

# OBJETOS DE EXCEL EN VISUAL BASIC

**Jerarquía de Objetos:** **Application** → **Workbooks(Workbook)** → **Worksheets(Worksheet)** → **Range**

Ejemplo: **Application.Workbooks("NombreLibro.xls").Worksheets("NombreHoja").Range("C2").Propiedad**  
**Application.Workbooks(ÍndiceLibro).Worksheets(ÍndiceHoja).Range("NombreRango").Método**

## Application

El propio Excel.

<u>Propiedades</u>	[Tipo, Lectura/Escritura]	Sintaxis: <b>Application.Propiedad</b> = Valor
<b>.Caption</b>	[String, L/E]	Título de la Ventana Principal de Excel.
<b>.Path</b>	[String, L]	Devuelve la Carpeta donde está instalado Excel.
<b>.WindowState</b>	[L/E]	<b>xlNormal</b> / <b>xlMaximized</b> / <b>xlMinimized</b> . Estado de la ventana.
<b>.DisplayAlerts</b>	[Boolean, L/E]	<b>True</b> / <b>False</b> . Muestra o no los mensajes de alerta.
<b>.ScreenUpdating</b>	[Boolean, L/E]	<b>True</b> / <b>False</b> . Actualiza o no la ventana a medida que se ejecuta la subrutina.
<b>.DisplayStatusBar</b>	[Boolean, L/E]	<b>True</b> / <b>False</b> . Muestra u oculta la barra de estado.
<b>.DisplayFormulaBar</b>	[Boolean, L/E]	<b>True</b> / <b>False</b> . Muestra u oculta la barra de fórmulas.
<b>.StatusBar</b>	[String, L/E]	Devuelve o establece el texto de la barra de estado. ( <b>False</b> si Excel tiene el control).

<u>Métodos</u>	<u>Argumentos</u>	Sintaxis: <b>Application.Método Arg1:=Valor, Arg2:="Cadena", ...</b>
<b>.Calculate</b>	Ninguno	[] Recalcula todas las fórmulas, de todas las hojas, de todos los libros abiertos.
<b>.Help</b>	<b>HelpFile, HelpContextId</b>	[String, Integer] Muestra un tema de ayuda.
<b>.Quit</b>	Ninguno	[] Cierra Excel.
<b>.Run</b>	<b>Macro, arg1, arg2, ...</b>	[String, ...] Ejecuta un Procedimiento con sus argumentos si los lleva.

## Workbooks

Libros de trabajo de Excel. (Los \*.xls)

<b>ThisWorkbook</b>	Libro que contiene la macro que se ejecuta.
<b>ActiveWorkbook</b>	Libro activo. (Ejemplo: NombreLibro = <b>ActiveWorkbook.Name</b> )
<b>Workbooks("Libro1.xls")</b>	Referencia a un libro por su nombre. Libro llamado "Libro1.xls". ;Debe estar abierto!
<b>Workbooks(1)</b>	Referencia a un libro por su Índice. Primer libro de la colección. Primer libro abierto.
<b>Workbooks(Workbooks.Count)</b>	Último libro de la colección. Último libro abierto. (Workbooks.Count es la cantidad de libros).

<u>Propiedades</u>	[Tipo, L/E]	Sintaxis: <b>Workbooks(Workbook).Propiedad</b> = "Texto"
<b>Workbooks.Count</b>	[Integer, L]	Devuelve el Número de libros de la colección de libros abiertos. <i>Propiedad de la Colección.</i>
<b>.Name</b>	[String, L]	Devuelve el Nombre del Libro. (Para cambiarlo usar el método SaveAs).
<b>.Path</b>	[String, L]	Devuelve la Ruta de acceso al archivo del libro.
<b>.Author</b>	[String, L/E]	Nombre del Autor del libro.
<b>.Saved</b>	[Boolean, L]	<b>True</b> / <b>False</b> . Si es False se han hecho cambios en el libro, y no están salvados.

<u>Métodos</u>	<u>Argumentos</u>	Sintaxis: <b>Workbooks(Workbook).Método Arg1:=Valor, Arg2:=" Texto", ...</b>
----------------	-------------------	--

*de la Colección Workbooks*

<b>.Open</b>	<b>FileName, ReadOnly</b>	[String, Bool] Abre un libro existente: <b>FileName:="c:\dir\libro.xls"</b>
<b>.Add</b>	Ninguno	[] Crea un nuevo libro vacío y lo agrega a la colección.

*de un libro determinado*

<b>.Activate</b>	Ninguno	[] Activa el libro especificado.
<b>.Close</b>	<b>SaveChanges, FileName</b>	[Bool, String] Cierra el libro, y lo guarda con un nombre si True.
<b>.Protect</b>	<b>Password, Structure, Windows</b>	[String, Bool, Bool] Protege el libro, estructura y ventanas.
<b>.Save</b>	Ninguno	[] Guarda el libro, con el nombre actual.
<b>.SaveAs</b>	<b>FileName</b>	[String] Guarda el libro, con un nuevo nombre.
<b>.SaveCopyAs</b>	<b>FileName</b>	[String] Guarda una copia del libro, con un nuevo nombre.

## Worksheets

Colección de Hojas de Trabajo. (Las hojas de un libro Excel)

<b>ActiveSheet</b>	Hoja activa. (Ejemplo: <b>NombreHoja</b> = <b>ActiveSheet.Name</b> )
<b>Worksheets("Hoja1")</b>	Hoja llamada "Hoja1".
<b>Worksheets(1)</b>	Primera hoja de la colección. (Ej: <b>NombreHoja</b> = <b>Worksheets(1).Name</b> )
<b>Worksheets(Worksheets.Count)</b>	Última hoja de la colección. ( <i>Worksheets.Count</i> es la cantidad de hojas).

<u>Propiedades</u>	[Tipo, L/E]	Sintaxis: <b>Worksheets(Worksheet).Propiedad</b> = "Cadena"
<b>Worksheets.Count</b>	[Integer, L]	Propiedad de la Colección. Devuelve el número de hojas de la colección.
<b>.Index</b>	[Integer, L]	Devuelve el índice numérico de la hoja dentro de la colección de hojas.
<b>.Name</b>	[String, L/E]	Nombre de la Hoja.
<b>.Next</b>	[Worksheet, L]	Devuelve la siguiente hoja. ( <b>.Previous</b> ) la anterior.
<b>.UsedRange</b>	[Range, L]	Devuelve un objeto Range con el rango de la hoja que tiene datos.

<u>Métodos</u>	<u>Argumentos</u>	Sintaxis: <b>Worksheets(Worksheet).Método Arg1:=Valor, Arg2:="Cadena", ..</b>
<b>.Activate</b>	Ninguno	[] Activa la hoja especificada.
<b>.Calculate</b>	Ninguno	[] Recalcula todas las fórmulas de la hoja especificada.
<b>.Delete</b>	Ninguno	[] Elimina la hoja especificada.
<b>.Protect</b>	<b>Password</b>	[ <b>String</b> ] Protege la hoja de cálculo.
<b>.Paste</b>	<b>Destination</b>	[ <b>Range</b> ] Pega el rango copiado previamente en otro rango.

♣ **Range**

**Range("B3")**  
**Range("C2:D3")**  
**Range("NombreDeRango")**  
**ActiveCell**

Rangos dentro de una hoja de Trabajo. Son Colecciones de celdas.

Se refiere al rango indicado (Colección de una sola celda).  
Se refiere al rango indicado (Colección de varias celdas).  
Se refiere al rango indicado por medio de un nombre.  
Se refiere a la Celda Activa (**¡Ojo!** No el Rango activo). (Si hay activo un rango de varias celdas, se refiere a la Celda Superior Izquierda).

**Selection.Cells**  
Colección de celdas seleccionadas.

<u>Propiedades</u>	[ <b>Tipo</b> , L/E]	Sintaxis: <b>Range().Propiedad</b>
<b>.Count</b>	[ <b>Integer</b> , L]	Devuelve el número de celdas contenidas en el rango especificado.
<b>.Name</b>	[ <b>String</b> , L/E]	Nombre del rango.
<b>.Value</b>	[ <b>Cualquiera</b> , L/E]	El valor que tiene un rango. (Si son varias celdas devuelve una matriz) / ...de fórmulas
<b>.Formula</b>	[ <b>String</b> , L/E]	La fórmula contenida en el rango en forma de cadena de texto. ( <i>Lo que hay en la barra...</i> )
<b>.Text</b>	[ <b>String</b> , L]	El valor formateado del rango en forma de cadena de texto. ( <i>Lo que muestra la celda</i> )
<b>.NumberFormat</b>	[ <b>String</b> , L/E]	Formato numérico (Ejemplo: "[Blue]#,##0.00 €;[Red]-#,##0.00 €;""Gratis""")
<b>.Font</b>	[ <b>Object</b> , L/E]	Contiene los atributos de fuente ( <b>.Name</b> , <b>.size</b> , <b>.Color</b> , <b>.Bold</b> , <b>.Italic</b> , <b>.Underline</b> , etc)
<b>.Interior</b>	[ <b>Object</b> , L/E]	Se refiere al interior del Objeto ( <b>.Color</b> , <b>.Pattern</b> , <b>.PatternColor</b> , etc)
<b>.EntireRow</b>	[ <b>Range</b> , L]	Se refiere a la Fila o Filas Completas del rango.
<b>.EntireColumn</b>	[ <b>Range</b> , L]	Se refiere a la Columna o Columnas Completas del rango.
<b>.Offset(RowOffset, ColumnOffset)</b>		Se refiere a un rango desplazado tantas filas y columnas desde el rango especificado.
<b>.Row y .Column</b>	[ <b>Long</b> , L]	Devuelve el número de la primera fila (o Columna) de la primera área del rango.
<b>.Rows y .Columns</b>	[ <b>Range</b> , L]	Se refiere a la Colección de Filas (o Columnas) del rango especificado.
<b>.Dependents</b>	[ <b>Range</b> , L]	Devuelve un objeto Range con las celdas dependientes de rango. ( <b>.Precedents</b> )

<u>Métodos</u>	<u>Argumentos</u>	Sintaxis: <b>Range().Método Arg1:=Valor, Arg2:="Cadena", ...</b>
<b>.Select</b>	Ninguno	[] Selecciona el Rango (de una o más celdas).
<b>.Activate</b>	Ninguno	[] Activa una sola celda que si está en la selección actual no cambia la selección.
<b>.Calculate</b>	Ninguno	[] Recalcula todas las fórmulas del rango especificado.
<b>.Copy</b>	<b>Destination</b>	[ <b>Range</b> ] Copia el rango en otro rango. Si no se pone el destino, copia al portapapeles.
<b>.PasteSpecial</b>	[ <b>Varios Opcionales</b> ]	Pega el rango copiado. Se puede especificar lo pegado (Valores, Fórmulas, etc.).
<b>.Delete</b>	<b>Shift</b>	[ <b>xlShiftToLeft / xlShiftUp</b> ] Elimina el rango y desplaza según indicación.
<b>.ClearContents</b>	Ninguno	[] Borra los valores del rango, pero mantiene el formato, reglas, etc.

## Programación de Código en Visual Basic for Applications (VBA)

<b>Option Explicit</b>	' Fuerza la Declaración explícita de Variables con Dim.
<b>Option Base 1</b>	' Indica el índice base de los arrays. Por omisión 0.
<b>Option Compare {Binary   Text   Database}</b>	' Especifica el método de Comparación de Textos en el módulo.
<b>Dim Variable1 As Tipo</b>	' Declara una <b>Variable Global</b> del Módulo de tipo <b>Tipo</b> .
<b>Public Variable1 As Tipo</b>	' Declara una <b>Variable Pública</b> del Proyecto de tipo <b>Tipo</b> .
<b>Type TipoPersonal</b>	' Define un Tipo de Datos Personal en forma de Registro
Campo1 <b>As Tipo</b>	' con varios campos del tipo deseado.
Campo2 <b>As Tipo</b>	
<b>End Type</b>	' <-- Después de una comilla simple todo es un Comentario aclaratorio y aparece en verde.
<b>Sub NombreProcedimiento(ByVal Arg1 As Tipo, ByRef Arg2 As Tipo, Optional ByRef Arg3opc As Variant)</b>	
<b>Const</b> Constante1 <b>As Tipo = Valor</b>	' Declara una <b>Constante</b> y le asigna el valor <b>Valor</b> .
<b>Dim</b> Variable1 <b>As Tipo</b>	' Declara una <b>Variable Local</b> de tipo <b>Tipo</b> .
<b>Dim</b> Array1(5, 8) <b>As Tipo</b>	' Declara un <b>Array</b> de 5 x 8 datos del tipo <b>Tipo</b> . Hasta <b>60</b> dimensiones.
<b>Dim</b> Array1(3 to 7) <b>As Tipo</b>	' Declara un <b>Array</b> de 5 datos (del 3 al 7) del tipo <b>Tipo</b> .
Array1(3,7) = Valor	' Asigna <b>Valor</b> al elemento (3,7) del array <b>Array1</b> .
Variable1 = <b>NombreFunción(Arg1, Arg2)</b>	' Asigna a <b>Variable1</b> lo que devuelve <b>NombreFunción</b> .
<b>Set</b> Var_Objeto = {[ <b>New</b> ] <b>Objeto</b>   <b>Nothing</b> }	' Asigna una Referencia de Objeto a una Variable o Propiedad
<b>Call NombreProcedimiento2(Arg1, Arg2)</b>	' Ejecuta el Procedimiento <b>NombreProcedimiento2</b> .
<b>NombreProcedimiento2 Arg1, Arg2</b>	' Ejecuta el Procedimiento <b>NombreProcedimiento2</b> .
Instrucción : Otra Instrucción más larga que no cabe en la línea anterior	' El <b>Espacio + Subrayado</b> sirve para continuar en la línea siguiente. ' Y los <b>(:)</b> para escribir varias instrucciones en la misma línea.
<b>End Sub</b>	

**Function NombreFunción(ByVal Arg1 As Tipo, ByRef Arg2 As Tipo, Optional ByRef Arg3opc As Variant) As Tipo**

Instrucciones y Cálculos necesarios para obtener una Expresión\_Resultado ....

**NombreFunción** = Expresión\_Resultado ' La Función **NombreFunción** devolverá **Expresión\_Resultado**.

**End Function**

## Tipos de Datos

Nombre	Tamaño	Contenido	Rango del Contenido
<b>Boolean</b>	2 bytes		<b>True / False</b>
<b>String</b>	1 byte	Cadena	1 byte / carácter + 10 bytes de cabecera. Hasta 2.000 millones de caracteres.
<b>Byte</b>	1 byte	Bytes	<b>0.....255</b>
<b>Integer</b>	2 bytes	Enteros	<b>-32.768.....32.767</b> Enteros
<b>Long</b>	4 bytes	Enteros largos	<b>-2.147.483.648.....2.147.483.647</b>
<b>Single</b>	4 bytes	Reales	Simple Precisión en coma flotante. <b>1,401298E-45.....3,402823E38</b> (7 dígitos)
<b>Double</b>	8 bytes	Reales	Doble Precisión en coma flotante. <b>4,94065645841247E-324.....1,79769313486232E308</b>
<b>Currency</b>	8 bytes	Moneda	<b>-922.337.203.685.477,5808.....922.337.203.685.477,5807</b> (922 Billones) (15 + 4)
<b>Decimal</b>	14 bytes	Reales largos	<b>+/-79.228.162.514.264.337.593.543.950.335</b> (29 ó 1 + 28)
<b>Date</b>	8 bytes	Fecha y Hora	<b>1/1/100.....31/12/9999</b> ( -657.434.....2.958.465 )
<b>Object</b>	4 bytes	Objeto	Almacena cualquier <b>referencia a un objeto</b> de Excel.
<b>Variant</b>	16 bytes	Variante	El tipo <b>por defecto</b> si no se especifica otro. (Admite cualquiera de los anteriores).

## Estructuras de Control de Flujo

### Flujo Alternativo Condicional (If)

1	<b>If Condición Then</b> Instruc. Si Condición <b>Cierta</b> : Otra Instr.
2	<b>If Condición Then</b> Instrucciones Si Condición es <b>Cierta</b> <b>Else</b> 'Cláusula Opcional Instrucciones Si Condición es <b>Falsa</b> <b>End If</b>
3	<b>If Condición Then</b> Instrucciones Si Condición es <b>Cierta</b> <b>Else</b> <b>If Condición 2 Then</b> Instrucciones Si Condición2 es <b>Cierta</b> <b>End If</b> <b>End If</b>
4	<b>If Condición 1 Then</b> Instrucciones Si Condición 1 es <b>Cierta</b> <b>ElseIf Condición 2 Then</b> Instrucciones Si Condición 1 es <b>Falsa</b> y Condición 2 es <b>Cierta</b> <b>End If</b>

### Bucle para Recorrer Colecciones (For Each)

<b>For Each Elemento In Colección</b> Instrucciones <b>Next Elemento</b>	<i>Una iteración para cada elemento de la Colección</i>
--	---

### Bucle de Repetición Conocida (For Next)

<b>For Contador = 1 To 100 Step 2</b> 'Step es Opcional Instrucciones <b>Next Contador</b>	<i>En cada iteración la variable se incrementa lo que indique Step</i>
--	--

### Bucles de Repetición Desconocida (Do Loop)

0	<b>While Condición</b> Instrucciones <b>mientras</b> Condición sea <b>Cierta</b> <b>Wend</b>
1	<b>Do While Condición</b> Instrucciones <b>mientras</b> Condición sea <b>Cierta</b> <b>Loop</b>
2	<b>Do Until Condición</b> Instrucciones <b>hasta</b> que Condición sea <b>Cierta</b> <b>Loop</b>
3	<b>Do</b> Instruc. se repiten <b>mientras</b> Condición <b>Cierta</b> <b>Loop While Condición</b>
4	<b>Do</b> Instruc. se repiten <b>hasta</b> que Condición <b>Cierta</b> <b>Loop Until Condición</b>

**Nota:** La sintaxis nº 0 está en desuso a favor de las más modernas 1 a 4 (do ... loop). (La nº 0 es idéntica a la nº1)

### Flujo Alternativo Evaluativo (Select Case)

<b>Select Case Expresión</b> <b>Case Valor1</b> : Instrucciones si Expresión = Valor1 <b>Case Valor2, Valor3</b> Instrucciones si Expresión = Valor2 ó Expresión = Valor3 <b>Case 1 To 4, 9</b> : Instrucciones si Expresión = 1...4 ó 9 <b>Case 5 To 8</b> : Instrucciones si Expresión = 5...8 <b>Case Else</b> : Instrucciones Si Ninguno Anterior 'Opcional <b>End Select</b>
--

## Referencias Abreviadas

<p><b>With Objeto</b> <b>.Propiedad = Valor</b> ' Idéntico a <b>Objeto.Propiedad = Valor</b> <b>With .SubObjeto</b> <b>.Método = Valor</b> ' Idéntico a <b>Objeto.SubObjeto.Método = Valor</b> <b>End With</b> <b>End With</b></p>	<p><i>Nota: Una vez que se ha entrado en un bloque <b>With</b> no es posible cambiar el objeto. Por tanto, no se puede utilizar una única instrucción <b>With</b> para varios objetos.</i></p>
--	--