

## A. Objetivos del capítulo

Cuando quiera ejecutar un programa según ciertas condiciones, por ejemplo, para poner una fila en verde si el importe de una columna es mayor que una cantidad determinada, o si quiere saber en qué hoja copiar un valor en función de los elementos situados en otra parte de su libro de trabajo, necesitará utilizar estructuras de condición y, más generalmente, los condicionales VBA. El objetivo de este capítulo es mostrarle las principales sintaxis de estas estructuras, cómo utilizarlas y cuáles elegir en cada situación.

Al final del capítulo, encontrará una serie de ejercicios para practicar.

## B. Estructuras condicionales

Cuando codifica en VBA, usted puede indicarle al programa que realice una prueba y, dependiendo del resultado de esta, que ejecute o no una serie de instrucciones. Esto se denomina **estructuras condicionales**.

## C. Noción de prueba

Cuando hablamos de una **prueba**, nos referimos a una igualdad o desigualdad que devuelve un valor booleano (Verdadero/Falso – `True/False`) que permite determinar el camino que debe seguir el programa en función de ese valor. Existen varias estructuras condicionales; empezaremos por las más sencillas antes de abordar las más elaboradas.

## D. Condición Si Entonces - If Then

La estructura de condición más simple en VBA es la Si Entonces, cuya sintaxis en VBA corresponde a **If Then** (traducción literal del inglés).

En lenguaje humano, esto equivaldría a:

```
Si (algo es cierto) entonces
    Ejecutamos este código
Fin Si
```

La sintaxis general de dicha estructura es la siguiente:

```
If Condicion_Que_Se_Debe_Probar Then
    Instrucciones ' se ejecutarán si Condicion_Que_Se_Debe_Probar
    devuelve True (Verdadero)
End If
```

### *Ejemplo 1: caso de una estructura de condición If Then*

La estructura de esta condición consiste en una primera línea que comienza con la palabra clave **If**, seguida de la condición que se va a probar y la palabra clave **Then** para terminar esta primera línea. A esto le siguen las instrucciones que se ejecutarán si la condición colocada entre las palabras clave **If** y **Then** es verdadera (True). Por último, la estructura termina con las palabras clave **End If** (Fin Si).

Las instrucciones dentro de la estructura no se ejecutarán si la condición es falsa (False).

He aquí algunos ejemplos:

```
' El usuario ha proporcionado la edad, si esta es menor de 18 años,
' se mostrará un mensaje indicando que es menor de edad
If Edad < 18 Then
    MsgBox "Su usuario es menor de edad."
End If
```

### *Ejemplo 2: estructura de condición simple*

```
Dim strNombre As String, strApellido As String
...
'Desea validar que el nombre y el apellido del usuario se introduzcan
correctamente
If strNombre="" Or strApellido = "" Then
    MsgBox "Introduzca el nombre y el apellido del usuario correctamente"
End If
```

### *Ejemplo 3: estructura de condición con un Or*



Más raro, pero poco recomendable, es ver también instrucciones que siguen a la palabra clave **Then**, en cuyo caso la estructura no necesitará terminar por **End If**.

```
| If Condicion_Que_Se_Debe_Probar Then Instrucciones
```

*Ejemplo 4: uso condensado, pero desaconsejado, de la estructura If Then*

## E. Condición Si no - Else

En caso de que la condición del **If** no se cumpla, es decir, devuelva **False** (falso), es posible ejecutar otras líneas de código.

En lenguaje humano, esto sería equivalente a:

```
| Si (algo es cierto) entonces
  Ejecutar el código A
| Si no
  Ejecutar el código B
| Fin Si
```

La palabra clave que hay que utilizar aquí es **Else**. Su uso resulta muy sencillo con la siguiente sintaxis:

```
| If Condicion_Que_Se_Debe_Probar Then
  InstruccionesA 'se ejecutan si Condicion_Que_Se_Debe_Probar
devuelve True (Verdadero)
| Else
  InstruccionesB 'se ejecutan si Condicion_Que_Se_Debe_Probar
devuelve False (Falso)
| End If
```

*Ejemplo 5: caso de una estructura condicional If Then Else*

Por lo tanto, esta palabra clave permite realizar una prueba y, en caso de que esta no cumpla las expectativas, usted puede ejecutar otro código.



*Cuando se utiliza una estructura If Else, solo podrá ejecutarse una de las dos instrucciones InstruccionesA e InstruccionesB, pero no las dos.*

## F. Condición Si no Si - ElseIf

A la hora de programar, puede ocurrir que, cuando una primera condición no se cumpla, se quiera probar otra para poder ejecutar otra serie de instrucciones. Esta estructura también existe en VBA.

En lenguaje humano, esto sería equivalente a:

```
| Si (una condición se cumple) Entonces
  Se ejecuta el código A
| Si no Si (otra condición se cumple) Entonces
  Se ejecuta el código B
| Si no
```

```
Se ejecuta el código C
Fin Si
```

La palabra clave que permite esta sintaxis es **ElseIf** (en una palabra). La sintaxis VBA equivalente es la siguiente:

```
If Condicion_Que_Se_Debe_Probar_1 Then
    InstruccionesA 'se ejecutarán si Condicion_Que_Se_Debe_Probar_1
                    'devuelve True (Verdadero)
ElseIf Condicion_A_Probar_2 Then
    InstruccionesB 'se ejecutarán si Condicion_Que_Se_Debe_Probar_1
                    'devuelve False (Falso) Y que
                    'Condicion_Que_Se_Debe_Probar_2 devuelve True
(Verdadero)
Else
    InstruccionesC 'se ejecutarán si tanto Condicion_Que_Se_Debe_
Probar_1 como
                    'Condicion_Que_Se_Debe_Probar_2 devuelven False (Falso)
End If
```

*Ejemplo 6: sintaxis general de un ElseIf*

Esta estructura permite realizar varias pruebas una tras otra. Este es un ejemplo para determinar la categoría de edad de un usuario. Así que, en lugar de tener una cascada de varios niveles para sus pruebas, como en el siguiente ejemplo:

```
Dim Edad As Integer
...
If Edad<18 Then
    MsgBox "Su usuario es menor de edad"
Else
    If Edad<50 Then
        MsgBox "Su usuario está en la categoría de edad 18-49 años"
    Else
        MsgBox "Su usuario está en la categoría de más de 50 años"
    End If
End If
```

*Ejemplo 7: usar If Else en cascada*

Puede simplificar el código con el uso de **ElseIf** como se muestra a continuación.

```
Dim Edad As Integer
...
If Edad<18 Then
    MsgBox "Su usuario es menor de edad"
ElseIf Edad<50 Then
    MsgBox "Su usuario está en la categoría de edad 18-49 años"
Else
    MsgBox "Su usuario está en la categoría de más de 50 años"
End If
```

*Ejemplo 8: usar If ElseIf Else*

Al igual que la estructura **If**, la línea que comienza con la palabra clave **ElseIf** termina con la palabra clave **Then**. Puede haber tantos **ElseIf** como quiera en su estructura condicional, pero solo puede haber un **Else**, y estará en la última parte de la estructura, es decir, después de todos los **ElseIf**. Así, el siguiente código no será válido:

```
Dim Edad As Integer
...
If Edad<18 Then
    MsgBox "Su usuario es menor de edad"
Else
    MsgBox "Su usuario está en la categoría de más de 50 años"
ElseIf Edad<50 Then
    MsgBox "Su usuario está en la categoría de edad 18-49 años"
End If
```

*Ejemplo 9: código erróneo a causa del orden de los Else y ElseIf*

El uso de esta estructura de condición es práctico, pero aún se puede utilizar otra, que también puede ser muy eficaz, sobre todo si tiene varios valores de prueba para los que las instrucciones que se han de ejecutar serían las mismas.

## G. Condición Selección de casos - Select Case

Cuando quiera colorear una celda en verde, por ejemplo, según tenga las letras A, B, C o D; en azul si tiene un valor entre 1 y 5; en amarillo si tiene un valor superior a 10, o en rojo para todos los demás casos, el uso de la sintaxis con **If ElseIf** será especialmente engorroso, como puede ver en este ejemplo:

```
If Range("A1").Value = "A" Then
    Range(" A1 ").Interior.Color = vbGreen
ElseIf Range("A1").Value = "B" Then
    Range("A1").Interior.Color = vbGreen
ElseIf Range("A1").Value = "C" Then
    Range("A1").Interior.Color = vbGreen
ElseIf Range("A1").Value = "D" Then
    Range("A1").Interior.Color = vbGreen
ElseIf Range("A1").Value = 1 Then
    Range("A1").Interior.Color = vbBlue
ElseIf Range("A1").Value = 2 Then
    Range("A1").Interior.Color = vbBlue
ElseIf Range("A1").Value = 3 Then
    Range("A1").Interior.Color = vbBlue
ElseIf Range("A1").Value = 4 Then
    Range("A1").Interior.Color = vbBlue
ElseIf Range("A1").Value = 5 Then
    Range("A1").Interior.Color = vbBlue
ElseIf Range("A1").Value > 10 Then
```

```

    Range("A1").Interior.Color = vbYellow
Else
    Range("A1").Interior.Color = vbRed
End If

```

*Ejemplo 10: usar intensivamente la estructura If ElseIf*

Del mismo modo, no se puede tener un código fácil encadenando **Or** entre cada caso, como en este ejemplo:

```

If Range("A1").Value = "A" Or Range("A1").Value = "B" Or _
    Range("A1").Value = "C" Or Range("A1").Value = "D" Then
    Range("A1").Interior.Color = vbGreen
ElseIf Range("A1").Value = 1 Or Range("A1").Value = 2 Or
    Range("A1").Value = 3 Or Range("A1").Value = 4 Or
    Range("A1").Value = 5 Then
    Range("A1").Interior.Color = vbBlue
ElseIf Range("A1").Value > 10 Then
    Range("A1").Interior.Color = vbYellow
Else
    Range("A1").Interior.Color = vbRed
End If

```

*Ejemplo 11: usar el operador Or*

Aunque el código del ejemplo 11 es más corto y menos indigesto de leer que el del ejemplo 10, imagine que, en lugar de las 4 letras y 5 números que tiene aquí, hay decenas que probar. El código será muy lento de escribir y luego de mantener.

Afortunadamente, para este caso, existe una estructura que permite agrupar las instrucciones que se van a ejecutar según el valor de su celda. Esta estructura se llama **Select Case** y aquí está la sintaxis general:

```

Select Case Variable
    Case Valor_1, Valor_2
        InstruccionesA
    Case Valor_3, Valor_4
        InstruccionesB
    Case Valor_5 To Valor_6
        InstruccionesC
    Case Is > Valor_7
        InstruccionesD
    Case Else 'Si no aparece ninguno de los valores anteriores
        InstruccionesPredefinidas
End Select

```

*Ejemplo 12: sintaxis general de la estructura Select Case*

Esta estructura comienza con las palabras clave **Select Case**, seguidas de la variable sobre la que se quieren probar los posibles valores. Para cada conjunto de valores, tendrá la palabra clave **Case** seguida del valor o valores para los que se ejecutarán las instrucciones. Por último, el bloque termina con un **End Select**, similar al **End If** visto en las secciones anteriores. Al igual que en la estructura If Then con la palabra clave **Else**, puede tener sentencias que se ejecutarán si no se encuentra ninguno de los casos, con las palabras clave **Case Else**.

Note que se puede proporcionar un rango de valores con esta estructura condicional.

Utilizando el caso propuesto en el ejemplo 10, esta es la sintaxis que se necesitaría para lograr el mismo resultado:

```
Select Case Range("A1").Value
    Case "A", "B", "C", "D"
        Range("A1").Interior.Color = vbGreen
    Case 1 To 5
        Range("A1").Interior.Color = vbBlue
    Case Is > 10
        Range("A1").Interior.Color = vbYellow
    Case Else
        Range("A1").Interior.Color = vbRed
End Select
```

*Ejemplo 13: usar pertinentemente Select Case*

## H. Condición condensada: IIf

Para terminar este capítulo, hay una última estructura de condición que puede utilizar. Se trata de la función **IIf( )**, cuya sintaxis general es la siguiente:

```
Dim Variable
Variable = IIf(Condición, ValorSiCondicionVerdadera, ValorSiCondicion-
Falsa)
```

*Ejemplo 14: sintaxis general de la función IIf*

Esta función toma tres parámetros en total. El primero corresponde a la condición que se va a probar. El segundo parámetro será el valor de retorno si la condición que se prueba devuelve True; el tercero es el valor devuelto si la condición es falsa.

Puede utilizar esta función cuando quiera asignar un valor a una variable según una condición.

He aquí un ejemplo de la variable **Edad** de nuestro usuario, y según esta edad, determinamos si el usuario es mayor o menor de edad.

```
Dim MayorOMenor As String  
...  
MayorOMenor = IIf(Edad<18, "Menor", "Mayor") & " de edad"
```

*Ejemplo 15: ejemplo de uso de la función IIf*

Esta función puede reemplazar una estructura **If Then Else** vista anteriormente en este capítulo y lograr el mismo objetivo.

```
Dim Edad As Integer  
Dim MayorOMenor As String  
...  
If Edad<18 Then  
    MayorOMenor = "Menor"  
Else  
    MayorOMenor = "Mayor"  
End If  
MayorOMenor = MayorOMenor & " de edad"
```

*Ejemplo 16: sintaxis equivalente a IIf usando un If Else*

Esta función a veces puede simplificar su código, pero se recomienda hacer estructuras en varias líneas para una mayor legibilidad y, no lo olvide:

1/ Funciona

Además, es imposible colocar varias sentencias en una función **IIf** porque esta función solo devuelve un valor.

## I. Ejercicios

### 1. Si entonces

☞ Escriba un procedimiento **MuestraSiNegativo**, de ámbito público, que requiera un parámetro numérico entero llamado **ValorDelDesencadenante**.

Este procedimiento deberá mostrar OK si el valor de la variable **ValorDelDesencadenante** es estrictamente inferior a 0.

☞ Cree la macro **Capítulo08\_Ejercicio1**.

Esta macro llamará primero a **MuestraSiNegativo** pasándole un valor de 35, y luego a **MuestraSiNegativo** pasándole un valor de -5.

☞ Ejecute la macro **SiEntonces**.

El mensaje OK deberá aparecer una sola vez.

## 2. SI si no

☞ Escriba una función **DevuelveDobleOMitad** de ámbito privado.

Tomará como parámetro de entrada un valor entero menor que 255, llamado **ValorEntrada**, y devolverá un valor posiblemente decimal.

Esta función devolverá el doble del valor **ValorEntrada** si este es superior a 128, o la mitad del **ValorEntrada** en el caso contrario.

☞ Cree la macro **Sisino**.

Esta macro mostrará sucesivamente el valor devuelto por **DevuelveDobleOMitad**, pasándole el valor 15, luego de nuevo por **DevuelveDobleOMitad**, pasándole esta vez el valor 129.

☞ Ejecute la macro **Sisino** (resultado: 7.5 y 258).

Aparecerán los valores 7.5 y luego 258.

## 3. SI si no si

☞ Escriba la función **PrecioEnvio**, de ámbito público, que devuelve un valor numérico decimal.

Esta función toma como parámetro una cadena de caracteres, llamada **TallaEnvio**.

Esta función devolverá M si el valor pasado es 0.52, L para 0.63, XL para 1.81, XXL para 2.62 o XXXL para 4.10.

☞ Cree la macro **Sisinosi**.

Esta macro llamará a la función **PrecioEnvio** y mostrará el valor devuelto pasando los valores M, L, XL y XXXL sucesivamente.

☞ Ejecute la macro **Sisinosi** (0.52, 0.63, 1.81 y 4.10).

Los valores 0.52, 0.63, 1.81 y 4.10 aparecerán en la pantalla.

## 4. Segundo Valor

☞ Utilizará la estructura **Select Case** para este ejercicio.

☞ Escriba la función **TallaAdulto**, de ámbito público, que devuelve una cadena.

Esta función tomará como parámetro un valor entero, entre 0 y 255, llamado **Peso**.

Esta función devolverá el valor XS para valores de **Peso** estrictamente inferiores a 55, S para valores entre 55 y 64, M para valores de 65 a 74, L para valores de 75 a 84 y XL para otros valores.

☞ Cree la macro **SegundoValor**.