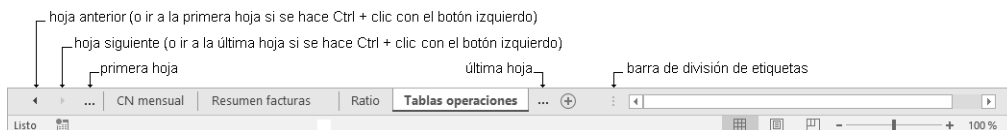




Activar una hoja

Recuerde que los libros se componen de hojas de cálculo y que estas se representan mediante etiquetas.


- Gracias a los botones de desplazamiento de las etiquetas, podrá visualizar el nombre de la hoja a la que desea acceder:



- Para mostrar todas las etiquetas, haga clic con el botón derecho en la herramienta  o en .

A continuación, haga clic en la etiqueta de la hoja a la que desea acceder.

- Con el teclado, use las teclas **Ctrl** **Av Pág** para pasar a la hoja siguiente o las teclas **Ctrl** **Re Pág** para volver a la hoja anterior.

-  El espacio reservado a las etiquetas de la hoja puede modificarse deslizando la barra de división de la etiqueta, situada a la izquierda de la barra de desplazamiento horizontal.


Para modificar el número de hojas de un libro nuevo, abra el cuadro de diálogo **Opciones de Excel** (pestaña **Archivo** - **Opciones**) y, en la categoría **General**, especifique el número de hojas deseado en la opción **Incluir este número de hojas de la zona Al crear nuevos libros**.

Cambiar el nombre de una hoja

- Haga clic en la etiqueta de la hoja cuyo nombre desea cambiar. Teclee el nuevo nombre.

El nombre no debe sobrepasar los 31 caracteres (espacios incluidos); algunos caracteres no pueden usarse: dos puntos (:), barra oblicua (/ o \), signo de interrogación (?), asterisco (), etc.*

- Confirme pulsando la tecla .

-  Otra posibilidad es usar el menú contextual de la etiqueta de hoja, opción **Cambiar nombre**.

Seleccionar hojas

- ☐ Para seleccionar una sola hoja, haga clic sobre su etiqueta; la hoja se habrá activado.
- ☐ Para seleccionar varias hojas contiguas, haga clic en la etiqueta de la primera hoja que quiera seleccionar, mantenga pulsada la tecla **Mayús** y haga clic en la etiqueta de la última hoja.
- ☐ Para seleccionar varias hojas no contiguas, haga clic en la etiqueta de la primera hoja que quiera seleccionar, mantenga pulsada la tecla **Ctrl** y haga clic en las etiquetas de las demás hojas.
- ☐ Para seleccionar todas las hojas, haga clic con el botón derecho del ratón en una de las etiquetas y luego haga clic en **Seleccionar todas las hojas**.

El nombre de las etiquetas seleccionadas aparecerá en negrita.

*Cuando hay varias hojas seleccionadas, aparece el término **[Grupo]** en la barra de título, tras el nombre del libro. Decimos que estamos en un grupo porque las modificaciones hechas en la hoja activa se trasladan a todas las hojas del grupo.*



- ☐ Para desactivar el grupo y seleccionar o activar una sola hoja, haga clic en la etiqueta de alguna hoja que no forme parte del grupo o muestre el menú contextual de una etiqueta (clic derecho) y active la opción **Desagrupar hojas**.

Si están seleccionadas todas las hojas del libro, el hecho de hacer clic sobre una de las etiquetas desactiva el grupo.

Modificar el color de las etiquetas

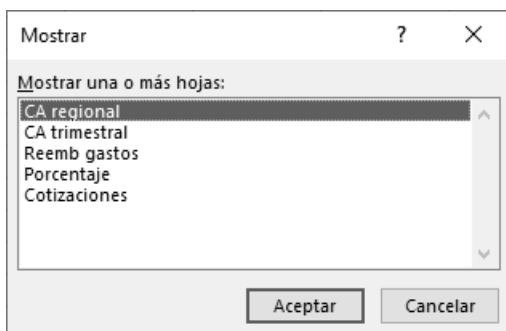
- ☐ Si desea modificar el color de varias etiquetas, seleccione las hojas correspondientes.
- ☐ Muestre el menú contextual de la etiqueta (clic derecho), señale la opción **Color de etiqueta** y haga clic en el color deseado. La opción **Más colores** permite abrir un cuadro de diálogo desde el que crear un color personalizado.
- ☞ Cuando la hoja está activa, su nombre aparece subrayado con una línea de color, pero cuando está inactiva, toda la etiqueta aparece coloreada.

-
- ☞ Para quitar el color a una etiqueta, muestre su menú contextual, señale la opción **Color de etiqueta** y active la opción **Sin color**.
-

Mostrar y ocultar una hoja

- ❑ Para ocultar una o varias hojas, seleccione las hojas que desea ocultar, muestre el menú contextual de una de las etiquetas seleccionadas (clic derecho) y escoja la opción **Ocultar**.
- ❑ Para mostrar una hoja oculta, muestre el menú contextual de una etiqueta cualquiera y active la opción **Mostrar**.

El nombre de las hojas ocultas aparece en el cuadro de diálogo que se muestra:



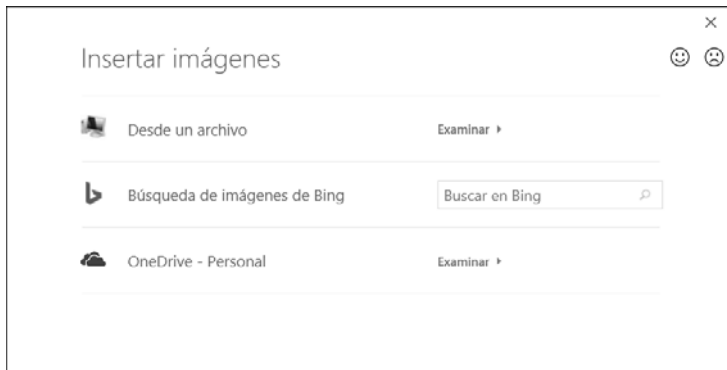
- ❑ Haga doble clic en el nombre de la hoja o haga clic una sola vez en el nombre y luego en el botón **Aceptar**.
- ❑ Repita la misma operación con todas las hojas que desee ver.

Mostrar una imagen como fondo de la hoja

Microsoft Excel ofrece la posibilidad de mostrar una imagen como fondo de la hoja. Sin embargo, este fondo no podrá imprimirse.

- ❑ Active la hoja y la pestaña **Diseño de página**.
- ❑ Haga clic en el botón **Fondo** del grupo **Configurar página**.

Las hojas de cálculo



- ☐ En función de la ubicación de la imagen, escoja una de las opciones siguientes:
Desde un archivo si la imagen está almacenada en su ordenador o en una red local.
Búsqueda de imágenes de Bing para realizar una búsqueda en la web.
OneDrive (u otro espacio en línea) si la imagen está almacenada en su espacio OneDrive o en otra ubicación en línea.
- ☐ Haga doble clic en el nombre de la imagen.



Observe que el botón *Fondo* del grupo *Configurar página* se ha convertido en el botón *Eliminar fondo*.



Para eliminar la imagen que hace de fondo de la hoja, active, si es preciso, la pestaña *Diseño de página* y haga clic en el botón *Eliminar fondo* del grupo *Configurar página*.

Mover y copiar una o varias hojas

En el libro activo

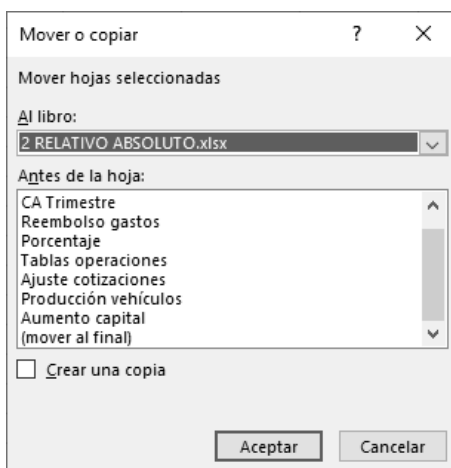
- ❑ Seleccione la hoja u hojas que quiere transferir.
- ❑ Para mover las hojas, señale una de las etiquetas de la selección y haga clic y arrastre hasta la nueva posición.

La nueva posición se representa mediante un triángulo negro.

- ❑ Para copiar las hojas, lleve a cabo la misma operación que para moverlas, manteniendo pulsada la tecla **Ctrl**.

De un libro a otro


- ❑ Abra el libro que contiene las hojas que se van a mover o copiar y el libro de destino.
- ❑ Seleccione la hoja u hojas que se van a transferir, haga clic con el botón derecho del ratón en una de las etiquetas seleccionadas y active la opción *Mover o copiar*.
- ❑ Abra la lista *Al libro* y haga clic en el nombre del libro de destino o bien escoja la opción *(nuevo libro)*.



Las hojas de cálculo

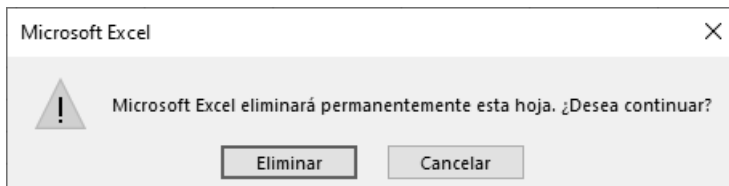
- ☐ En la lista **Antes de la hoja**, seleccione la hoja del libro de destino delante de la cual desea transferir la hoja en cuestión.
- ☐ Si desea hacer una copia de la hoja u hojas seleccionadas, marque la opción **Crear una copia**. Si desea desplazarlas, deje desmarcada esta opción.
- ☐ Haga clic en **Aceptar**.
- ☞ El libro de destino se convierte en el libro activo.

Insertar y agregar hojas

- ☐ Para añadir una hoja después de la hoja activa, haga clic en la herramienta **Hoja nueva**  de la barra de hojas. Para añadir una hoja nueva antes de la hoja activa, utilice el método abreviado **Mayús | F11**.
- ☐ Para insertar una sola hoja entre dos hojas que ya existen, seleccione la hoja delante de la cual desea que esta se inserte. Para insertar varias hojas al mismo tiempo, seleccione tantas etiquetas consecutivas como hojas desee insertar.
En este último caso, las hojas se insertarán entre la penúltima hoja seleccionada y la última.
- ☐ Haga clic con el botón derecho del ratón en una de las etiquetas seleccionadas y active la opción **Insertar**.
- ☐ En el cuadro de diálogo **Insertar** que aparece a continuación, compruebe que está activa la opción **Hoja de cálculo** y haga clic en **Aceptar**.

Eliminar hojas

- ☐ Seleccione la hoja u hojas que desea eliminar.
- ☐ Haga clic con el botón derecho del ratón en una de las etiquetas seleccionadas y active la opción **Eliminar**.
Si alguna de las hojas contiene datos, aparece el siguiente mensaje:



- ☐ En ese caso, haga clic en el botón **Eliminar**.

Insertar filas y columnas

- ❑ Para insertar una sola fila o columna, seleccione la fila o la columna (haciendo clic en el número de la fila o en la letra de la columna) después de la cual desea insertar la nueva.

Para insertar varias filas o columnas, seleccione tantas filas o columnas como desee insertar.

No se pueden insertar a la vez filas y columnas.

- ❑ Active la pestaña **Inicio** y haga clic en el botón **Insertar** del grupo **Celdas**, use el método abreviado **Ctrl** + o bien active la opción **Insertar** del menú contextual de la selección.



Cuando inserta una fila (o una columna), el formato del elemento agregado retoma el del elemento precedente. Esta opción puede modificarse haciendo clic en el botón



(que aparece al lado del elemento agregado) y seleccionando luego la opción **El mismo formato de arriba/abajo** (para las filas), **El mismo formato de la derecha/izquierda** (para las columnas) o la opción **Borrar formato**.

Eliminar filas y columnas

- ❑ Seleccione las filas (o columnas) que desea eliminar haciendo clic sobre los números de fila (o las letras de columna).
- ❑ Active la pestaña **Inicio** y haga clic en el botón **Eliminar** del grupo **Celdas**, use el método abreviado **Ctrl** - o bien active la opción **Eliminar** del menú contextual de la selección.

Modificar el ancho de columna y el alto de fila

- ❑ Seleccione las columnas a las que desea dar el mismo ancho o las filas a las que desea dar el mismo alto. Si se trata de una sola columna o fila, no es necesario seleccionarla.
- ❑ Señale la línea vertical situada a la derecha de una de las columnas (o la línea horizontal situada debajo del número de una de las filas).

Las hojas de cálculo

Observe el nuevo aspecto del puntero del ratón.

	A	B	C
1	PEDIDO DE CAMISETAS Y DORSALES		
2			
3	Referencia	Descripción	Precio
4	DJ48	Dorsal benjamín rojo	28,99 €

- ☐ Ahora haga clic y arrastre sin soltar el botón del ratón.
El nuevo ancho (alto) se representa con una línea de puntos y su valor aparece en una etiqueta.
 - ☐ Suelte el botón del ratón cuando alcance el ancho (alto) que estime conveniente.
 - ☞ El ancho de las columnas se calcula en número de caracteres (y en píxeles), y el alto de fila, en puntos (y en píxeles).
-
- 💡 Para ahorrar memoria en su ordenador, «aligere» las hojas de cálculo usando esta técnica, y no insertando filas y columnas.
-

Ajustar el ancho de columna y el alto de fila

Los anchos de columna se calculan en función de la entrada de celda más larga de la columna y los altos de fila se basan en la entrada de celda más alta de la fila.

- ☐ Seleccione, si es preciso, las filas o columnas en cuestión.
- ☐ Para el ancho de columna, haga doble clic en la línea vertical situada a la derecha de la letra de la columna.
Para el alto de fila, haga doble clic en la línea horizontal situada debajo del número de la fila.

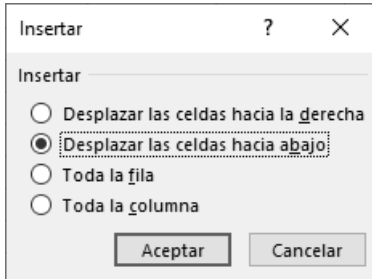
Insertar celdas vacías

Las celdas se insertarán debajo o a la izquierda del rango de celdas seleccionadas.

- ☐ Seleccione tantas celdas como desee insertar.

Filas, columnas y celdas

- ☐ Active la pestaña Inicio, abra la lista del botón Insertar del grupo Celdas y haga clic en la opción Insertar celdas, o use el método abreviado **Ctrl** + o bien active la opción Insertar del menú contextual de la selección.

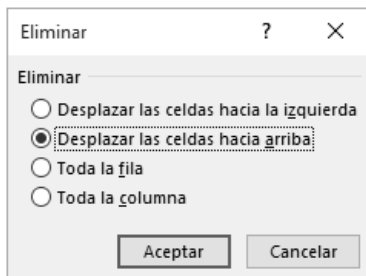


- ☐ Active la primera o la segunda opción para indicar de qué manera separar las celdas existentes tras insertar las nuevas.
- ☐ Confirme pulsando Aceptar.

💡 Para insertar una única celda encima de otra, haga clic en la celda y después en el botón Insertar del grupo Celdas.

Eliminar celdas

- ☐ Seleccione las celdas que desea eliminar.
- ☐ Active la ficha Inicio, abra la lista del botón Eliminar del grupo Celdas y haga clic en la opción Eliminar celdas o active la opción Eliminar del menú contextual de la selección.



- ☐ Active la primera o la segunda opción para indicar de qué manera deben moverse las celdas existentes tras eliminar las celdas seleccionadas.

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 **Capitulo08_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. Según 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 **SegúnValor**.