

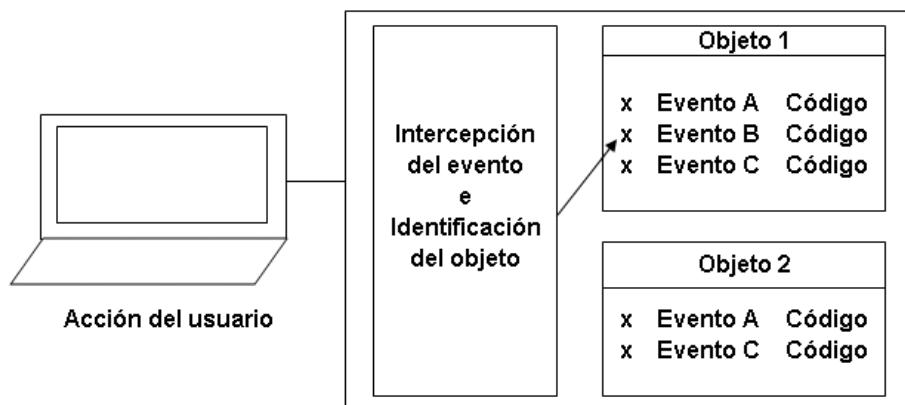
## Capítulo 9

# Administración de eventos

### 1. Presentación

Un evento es una **acción** del usuario o del sistema reconocido por un objeto de Microsoft Excel. El evento desencadena un procedimiento, asociado al evento del objeto activo.

Estos procedimientos le permiten asociar un código personalizado en respuesta a un evento que se produce en un objeto de Excel (libro, hoja, formulario, gráfico, etc.).



## 2. Escritura de eventos

### 2.1 Eventos de libro, de hoja o de formulario

Usted puede acceder a los procedimientos de eventos asociados a un objeto de la siguiente manera:

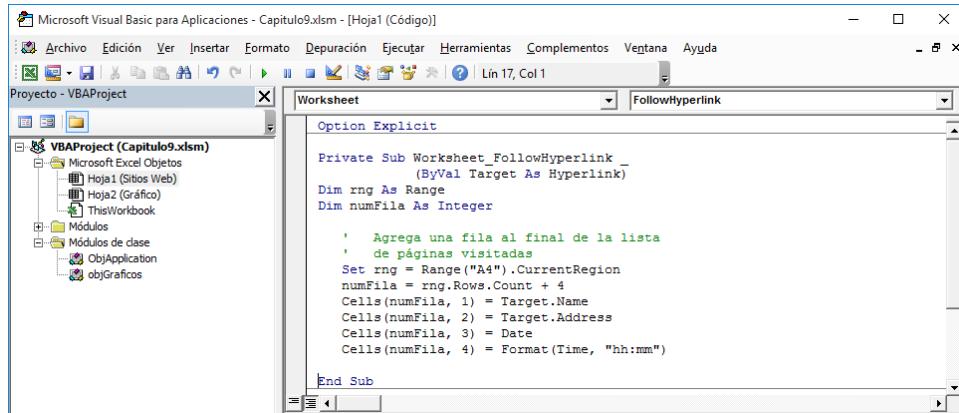
- En la ventana **Explorador de proyectos**, haga doble clic en el objeto deseado (libro, hoja o formulario) para hacer que aparezca la ventana de código correspondiente.
- Abra la lista desplegable a la izquierda de la ventana de código y seleccione **Workbook**, **Worksheet** o **UserForm**, según el objeto seleccionado.
- También puede seleccionar un evento vinculado al objeto seleccionado en la lista desplegable de la derecha, para asociarle un código personalizado.

#### Observación

La ejecución de los procedimientos de eventos se puede desactivar en cualquier momento asignando el valor **False** a la propiedad **EnableEvents** del objeto **Application**.

### Ejemplo

Este ejemplo muestra cómo obtener un listado histórico de todos los hipervínculos visitados en la hoja de cálculo activa.



The screenshot shows the Microsoft Visual Basic for Applications (VBA) editor. The title bar reads "Microsoft Visual Basic para Aplicaciones - Capítulo9.xlsx - [Hoja1 (Código)]". The menu bar includes Archivo, Edición, Ver, Insertar, Formato, Depuración, Ejecutar, Herramientas, Complementos, Ventana, and Ayuda. The toolbar has icons for New, Open, Save, Print, and Run. The Project Explorer on the left shows "VBAProject (Capítulo9.xlsx)" with nodes like "Microsoft Excel Objetos", "Hoja1 (Sitios Web)", "Hoja2 (Gráfico)", "ThisWorkbook", "Módulos", "Módulo de clase", "ObjApplication", and "objGraficos". The code editor on the right contains the following VBA code:

```
Option Explicit

Private Sub Worksheet_FollowHyperlink _
    (ByVal Target As Hyperlink)
Dim rng As Range
Dim numFila As Integer

    ' Agrega una fila al final de la lista
    ' de páginas visitadas
    Set rng = Range("A4").CurrentRegion
    numFila = rng.Rows.Count + 4
    Cells(numFila, 1) = Target.Name
    Cells(numFila, 2) = Target.Address
    Cells(numFila, 3) = Date
    Cells(numFila, 4) = Format(Time, "hh:mm")

End Sub
```

Resultado en Excel:

A	B	C	D	E	F	G	H
1							
2	LISTA DE PÁGINAS VISITADAS				PÁGINAS DISPONIBLES		
Nombre	Dirección	Fecha	Hora				
Microsoft	http://www.microsoft.com/	18/01/2019	16:45	Microsoft			
Microsoft Ibérica	http://www.microsoft.com/es/	04/02/2019	18:43	Microsoft Ibérica			
RENFE	http://www.renfe.es/	08/02/2019	17:42	Ediciones ENI			
Iberia	http://www.iberia.com/	12/02/2019	12:11	Bolsa de Madrid			
Bolsa de Madrid	http://www.bolsamadrid.es/	13/03/2019	19:18	El tiempo			
El tiempo	http://www.eltiempo.es/	15/03/2019	9:14	RENFE			
Microsoft	http://www.microsoft.com/	27/03/2019	10:02	Iberia			
RENFE	http://www.renfe.es/	06/04/2019	0:35				
Ediciones ENI	http://www.ediciones-eni.com/	06/04/2019	0:36				
14							
15							
16							

## 2.2 Eventos del objeto Application

Se necesitan tres etapas para la escritura y ejecución de los eventos del objeto **Application**.

### Etapa 1

■ Inserte un módulo de clase:

#### **Insertar - Módulo de clase**



o abra la lista y haga clic en **Módulo de clase**.

■ Una vez insertado el módulo, asígnele un nombre.

### Ejemplo

Dale el nombre **ObjApplication** al módulo de clase.

### Etapa 2

■ En el módulo de clase, cree un objeto **Application** con el siguiente código:

```
Public WithEvents NomObjeto As Application
```

### Ejemplo

Creación del objeto **oMiAplicación** como aplicación.

```
Public WithEvents oMiAplicacion As Application
```

El objeto así creado queda disponible en la lista de la izquierda del módulo.

■ Seleccione el objeto creado en la lista de la izquierda del módulo y luego seleccione el evento esperado en la lista de la derecha. Escriba el código de los procedimientos que desea generar.

Ejemplo

Creación de dos procedimientos de eventos: el primero realiza la inserción de una nueva hoja; el segundo, la creación de un nuevo libro.

```
Public WithEvents oMiAplicacion As Excel.Application

Private Sub oMiAplicacion_WorkbookNewSheet _
    (ByVal Wb As Workbook, ByVal Sh As Object)
    Dim oNomHoja As String
    ' Cada vez que se agrega una hoja, se pide al usuario
    ' que introduzca un nombre que a continuación se asignará a la hoja
    ' insertada tras las hojas existentes
    oNomHoja = InputBox("Introduzca el nombre de la hoja")
    ActiveSheet.Name = oNomHoja
    ActiveSheet.Move After:=Sheets(Sheets.Count)
End Sub

Private Sub oMiAplicacion_NewWorkbook(ByVal Wb As Workbook)
    Dim iNbHojas As Integer
    Dim iNbActual As Integer
    Dim iDiferencia As Integer
    ' Por cada nuevo libro,
    ' solicitamos al usuario la cantidad de hojas
    ' Según el caso, se agregan o eliminan las hojas necesarias
    Do
        iNbHojas = Application.InputBox(
            ("¿Cantidad de hojas?", Type:=1))
    Loop While iNbHojas = False
    iNbActual = Sheets.Count
    iDiferencia = iNbActual - iNbHojas
    ' Eliminar las hojas de más
    ' Eliminar los mensajes de alerta con el fin
    ' de no obtener mensajes en la eliminación de hojas
    Do While iDiferencia > 0
        Application.DisplayAlerts = False
        Sheets.Item(iDiferencia).Select
        ActiveWindow.SelectedSheets.Delete
        iDiferencia = iDiferencia - 1
    Loop

    ' Agregar hojas si es necesario
    ' Se desactivan los eventos para
    ' no indicar los nombres de las nuevas hojas
    Do While iDiferencia < 0
```

# 314 VBA Excel (2019 y Office 365)

## Programación en Excel: macros y lenguaje VBA

```
Application.EnableEvents = False
Sheets.Add After:=Sheets(Sheets.Count)
iDiferencia = iDiferencia + 1
Loop
' Reactivar eventos y alertas
Application.EnableEvents = True
Application.DisplayAlerts = True
End Sub
```

### Etapa 3

■ Active un módulo cualquiera y conecte el objeto declarado en el módulo de clase con el objeto **Application** para las siguientes instrucciones:

```
Dim oNomVariable As New NomModuloDeClase

Sub NomProced ()
Set oNomVariable.NomObjeto = Application
End Sub
```

### Ejemplo

Agregue el siguiente código en el módulo Declaraciones.

```
Option Explicit
Dim oApp As New ObjApplication

Sub InicializaMiAplicacion()
    Set oApp.oMiAplicacion = Application
End Sub
```

Finalmente, llame al procedimiento InicializaMiAplicacion al abrir el libro (módulo de clase ThisWorkbook).

```
Private Sub Workbook_Open()
    InicializaMiAplicacion
End Sub
```

Cuando se abra el libro, se ejecutarán automáticamente los procedimientos de eventos creados durante la etapa 2 y se agregarán los libros o las hojas. Estos procedimientos se desactivarán al cerrar el libro.