoubleClick

Se ejecuta cuando se hace doble click en la hoja, y antes de ejecutarse la acción de doble click predeterminada.

5.3.3 BeforeRightClick

Se ejecuta cuando se hace click con el botón derecho en la hoja, y antes de ejecutarse la acción de click derecho predeterminada.

5.3.4 Calculate

Se ejecuta al calcular la hoja.

5.3.5 Change

Se ejecuta al producirse un cambio en la hoja.

5.3.6 Deactivate

Se ejecuta al desactivar la hoja.

5.3.7 FollowHyperLink

Se ejecuta al hacer click en un hipervínculo de la hoja.

5.3.8 PivotTableUpdate

Se ejecuta después de que se actualice la hoja de un informe de tabla dinámica de la hoja.

5.3.9 SelectionChange

Se ejecuta al moverse entre las celdas de la hoja.

6 OPM

6.1 Application

6.1.1 .Caption

Establece el título de la ventana del programa. La propiedad Caption se usa de la siguiente forma.

Application.Caption="TítuloVentana"

6.1.2 .Path

Devuelve la ruta de la carpeta donde está instalado el programa. La propiedad Path se usa de la siguiente forma.

Application.Path="RutaCarpetaPrograma"

6.1.3 .ScreenUpdating

Activa o desactiva la actualización instantánea de las ventanas del programa. La propiedad ScreenUpdating se usa de la siguiente forma.

Application.ScreenUpdating=True/False

6.1.4 .WindowState

Establece el estado de la ventana del programa. La propiedad WindowState se usa de la siguiente forma.

Application.WindowState=XINormal/XIMaximized/XIMinimized

6.1.5 .DisplayStatusBar

Activa o desactiva la visualización de la barra de estado. La propiedad DisplayStatusBar se usa de la siguiente forma.

Application.DisplayStatusBar=True/False/"Mensaje"

6.1.6 .DisplayFormulaBar

Activa o desactiva la visualización de la barra de fórmula. La propiedad DisplayFormulaBar se usa de la siguiente forma.

Application.DisplayFormulaBar=True/False

6.1.7 .Calculate

Recalcula todas las formulas de todas las hojas de todos los libros abiertos. El método Calculate se usa de la siguiente forma.

Application.Calculate

6.1.8 .Help

Muestra un tema de ayuda. El método Help se usa de la siguiente forma.

Application.Help Arg1:=HelpFile[String] Arg2:=HelpContextId[Integer]

6.1.9 .Run

Ejecuta una subrutina con sus argumentos respectivos. El método Run se usa de la siguiente forma.

Application.Run Arg1:=Macro[String] Arg2:=Arg[String] Arg3:=Arg[String].....

6.1.10 .EnableEvents

Activa o desactiva los eventos existentes en el libro u hoja. La propiedad EnableEvents se usa de la siguiente forma.

Application.EnableEvents = True/False

6.1.11 .DisplayAlerts

Activa o desactiva la aparición de mensajes de alerta. La propiedad DisplayAlert se usa de la siguiente forma.

Application.DisplayAlerts = True/False

6.1.12 .Quit

Cierra el Excel. El método Quit se usa de la siguiente forma.

Application.Quit

6.1.13 .Calculation

Establece la forma de cálculo de las fórmulas. La propiedad Calculation es la siguiente.

Application.Calculation = xlAutomatic/xlSemiAutomatic/xlManual

6.1.14 .CutCopyMode

Mantiene o borra el portapapeles luego de copiar y pega un elemento. La propiedad CutCopyMode es la siguiente.

Application.CutCopyMode=True/False

6.2 WorkBooks

Los objetos de Libros son los siguientes.

ThisWorkbook	Libro que contiene la macro
ActiveWorkbook	Libro Activo
Workbooks("NombreLibro.xlsx")	Libro de nombre NombreLibro.xlsx
Workbooks(1)	Primer libro abierto

A modo de ejemplo se utilizará como objeto "Workbooks" en reemplazo de cualquiera de los casos anteriores.

6.2.1 .Name

Devuelve el nombre del libro. La propiedad Name se usa de la siguiente forma.

Workbooks.Name="NombreLibro"

6.2.2 .Path

Devuelve la ruta de la carpeta donde se encuentra el libro. La propiedad Path se usa de la siguiente forma.

Workbooks.Path="RutaLibro"

6.2.3 .Author

Devuelve el nombre del autor del libro. La propiedad Author se usa de la siguiente forma.

Workbooks.Author="AutorLibro"

6.2.4 .Saved

Devuelve True o False dependiendo de si se han guardado o no los últimos cambios del libro. La propiedad Saved se usa de la siguiente forma.

Workbooks.Saved=True/False

6.2.5 .FullName

Devuelve el nombre completo del libro, es decir, la ruta mas el nombre del libro. La propiedad FullName se usa de la siguiente forma.

Workbooks.FullName

6.2.6 .Open

Abre un libro existente. El método Open se usa de la siguiente forma.

**Workbooks.Open Filename:="RutaLibro" ReadOnly:=True/False Password:="Password"
WriteResPassword:="Password" UpdateLinks:=True/False**

6.2.7 .Activate

Activa el libro. El método Activate se usa de la siguiente forma.

Workbooks.Activate

6.2.8 .Close

Cierra el libro. El método Close se usa de la siguiente forma.

Workbooks.Close SaveChanges:=True/False FileName:="NombreLibro"

6.2.9 .Add

Crea un libro en blanco y lo agrega a la colección. El método Add se usa de la siguiente forma.

Workbooks.Add

6.2.10 .Protect

Protege el libro, estructura y ventanas. El método Protect se usa de la siguiente forma.

**Workbooks.Protect Structure:=True Windows:=True
Workbooks.Unprotect ("Password")
Workbooks.Protect ("Password")**

6.2.11 .Unprotect

Desprotege el libro. El método Unprotect se usa de la siguiente forma.

Workbooks.Unprotect("Password")

6.2.12 .Save

Guarda el libro con el nombre actual. El metodo Save se usa de la siguiente forma.

Workbooks.Save

6.2.13 .SaveAs

Guarda el libro con un nuevo nombre. El método SaveAs se usa de la siguiente forma.

**Workbooks.SaveAs Filename:="NombreLibro"
FileFormat:=xlTemplate/xlNormal/xlHtml/xlUnicodeText**

6.2.14 .SaveCopyAs

Guarda una copia del libro con un Nuevo nombre. El método SaveCopyAs se usa de la siguiente forma.

Workbooks.SaveCopyAs Filename:="NombreLibro"

6.2.15 .Delete

Borra un libro. El método Delete se usa de la siguiente forma.

Workbooks.Delete

6.2.16 .Calculate

Recalcula todas las formulas de todas las hojas del libro. El método Calculate se usa de la siguiente forma.

Workbooks.Calculate

6.2.17 Kill

Borra un libro. El método Kill se usa de la siguiente forma.

Kill "RutaNombreLibro"

6.3 Worksheets

Los objetos de Hojas son los siguientes.

ActiveSheet	Hoja activa
Worksheets("NombreHoja")	Hoja de nombre NombreHoja
Worksheets(1)	Primera hoja del libro
Sheets("NombreHoja")	Hoja de nombre NombreHoja
Sheets(1)	Primera hoja del libro

A modo de ejemplo se utilizará como objeto "Sheets" en reemplazo de cualquiera de los casos anteriores.

6.3.1 .UsedRange

Devuelve el rango de datos utilizado en la hoja. La propiedad UsedRange se usa de la siguiente forma.

Range.UsedRange

6.3.2 .Tab

Devuelve la pestaña de la hoja. La propiedad Tab se usa de la siguiente forma.

Sheets.Tab

6.3.3 .Name

Devuelve el nombre de la hoja. La propiedad Name se usa de la siguiente forma.

Sheets.Name = "NombreHoja"

6.3.4 .Index

Devuelve el índice numérico de la hoja dentro de la colección de hojas. La propiedad Index se usa de la siguiente forma.

Sheets.Index=NumeroHoja

6.3.5 .Count

Devuelve el número de hojas de la colección. La propiedad Count se usa de la siguiente forma.

Sheets.Count=CantidadHojas

6.3.6 .Next

Selecciona la hoja siguiente. La propiedad Next se usa de la siguiente forma.

Sheets.Next

6.3.7 .Previous

Selecciona la hoja previa. La propiedad Next se usa de la siguiente forma.

Sheets.Previous

6.3.8 .Activate

Activa la hoja. El método Activate se usa de la siguiente forma.

Sheets.Activate

6.3.9 .Calculate

Recalcula todas las formulas de la hoja. El método Calculate se usa de la siguiente forma.

Sheets.Calculate

6.3.10 .Delete

Borra la hoja. El método Delete se usa de la siguiente forma.

Sheets.Delete

6.3.11 .Protect

Protege la hoja. El método Protect se usa de la siguiente forma.

**Sheets.Protect DrawingObjects:=True/False Contents:=True/False Scenarios:=True/False
AllowFormattingCells:=True/False AllowFormattingColumns:=True/False
AllowFormattingRows:=True/False AllowInsertingColumns:=True/False
AllowInsertingRows:=True/False AllowInsertingHyperlinks:=True/False
AllowDeletingColumns:=True/False AllowDeletingRows:=True/False
AllowSorting:=True/False AllowFiltering:=True/False AllowUsingPivotTables:=True/False
Password:="Password"**

6.3.12 .Unprotect

Desprotege la hoja. El método Unprotect se usa de la siguiente forma.

Sheets.Unprotect Password:="Password"

6.3.13 .Paste

Pega la hoja copiada. El método Paste se usa de la siguiente forma.

Sheets.Paste

6.3.14 .Add

Inserta una hoja o varias hojas a la colección de hojas existentes. El método Add se usa de la siguiente forma.

**Sheets.Add Before:=Hoja After:=Hoja Count:=CantidadHojas Type:=
xlSheetType/xlWorksheet/ xlChart/xlExcel4MacroSheet/xlExcel4IntlMacroSheet**

6.3.15 .Select

Selecciona la hoja. El método Select se usa de la siguiente forma.

Sheets.Select

6.3.16 .Move

Mueve la hoja a otra posición dentro de la colección. El método Move se usa de la siguiente forma.

Sheets.Move Before:=Hoja After:=Hoja

6.3.17 .Copy

Copia la hoja en una determinada posición dentro de la colección. El método Copy se usa de la siguiente forma.

Sheets.Copy Before:=Hoja After:=Hoja

6.3.18 .Visible

Muestra u oculta la hoja. La propiedad Visible se usa de la siguiente forma.

Sheets.Visible=False/True/xlSheetVeryHidden

6.3.19 .PrintOut

Imprime la hoja. El método PrintOut se usa de la siguiente forma.

Sheets.PrintOut

6.3.20 .PageSetup

Devuelve la configuración de la hoja. La propiedad PageSetup se usa de la siguiente forma.

Sheets.PageSetup

6.3.21 .PageSetup.PrintArea

Establece el área de impresión de la hoja. La propiedad PageSetup.PrintArea se usa de la siguiente forma.

Sheets.PageSetup.PrintArea=DireccionRango

6.3.22 .PageSetup.LeftHeader

Establece el encabezado de la hoja. La propiedad PageSetup.LeftHeader se usa de la siguiente forma.

Sheets.PageSetup.LeftHeader="TextoEncabezado"

6.3.23 .PageSetup.LeftFooter

Establece el pie de página de la hoja. La propiedad PageSetup.LeftFooter se usa de la siguiente forma.

Sheets.PageSetup.LeftFooter="TextoPiePagina"

6.4 Range

Los objetos de Rangos son los siguientes.

ActiveCell	Celda activa
Range("A1")	Celda A1
Range("A1:C3")	Rango A1:C3
Cells(1,1)	Celda A1
Range(Cells(1,1),Cells(3,3))	Rango A1:C3
Range("NombreRango")	Rango llamado NombreRango
Selection	Celda o Rango seleccionado
Rows("A1")	Fila de la celda A1
Rows("A1:A3")	Filas del rango A1:A3
Columns("A1")	Columna de la celda A1
Columns("A1:C3")	Columnas del rango A1:C3
[A1]	Celda A1
[A1:C3]	Rango A1:C3

A modo de ejemplo se utilizará como objeto "Range" en reemplazo de cualquiera de los casos anteriores.

6.4.1 .EntireRow

Devuelve la fila completa del rango. La propiedad EntireRow se usa de la siguiente forma.

Range.EntireRow

6.4.2 .EntireColumn

Devuelve la columna completa del rango. La propiedad EntireColumn se usa de la siguiente forma.

Range.EntireColumn

6.4.3 .Dependents

Devuelve el rango de celdas dependientes del rango. La propiedad Dependents se usa de la siguiente forma.

Range.Dependents

6.4.4 .SpecialCells

Devuelve el rango de celdas con una determinada característica. El método SpecialCells se usa de la siguiente forma.

Range.SpecialCells

Type:=xlCellTypeAllFormatConditions/xlCellTypeAllValidation/xlCellTypeBlanks/xlCellTypeComments/xlCellTypeConstants/xlCellTypeFormulas/xlCellTypeLastCell/xlCellTypeSameFormatConditions/xlCellTypeSameValidation/xlCellTypeVisible
Value:=xlErrors/xlLogical/xlNumbers/xlTextValues

6.4.5 .End

Devuelve el rango de celdas comprendido entre la celda actual y un extremo. El método End se usa de la siguiente forma.

Range.End Direction:=xlToRight/xlDown/xlToLeft/xlUp

6.4.6 .CurrentRegion

Devuelve el rango de celdas correspondiente a aquel con contenido entorno al rango actual. La propiedad CurrentRegion se usa de la siguiente forma.

Range.CurrentRegion

6.4.7 .Count

Devuelve la cantidad de celdas del rango. La propiedad Count se usa de la siguiente forma.

Range.Count=CantidadCeldas

6.4.8 .FormulaR1C1

Inserta una formula en el rango con el format R1C1. La propiedad FormulaR1C1 se usa de la siguiente forma.

Range.FormulaR1C1 = "=Formula"

6.4.9 .Name

Establece el nombre del rango. La propiedad Name se usa de la siguiente forma.

Range.Name = "NombreRango"

6.4.10 .Value

Devuelve o asigna el valor de una celda o rango. La propiedad Value se usa de la siguiente forma.

Range.Value=Valor

6.4.11 .Formula

Devuelve o asigna la formula de una celda o rango. La propiedad Formula se usa de la siguiente forma.

Range.Formula="=Formula"

6.4.12 .Text

Devuelve o asigna el valor de una celda o rango como texto. La propiedad Text se usa de la siguiente forma.

Range.Text="Valor"

6.4.13 .Insert

Inserta un rango. El método Insert se usa de la siguiente forma.

Range.Insert

6.4.14 .Delete

Elimina un rango. El método Delete se usa de la siguiente forma.

Range.Delete

6.4.15 .Address

Devuelve un String con la dirección del rango. La propiedad Address se usa de la siguiente forma.

Range.Address

6.4.16 .HasFormula

Devuelve True o False dependiendo de si el rango tiene o no fórmulas. La propiedad HasFormula se usa de la siguiente forma.

Range.HasFormula

6.4.17 .NumberFormat

Devuelve o asigna el String que representa el formato de un rango. La propiedad NumberFormat se usa de la siguiente forma.

Range.NumberFormat="NumeroFormato"

6.4.18 .Offset

Desplaza la selección un número determinado de filas y columnas desde el rango seleccionado. La propiedad Offset se usa de la siguiente forma.

Range.Offset(NumeroFilas, NumeroColumnas)

6.4.19 .Cells

Desplaza la selección un número determinado de filas y columnas desde el rango seleccionado. La propiedad Cells se usa de la siguiente forma.

Range.Cells(NumeroFilas, NumeroColumnas)

6.4.20 .HorizontalAlignment

Establece la alineación horizontal del contenido de las celdas del rango. La propiedad HorizontalAlignment se usa de la siguiente forma.

Range.HorizontalAlignment=XlCenter/XlLeft/XlRight

6.4.21 .Select

Selecciona el rango. El método Select se usa de la siguiente forma.

Range.Select

6.4.22 .Activate

Activa el rango. El método Activate se usa de la siguiente forma.

Range.Activate

6.4.23 .Calculate

Recalcula todas las formulas del rango. El método Calculate se usa de la siguiente forma.

Range.Calculate

6.4.24 .Copy

Copia el rango y lo pega en otro especificado. El método Copy se usa de la siguiente forma.

Range.Copy Destination:=RangoPegado

6.4.25 .Delete

Elimina el rango y desplaza las celdas según indicación. El metodo Delete se usa de la siguiente forma.

Range.Delete Shift:=XlShiftToLeft/XlShiftToUp

6.4.26 .Clear

Borra todo del rango. El método Clear se usa de la siguiente forma.

Range.Clear

6.4.27 .ClearComments

Borra los comentario del rango. El método ClearComments se usa de la siguiente forma.

Range.ClearComments

6.4.28 .ClearFormats

Borra los formatos del rango. El método ClearFormats se usa de la siguiente forma.

Range.ClearFormats

6.4.29 .ClearContents

Borra el contenido del rango. El método ClearContents se usa de la siguiente forma.

Range.ClearContents

6.4.30 .ClearNotes

Borra las notas del rango. El método ClearNotes se usa de la siguiente forma.

Range.ClearNotes

6.4.31 .MergeCells

Transforma el rango en una sola celda. El método MergeCells se usa de la siguiente forma.

Range.MergeCells = True/False

6.4.32 .Hidden

Establece la visibilidad del rango. La propiedad Hidden se usa de la siguiente forma.

Range.Hidden = True/False

6.4.33 .Locked

Establece el bloqueo de un rango. La propiedad Locked se usa de la siguiente forma.

Range.Locked = True/False

6.4.34 .PasteSpecial

Pega el rango según algún atributo especial. El método PasteSpecial se usa de la siguiente forma.

Range.PasteSpecial

Paste:=xlPasteAll/xlPasteValuesAndNumberFormats/xlPasteValues/xlPasteValidation/xlPasteFormulasAndNumberFormats/xlPasteFormulas/xlPasteComments/xlPasteColumnWidths/xlPasteAllExceptBorders/xlPasteAll Transpose:=True/False

6.4.35 .Cut

Corta el rango. El método Cut se usa de la siguiente forma.

Range.Cut

6.4.36 .Sort

Ordena el rango según algún criterio. El método Sort se usa de la siguiente forma.

Range.Sort Key1:=Range Order1:=xlAscending/xlDescending Header:=xlGuess

6.4.37 .ScrollArea

Establece el área permitida de desplazamiento para el Scroll. La propiedad ScrollArea se usa de la siguiente forma.

Sheets.ScrollArea = "Rango"

6.4.38 .AutoFilter

Filtra un rango de datos segun el criterio de alguna columna. El método AutoFilter se usa de la siguiente forma.

**Range.AutoFilter Field:=NumeroCampoFiltro Criteria1:="Criterio"
Operator:=xlAnd/xlBottom10Items/xlBottom10Percent/xlOr/xlTop10Items/xlTop10Percent VisibleDropDown:=True/False**

6.4.39 .Font

Devuelve la fuente del texto del rango. La propiedad Font se usa de la siguiente forma.

Range.Font

6.4.40 .Font.Bold

Establece si la fuente es negrita o no. La propiedad Font.Bold se usa de la siguiente forma.

Range.Font.Bold=True/False

6.4.41 .Font.Size

Establece el tamaño de la fuente. La propiedad Font.Size se usa de la siguiente forma.

Range.Font.Size=TamañoFuente

6.4.42 .Interior

Devuelve el interior de las celdas del rango. La propiedad Interior se usa de la siguiente forma.

Range.Interior

6.4.43 .Interior.ColorIndex

Establece el color de las celdas del rango. La propiedad Interior.ColorIndex se usa de la siguiente forma.

Range.Interior.ColorIndex=NumeroColor

6.5 ACTIVEWINDOW

6.5.1 .ScrollRow

Establece la fila que aparece en la parte superior de la ventana. La propiedad ScrollRow se usa de la siguiente forma.

ActiveWindow.ScrollRow = NumeroFila

6.5.2 .ScrollColumn

Establece la columna que aparece en la parte izquierda de la ventana. La propiedad ScrollColumn se usa de la siguiente forma.

ActiveWindow.ScrollColumn = NumeroColumna

6.6 NAMES

6.6.1 .Add

Agrega un nombre al administrador de nombres del libro. El método Add se usa de la siguiente forma.

Names.Add Name:="Nombre", RefersTo:=Valor

6.7 MSGBOX

Muestra una caja de mensaje con un boton para Aceptar. Adicionalmente se puede agregar un boton Cancelar. El MsgBox se usa de la siguiente forma.

Msgbox ("MensajeMsgbox")

6.8 INPUTBOX

Muestra una caja de ingreso de datos. El InputBox se usa de la siguiente forma.

**Nombre = InputBox Prompt:="Input" Title:="Titulo" Default:="ValorDefault"
Type:=1/2/4/5**