
Capítulo 3

La programación de objetos en Excel

1. Presentación

VBA Excel es un lenguaje de programación **orientado a objetos**, si bien no dispone de todas las funcionalidades de los lenguajes de este tipo.

La mayoría de los elementos que maneja Excel son objetos: los libros, las hojas de cálculo, los rangos de celdas, las celdas, etc.

Los objetos se organizan según un **modelo jerárquico**: ciertos objetos contienen otros objetos que pueden, a su vez, contener otros. Estos objetos se llaman **contenedor** u **objetos parent**. Por ejemplo, el objeto **Application** es un contenedor de objetos **Workbook** (libros abiertos en Excel), que a su vez contiene objetos **Worksheet** (hojas de cálculo de un libro). El contenedor principal es el objeto **Application**.

Un conjunto de objetos del mismo tipo constituye una **colección** (colección Workbooks: conjunto de libros abiertos en Excel; colección Worksheets: conjunto de hojas de cálculo de un libro).

Un objeto dispone de un conjunto de características llamadas **propiedades** (por ejemplo, para el objeto **Application**: la propiedad **UserName** representa el nombre del usuario, la propiedad **Version** devuelve el número de versión de Microsoft Excel) y de comportamientos o acciones llamados **métodos** (por ejemplo, para el objeto **Application**, el método **FindFile** muestra el cuadro de diálogo **Abrir**; el método **Quit** sale de Excel, etc.).

100 _____ VBA Excel (2024 y Microsoft 365)

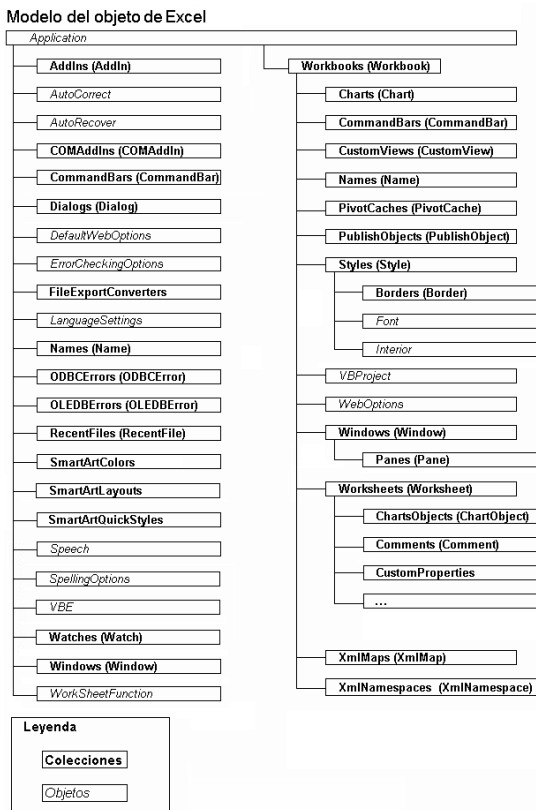
Programar en Excel: macros y lenguaje VBA

A un objeto le suceden **eventos** provocados por el usuario (por ejemplo: la apertura de un libro, un clic en un botón de comando, el cambio de la celda activa, etc.) o por el sistema.

Las **clases** son modelos que permiten crear objetos de un mismo tipo. Los objetos de una misma clase heredan sistemáticamente todos los métodos, propiedades y eventos de su clase. Es posible crear clases de objetos con VBA Excel usando módulos de clase.

2. El modelo de objeto de Excel

2.1 Presentación



2.2 Principales objetos y colecciones

Objetos

Application

Objeto que hace referencia a la aplicación de Microsoft Excel activa.

AutoCorrect

Objeto que representa los atributos de autocorrección de Microsoft Excel.

AutoRecover

Objeto que representa las opciones de recuperación automática de un libro. Estas macros son accesibles desde Excel a partir de la pestaña **Guardar** del menú **Herramientas - Opciones**.

DefaultWebOptions

Objeto que devuelve los atributos usados por Excel para la apertura o grabación de una página web.

ErrorCheckingOptions

Objeto que representa las opciones de comprobación de errores de la aplicación de Excel.

LanguageSettings

Objeto que contiene información sobre la configuración de idioma de Excel.

Speech

Objeto que contiene los métodos y propiedades que se relacionan con las funciones de síntesis de voz.

SpellingOptions

Objeto que representa las opciones de ortografía de la aplicación.

VBE

Objeto que representa al Editor de Visual Basic.

102 _____ VBA Excel (2024 y Microsoft 365)

Programar en Excel: macros y lenguaje VBA

WorksheetFunction

Objeto que contiene todas las funciones disponibles en Excel. Este objeto permite obtener el resultado de una función aplicada a un rango de celdas.
Ejemplo: `Prom = Application.WorksheetFunction.Average(Selection)`

Colecciones

AddIns

Colección que contiene todos los complementos (objetos AddIn).

COMAddIns

Representa los complementos COM actualmente instalados en Microsoft Excel.

CommandBars

Colección de barras de herramientas de la aplicación activa (objetos CommandBar).

Dialogs

Colección de los cuadros de diálogo integrados de Excel.

FileExportConverters

Colección de los convertidores de archivos para guardar libros de Excel.

Names

Colección de todos los nombres (celdas y rangos con nombre) del libro activo.

ODBCErrors

Colección de todos los errores ODBC generados por la última operación efectuada en un informe de tabla dinámica o en una tabla de consulta.

OLEDBErrors

Colección que representa la información relacionada con el error devuelto por la consulta OLE DB más reciente.

RecentFiles

Colección de los últimos archivos usados.

SmartArtColors

Colección de los estilos de colores SmartArt cargados actualmente en la aplicación.

SmartArtLayouts

Colección de los diagramas de diseño cargados actualmente en la aplicación.

SmartArtQuickStyles

Colección de los estilos rápidos de SmartArt cargados actualmente en la aplicación.

Watches

Colección de objetos que representa los rangos de inspección cuando la hoja de cálculo se recalcula.

Windows

Colección de todas las ventanas de la aplicación de Excel o de un libro.

Workbooks

Colección de los libros (objeto Workbook) abiertos.

Worksheets

Colección de las hojas de cálculo (objeto Worksheet) de un libro.

3. Principios de uso de los objetos y las colecciones

3.1 Propiedades

Las propiedades sirven para describir un objeto. Ciertas propiedades son de solo lectura y no se pueden, por lo tanto, modificar con el código VBA.

Sintaxis

```
{<objeto> | <variable objeto>}.<propiedad>
```

Ejemplo

```
' Modificación del puntero del ratón
Application.Cursor = xlWait

' Muestra la versión de la aplicación de Excel activa
' Esta propiedad es de solo lectura
MsgBox Application.Version

' Puntero del ratón por defecto
Application.Cursor = xlDefault
```

3.2 Propiedades que representan objetos

Los objetos globales y los objetos definidos en el código a través de clases manejadas por VBA tienen ciertas propiedades cuyo valor se actualiza automáticamente por el sistema.

Estas **propiedades específicas** permiten acceder directamente a ciertos objetos: ventana activa, libro activo, celdas de la hoja activa, etc. La siguiente tabla muestra las propiedades específicas más usadas.

Propiedad	Objeto Parent	Objeto devuelto
ActiveCell	Application Window	Objeto Range que representa la primera celda activa de la ventana activa o especificada.

Propiedad	Objeto Parent	Objeto devuelto
ActiveChart	Application Window Workbook	Objeto Chart que representa el gráfico activo.
ActiveControl	Frame Page UserForm	Objeto Control que representa el control (ActiveX) activo.
ActivePane	Window	Objeto Pane que representa el panel activo de la ventana activa.
ActiveSheet	Application Window Workbook	Objeto Worksheet que representa la hoja activa del libro activo o del libro especificado.
ActiveWindow	Application	Objeto Window que representa la ventana activa.
ActiveWorkBook	Application	Objeto Workbook que representa el libro de la ventana activa.
Parent	Objetos múltiples	Devuelve el objeto contenedor.
Selection	Application Windows	Objeto Range que representa la celda o las celdas seleccionadas.
ThisCell	Application	Devuelve la celda por la que la función definida por el usuario es llamada como objeto Range.
ThisWorkbook	Application	Objeto Workbook que representa el libro sobre el que se ejecuta el código de la macro actual.

■ Observación

Las propiedades específicas que devuelven un objeto Range (Cells, Offset, Columns, Rows, etc.) se explican en detalle en el capítulo Objetos de Excel.

3.3 Métodos

Los métodos permiten realizar acciones sobre los objetos.

Son similares a los procedimientos:

- Pueden usar o no argumentos.
- Ciertos métodos pueden devolver un valor, como los procedimientos **Function**, y otros no, como los procedimientos **Sub**.

Sintaxis de método que no devuelve un valor

```
{<objeto> | <variable objeto>}.<método> [<Lista de argumentos>]
```

Ejemplo

```
' Activa la celda A1 de la segunda hoja de cálculo del libro activo  
Application.Goto ActiveWorkbook.Worksheets(2).Range("A1")  
' Selecciona un rango de celdas  
Range("A1:C12").Select  
' Borra las celdas seleccionadas  
Selection.Clear  
' Guarda el libro activo con un nuevo nombre  
ActiveWorkbook.SaveAs "C:\presupuesto\presupuesto2.xlsm"
```

Observación

Como en el caso de los procedimientos, los distintos argumentos de un método se separan por comas. Si un argumento opcional no se define explícitamente, el método usará un valor por defecto.

Sintaxis de método que devuelve un valor

```
<variable> = {<objeto> | <variable objeto>}.<método>  
[<Lista de argumentos>]
```

Ejemplo

```
Dim strFileName as String
    Muestra el cuadro de diálogo Abrir
strFileName = Application.GetOpenFilename _
    (FileFilter:="Libros Excel (*.xlsm), *.xlsm", _
    Title:="Seleccione el archivo que desea abrir")
    Si selecciona un archivo Excel, lo abre
If InStr(strFileName, "xls") <> 0 Then
    Workbooks.Open strFileName
Else
    MsgBox "Debe seleccionar un archivo Excel"
End If
```

3.4 Eventos

Un evento es una **acción específica** que se realiza sobre un cierto objeto. Microsoft Excel responde a varios tipos de eventos: apertura o cierre de un libro, selección de celdas, agregar una hoja de cálculo, etc. Los eventos resultan generalmente de una acción del usuario.

El uso de un procedimiento asociado a eventos le permite definir su propio código como respuesta a un evento que se produce en un libro, una hoja o un formulario.

Ejemplo

Cuando se agrega una nueva hoja de cálculo al libro, aparece un mensaje para el usuario.

```
Private Sub Workbook_NewSheet(ByVal Sh As Object)

MsgBox "Se ha añadido la hoja " & Sh.Name & Chr(13) & _
    "al libro " & _
    ActiveWorkbook.Name & Chr(13) & _
    "Ahora el número de hojas del libro es de " & _
    ActiveWorkbook.Worksheets.Count

End Sub
```