

Material Imprimible

Curso de Excel Macros

Módulo 4: Herramientas para la programación II

Programación de una macro

La estructura condicional compleja (and – or – elseif)

La estructura IF-THEN-ELSE nos permite condicionar la ejecución de un grupo de comandos de nuestra macro al cumplimiento de una condición.

En esta estructura podemos evaluar una o más condiciones en forma simultánea o en forma sucesiva.

En forma simultánea hace referencia a las condiciones que deberán evaluarse al mismo tiempo en su conjunto.

| |
|---|
| (CONDICIÓN 1 + CONDICIÓN 2 + CONDICIÓN 3) → ¿VERDADERO O FALSO? |
|---|

En forma sucesiva, a aquellas que deberán evaluarse sólo si la anterior no se cumplió.

| |
|-------------------------------------|
| (CONDICIÓN 1) → ¿VERDADERO O FALSO? |
|-------------------------------------|

| |
|---|
| Si es FALSO → (CONDICIÓN 2) → ¿VERDADERO O FALSO? |
|---|

| |
|---|
| Si es FALSO → (CONDICIÓN 3) → ¿VERDADERO O FALSO? |
|---|

Si las condiciones se evalúan **en forma simultánea** se usará el operador AND o OR. Se utilizará AND cuando todas las condiciones deban ser verdaderas para que el condicional ejecute los comandos posteriores. Se utilizará OR cuando sólo sea necesaria que al menos una condición sea verdadera para que se ejecuten estos comandos.

Por ejemplo:

```
IF Range("E2").Value >= 6 AND Range("F2").Value >= 6 THEN  
    MsgBox "APROBADO"  
ELSE  
    MsgBox "DESAPROBADO"  
END IF
```

En este ejemplo, chequeamos la nota de dos exámenes, que están escritas en las celdas E2 y F2. Si la nota de ambos es igual o superior a 6 aparecerá un cuadro de mensaje que diga "APROBADO", caso contrario, aparecerá un cuadro de mensaje que diga "DESAPROBADO". Ambas condiciones deben cumplirse para que se ejecute el *msgbox* que diga "APROBADO".

```
IF Range("E2").Value >= 6 OR Range("F2").Value >= 6 THEN  
    MsgBox "APROBADO"  
ELSE  
    MsgBox "DESAPROBADO"  
END IF
```

En este ejemplo, chequeamos la nota de dos exámenes, que están escritas en las celdas E2 y F2. Si al menos una nota es igual o superior a 6 aparecerá un cuadro de mensaje que diga "APROBADO", caso contrario, aparecerá un cuadro de mensaje que diga "DESAPROBADO". Basta con que se cumpla una de las condiciones para que se ejecute el *msgbox* que diga "APROBADO".

Si las condiciones se evalúan **en forma sucesiva** se usará el comando ELSEIF. Con éste, se podrá evaluar una nueva condición la cantidad de veces que sea necesario. Se usará ELSE para el caso en que no se cumpla ninguna de las condiciones anteriormente evaluadas.

Por ejemplo:

```
IF Range("E2").Value >= 6 THEN  
    msgbox "APROBADO"  
ELSEIF Range("E2").Value >= 4 THEN  
    msgbox "DESAPROBADO"  
ELSE  
    msgbox "APLAZADO"  
END IF
```

En este ejemplo, chequeamos la nota de un examen, que está escrita en la celda E2. Si la nota es igual o superior a 6 aparecerá un cuadro de mensaje que diga "APROBADO", caso contrario, se evaluará si la nota es superior o igual a 4. En este último caso aparecerá un cuadro de mensaje que diga "DESAPROBADO", caso contrario, "APLAZADO".

Los bucles

Un bucle es una estructura en VBA que nos permite repetir un grupo de instrucciones una determinada o indeterminada cantidad de veces.

Existen distintos tipos de bucles, según la forma en que se los exprese.

Los bucles se llaman: FOR – FOR EACH – DO WHILE – DO UNTIL

En todos los casos será necesario crear una variable que permita guardar el número de repeticiones del bucle y utilizarlo para las referencias dentro de sus comandos.

El bucle se ejecutará tantas veces según cómo se lo indiquemos. Al usar una variable, debemos asignarle un valor inicial y podremos hacer que se incremente de 1 en 1 para aprovecharla en nuestro código, por ejemplo, al indicarle que haga una determinada acción cada vez en una fila o columna distinta.

La estructura de un bucle **FOR** es la siguiente:

```
FOR inicio_variable TO Limite_variable  
    Conjunto_instrucciones_a_repetir  
NEXT
```

Por ejemplo:

```
Dim Rep As Integer
```

```
FOR Rep = 1 TO 5
```

```
    Cells(Rep, 1).Value = "Hola mundo"
```

```
NEXT
```

En este ejemplo, se repetirán 5 veces los comandos del bucle. El bucle escribe la frase "Hola mundo" en las celdas A1, A2, A3, A4 y A5. Observemos que el número de la fila del objeto *Cells* está representado por la variable *Rep*. Esto quiere decir que cada vez que el bucle se ejecute, le indicará a Excel que lo haga en la fila siguiente (irá probando del 1 al 5 pasando por todos los números intermedios).

La estructura de un bucle **FOR EACH** es la siguiente:

```
FOR EACH objeto IN conjunto_de_objetos
```

```
    Conjunto_instrucciones_a_repetir
```

```
NEXT
```

Por ejemplo:

```
Dim Rep As Integer
```

```
Rep = 1
```

```
FOR EACH Row IN Range("A1:A5").Rows
```

```
    Cells(Rep, 1).Value = "Hola mundo"
```

```
    Rep = Rep + 1
```

```
Next
```

En este ejemplo, los comandos del bucle se repetirán tantas veces como filas haya en el conjunto de filas del rango A1:A5, es decir, 5 veces. El bucle escribe la frase "Hola mundo" en las celdas A1, A2, A3, A4 y A5, tal como vimos en el ejemplo del bucle FOR.

En este caso, tendremos que indicarle previamente con qué valor empieza la variable (Rep = 1) e indicarle antes de que finalice el bucle que se incremente en 1 (Rep = Rep + 1). En el bucle FOR esto último no era necesario.

La estructura de un bucle **DO WHILE** es la siguiente:

```
DO WHILE condición
    Conjunto_instrucciones_a_repetir
LOOP
```

Por ejemplo:

```
Dim Rep As Integer

Rep = 1

DO WHILE Not( IsEmpty( Cells(Rep, 1) ) )
    Cells(Rep, 2).Value = "Hola mundo"
    Rep = Rep + 1
LOOP
```

En este ejemplo, los comandos del bucle se repetirán tantas veces mientras se cumpla en cada caso la condición prevista, es decir, que no esté vacía una determinada celda. El bucle escribirá la frase "Hola mundo" en las celdas de la columna B, mientras que no esté vacía la celda correspondiente de la columna A. Como hicimos con el bucle FOR EACH, tendremos que indicarle previamente con qué valor empieza la variable (Rep = 1) e indicarle antes de que finalice el bucle que se incremente en 1 (Rep = Rep + 1).

La estructura de un bucle **DO UNTIL** es la siguiente:

```
DO UNTIL condición
    Conjunto_instrucciones_a_repetir
LOOP
```

Por ejemplo:

```
Dim Rep As Integer
```

```
Rep = 1
```

```
DO WHILE IsEmpty( Cells(Rep, 1) )
```

```
    Cells(Rep, 2).Value = "Hola mundo"
```

```
Rep = Rep + 1
```

```
LOOP
```

En este ejemplo, los comandos del bucle se repetirán tantas veces hasta que se cumpla la condición prevista, es decir, que esté vacía una determinada celda. El bucle escribirá la frase "Hola mundo" en las celdas de la columna B, hasta que encuentre una celda que esté vacía en la columna A. Como hicimos con el bucle FOR EACH y DO WHILE, tendremos que indicarle previamente con qué valor empieza la variable (Rep = 1) e indicarle antes de que finalice el bucle que se incremente en 1 (Rep = Rep + 1).

Resumen BUCLES:

| | FOR | FOR EACH | DO WHILE | DO UNTIL |
|--|------|----------|----------|----------|
| Inicia con... | FOR | FOR EACH | DO WHILE | DO UNTIL |
| Finaliza con... | NEXT | NEXT | LOOP | LOOP |
| ¿Indicamos valor inicial de la variable? | NO | SI | SI | SI |
| ¿Incrementamos el valor de la variable? | NO | SI | SI | SI |

Los eventos en VBA

Un evento es un hecho o un momento que desencadena un conjunto de comandos en VBA.

Hasta el momento, nuestras macros sólo se ejecutaban cuando el usuario se lo pedía, por ejemplo, a partir de un botón o un método abreviado.

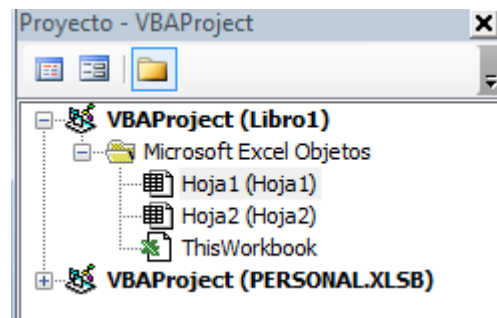
Con los eventos, podemos hacer que las macros se ejecuten **automáticamente** cuando ocurra un determinado hecho en nuestra planilla, por ejemplo, al abrir el libro de Excel, al cambiar una hoja, al cambiar una selección, etcétera.

Por un lado, tenemos los eventos que suceden dentro de cada hoja (cambiar una selección, activarla, etcétera).

Por otro lado, tenemos los eventos del libro en general (abrirlo, cambiar de hoja, etcétera).

La escritura de los comandos es la misma para el caso que queramos utilizar un evento como desencadenante de nuestra macro sólo que, en este caso, no la escribiremos en un módulo, sino que en otro espacio.

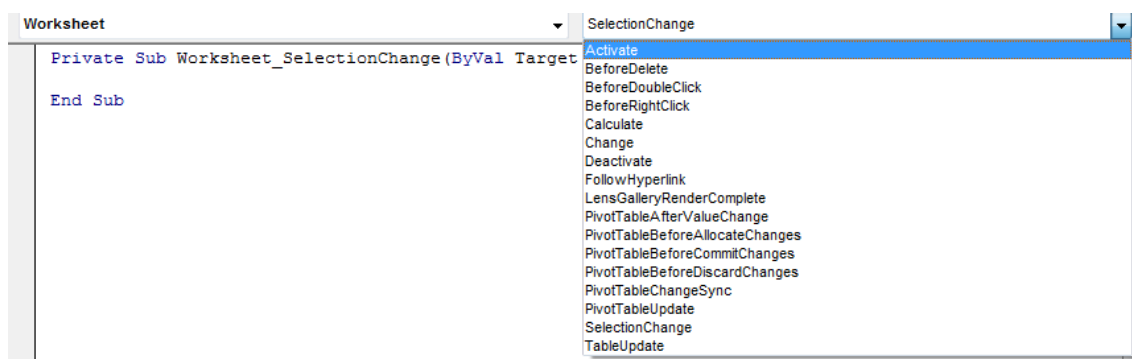
Iremos al explorador de proyectos en el Editor de Visual Basic. Allí, dentro de la carpeta "Microsoft Excel Objetos" haremos doble click sobre una hoja del libro (si quisiéramos trabajar con los eventos de esa hoja) o sobre ThisWorkbook (si quisiéramos trabajar con los eventos del libro entero).



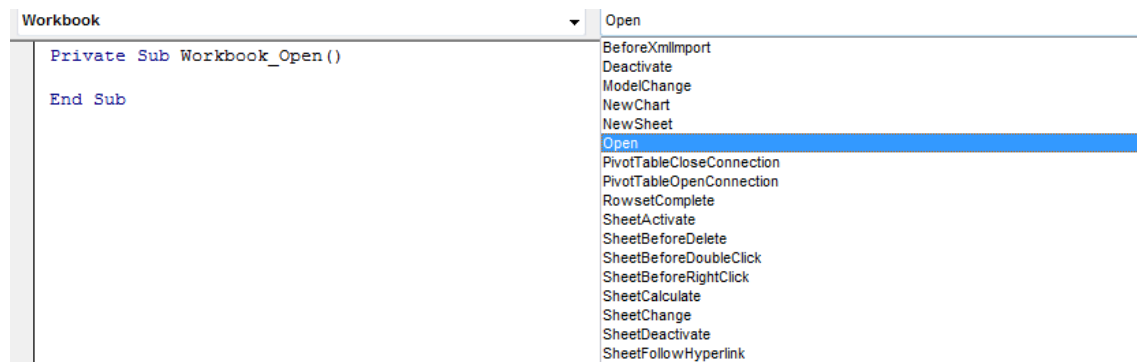
Dentro del espacio que se abra para escribir nuestras macros, seleccionaremos previamente el evento que necesitamos en la parte superior de la ventana.

Del lado izquierdo, elegir el objeto (Worksheet o ThisWorkbook) y del lado derecho, el evento.

Ejemplo de eventos sobre una HOJA



Ejemplo de eventos sobre el LIBRO



Una vez seleccionado el evento deseado, se escribirán automáticamente las líneas de apertura y de cierre correspondiente a nuestra macro.

Por ejemplo:

```
Private Sub Workbook_Open()
```

```
End Sub
```

Dentro de esas líneas tendremos que escribir todos los comandos que queremos que se ejecuten automáticamente cuando ocurra dicho evento; en el caso de la foto, cuando se abra el libro.

De esta manera, los eventos nos permitirán otorgarle mayor funcionalidad a nuestras macros en las planillas de Excel.