

Descripción del principio en que se basan las fórmulas de cálculo

- ☐ Las fórmulas de cálculo efectúan cálculos con los valores contenidos en la hoja.
- ☐ Las fórmulas empiezan con un signo de igual (=).
- ☐ Las fórmulas pueden contener los siguientes elementos:

- referencias de celda,
- operadores de cálculo:

Operadores matemáticos:

+ para la suma

- para la resta

/ para la división

* para la multiplicación

% para calcular un porcentaje (p. ej.: =5% da como resultado 0,05)

^ para elevar a la potencia (p. ej.: =2^3 da como resultado 8)

Operadores de comparación: el resultado es un valor lógico: VERDADERO o FALSO:

= igual a (p. ej.: =30=40 da como resultado FALSO)

< inferior a (p. ej.: =28<35 da como resultado VERDADERO)

<= inferior o igual a

> superior a

>= superior o igual a

<> diferente de

Operadores de concatenación de texto: & concatena cadenas de caracteres:

(p. ej.: ="Oeste"&" y "&"Norte" da como resultado "Oeste y Norte").

Los operadores de referencia permiten combinar rangos de celdas:

: (dos puntos). P. ej.: B1:C4 representa el rango de celdas de B1 a C4.

; (punto y coma). P. ej.: B1;C4 representa la celda B1 y la celda C4.

- las constantes, es decir, los valores que no se calculan y, por tanto, no cambian (por ejemplo, el número 1210 o el texto "Totales trimestrales" son constantes).
- algunas funciones de cálculo son fórmulas preescritas que toman uno o varios valores, ejecutan una operación y devuelven uno o varios valores; por ejemplo, la función PROMEDIO calcula el promedio de un grupo de celdas.





Crear una fórmula de cálculo simple

Se trata de elaborar un cálculo que integre las referencias de celdas, los operadores de cálculo y las constantes.

- ☒ Haga clic en la celda donde desea introducir la fórmula y ver el resultado.
- ☒ Escriba un signo de igual =.
- ☒ Elabore la fórmula teniendo en cuenta estas instrucciones:
 - para integrar el contenido de una celda, haga clic en la celda en cuestión o introduzca su referencia (por ejemplo, C4);
 - para integrar un operador de cálculo o una constante, introduzca el dato correspondiente (por ejemplo, * para multiplicar);
 - si utiliza varios operadores, defina, si es preciso, el orden de prioridad para reagrupar valores usando paréntesis.

	C	D	E
2			
3	PRECIO UNITARIO	TOTAL (SIN IVA)	TOTAL (IMP. INCLUIDOS)
4	8,60 €	129,00	=(D4*20%)+D4
5	4,10 €	86,10	
6	7,60 €	136,80	
7	10,50 €	493,50	
8			

Observará la evolución de la fórmula en la barra de fórmulas. En este caso se trata de una fórmula que permite calcular el coste total de un artículo, IVA incluido.

- ☒ Confirme la fórmula pulsando la tecla  o haciendo clic en la herramienta  de la barra de fórmulas.
- ☞ Las fórmulas de cálculo se calculan de nuevo de forma predeterminada al modificar los valores que intervienen en ellas. Para bloquear ese nuevo cálculo automático, active la opción Manual del botón Opciones para el cálculo (pestaña Fórmulas - grupo Cálculo) (puede acceder también a las Opciones de cálculo a través de las Opciones de Excel: pestaña Archivo - Opciones - Fórmulas - zona Cálculo de libro). Si desactiva el modo de cálculo automático, para volver a activar el cálculo manual del libro completo, utilice la herramienta Calcular ahora  (o **F9**) y para calcular manualmente la hoja activa, utilice la herramienta Calcular la hoja  (o **Mayús F9**) de la pestaña Fórmulas - grupo Cálculo).

Al modificar una fórmula de cálculo, las referencias de las celdas que la componen aparecen en colores diferentes en la barra de fórmulas. En la hoja de cálculo, las celdas o rangos de celdas afectados por la fórmula aparecen rodeados de un borde del mismo color.



No olvide que para copiar una fórmula en celdas contiguas es posible usar el indicador de relleno, situado en la esquina inferior derecha de la celda activa, arrastrándolo sobre las celdas o haciendo doble clic en él si las celdas de la columna izquierda contienen datos (véase Copiar y mover - Copiar un contenido en celdas contiguas).

Convertir en absoluta una referencia de celda en una fórmula

Esta técnica permite fijar la referencia de una celda en una fórmula, de forma que no se modifique al copiar la fórmula.

- ☐ Empiece a teclear la fórmula y deténgase cuando se seleccione la celda cuya referencia desea hacer absoluta.
- ☐ Pulse la tecla **F4**.

La referencia de la celda muestra entonces el carácter \$ delante de la referencia de columna y del número de fila.

	B	C	D	E	F	G	H
1	0,30 €						
2	NOMBRE	SERVICIO	KM	FACTURA			TOTAL
3				KM	PARKING	PEAJE	FACTURA
4	Marcos	Marketing	150	=D4*\$B\$1	52	155	
5	Prudencio	Comunicación	258		35	499	


*Al pulsar la tecla **F4**, obtendrá una referencia absoluta de celda (**\$B\$1**); si pulsa **F4** una segunda vez, solo se hará absoluta la referencia de la fila (**B\$1**); si pulsa **F4** una tercera vez, se hará absoluta la referencia de la columna (**\$B1**) y, si pulsa una cuarta vez, la referencia se convierte en relativa (**B1**).*

- ☐ Pulse la tecla **F4** tantas veces como sea necesario para convertir en absoluto el elemento deseado.
- ☐ Si es preciso, acabe de introducir la fórmula y confirme.

Los cálculos

En nuestro ejemplo, hemos copiado la celda E4 en las celdas E5 a E10; observe que la referencia absoluta (B1) permanece fija en las diversas fórmulas, contrariamente a las otras celdas. En este ejemplo, únicamente podría haberse fijado la referencia a la fila, ya que hemos copiado hacia abajo.

	A	B	C	D	E	F	G	H
1	Precio:	0,30 €						
2					FACTURA			TOTAL
3	APELLIDO	NOMBRE	SERVICIO	KM	KM	PARKING	PEAJE	FACTURA
4	Álvarez	Marcos	Marketing	150	=D4*\$B\$1	52	155	
5	Azcona	Prudencio	Comunicación	258	=D5*\$B\$1	35	499	
6	Benítez	Juan	Comercial	89	=D6*\$B\$1	0	217	
7	Caimán	Menchu	Comercial	452	=D7*\$B\$1	47	69	
8	Eire	Tomás	Marketing	579	=D8*\$B\$1	38	248	
9	Estampa	Guillermo	Personal	785	=D9*\$B\$1	89	234	
10	Gokiri	Dalia	Marketing	89	=D10*\$B\$1	0	0	
11								

Para ilustrar esta función, mostramos las fórmulas en lugar de los resultados en las celdas (herramienta Mostrar fórmulas  de la pestaña Fórmulas - grupo Auditoría de fórmulas).

Para utilizar una referencia absoluta en una fórmula, también puede dar primero un nombre a la celda (véase Poner nombre a los rangos de celdas, en el capítulo Rangos con nombre) y referirse a esta celda en la fórmula de cálculo (véase Usar zonas con nombre en las fórmulas).

Introducir una fórmula multihoja

Esta técnica permite insertar en una hoja fórmulas (llamadas fórmulas 3D) que hacen referencia a celdas de una o varias hojas diferentes.


- ☐ Active la celda en la que desea que aparezca el resultado.
- ☐ Introduzca el signo =
- ☐ Empiece a escribir la fórmula y, cuando le parezca oportuno, haga clic en la ficha de la hoja, seleccione la celda o celdas que desee y concluya la fórmula.
- ☐ Confirme.

En este caso, la celda E3 de esta hoja suma el contenido de las celdas E3 de las hojas "Semestre 1" y "Semestre 2".

Total año				
Código	PRODUCTOS	Categoría	IMPORTE	
B5001	Surtido	GALLETAS	1778,00	
CF001	Almendrado	CONFITERÍA		
CH003	Chocolate blanco	CHOCOLATE		
CH006	Chocolate caramelo	CHOCOLATE		
CH001	Chocolate negro	CHOCOLATE		

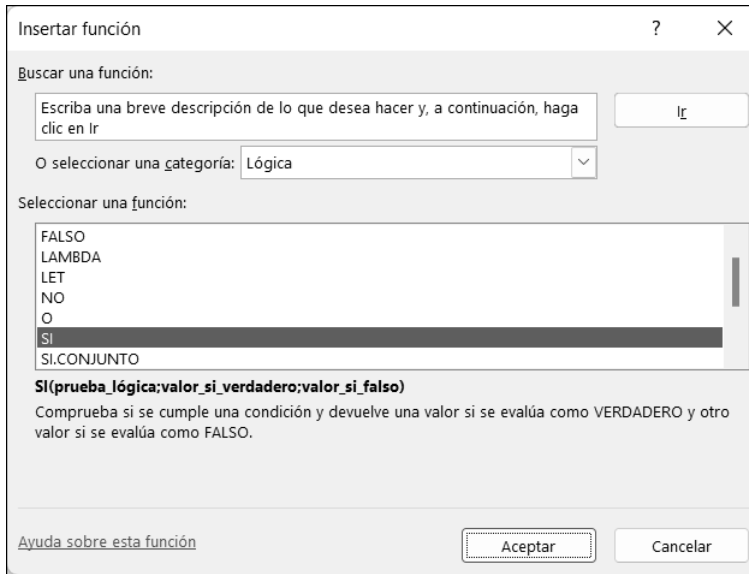
También es posible elaborar fórmulas multilibro. Para ello deberán estar abiertos todos los libros que intervengan en el cálculo. Para desplazarse hasta una celda ubicada en una hoja de otro libro, pulse el botón **Cambiar ventanas** de la pestaña **Vista** (o el icono del archivo en la barra de tareas) a fin de activar el libro correspondiente.



Usar las funciones de cálculo

- ☐ Active la celda en la que desea que aparezca el resultado.
- ☐ Haga clic en la herramienta **Insertar función** , situada en la barra de fórmulas, o en la pestaña **Fórmulas**, o pulse **Mayús** **F3**.
- ☐ En el cuadro de diálogo **Insertar función**, abra la lista desplegable **O seleccionar una categoría** si desea ver una categoría concreta de funciones.
La categoría Usadas recientemente muestra una lista con las funciones usadas por usted y también con las más habituales. La categoría Todo muestra todas las funciones disponibles.
- ☐ Para buscar una función concreta existen dos posibilidades: introducir en el cuadro **Buscar una función** el nombre exacto de la función o bien una descripción de lo que desea hacer con ella y luego confirmar la búsqueda pulsando el botón **Ir** o la tecla **↵**.
- ☐ Haga clic en la función buscada dentro del cuadro **Seleccionar una función** para seleccionarla.

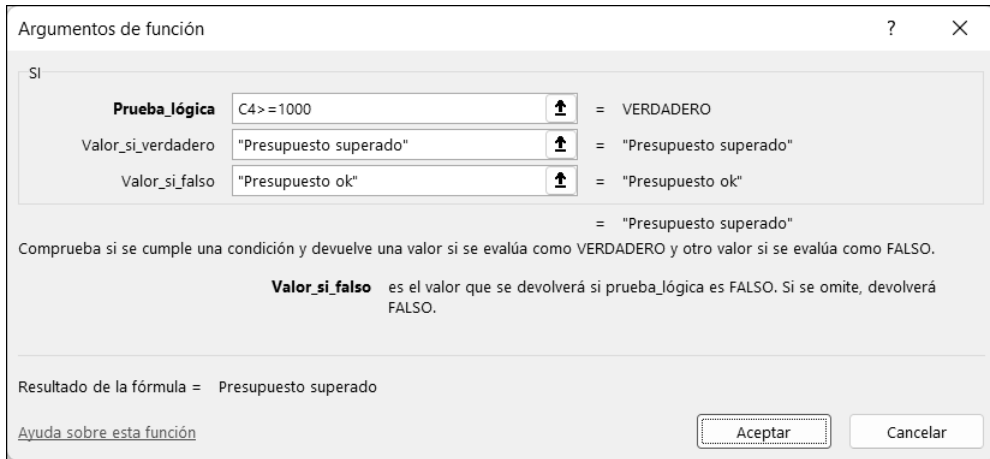
Los cálculos

Al seleccionar una función, su sintaxis y su descripción aparecen en la parte inferior del cuadro.




- ☐ Si es preciso, haga clic en el vínculo **Ayuda sobre esta función** para consultar la ayuda de Excel referente a la función seleccionada.
 - ☐ Haga clic en **Aceptar** para activar el cuadro de diálogo **Argumentos de función**.
 - ☐ Para definir los argumentos de la función:
 - Haga clic en el cuadro de texto correspondiente y, a continuación, haga clic en el botón .
 - Seleccione la celda o celdas correspondientes al argumento en la hoja de cálculo.
 - Haga clic en el botón  para ver de nuevo el cuadro de diálogo.
- Otra posibilidad es introducir directamente un argumento.*

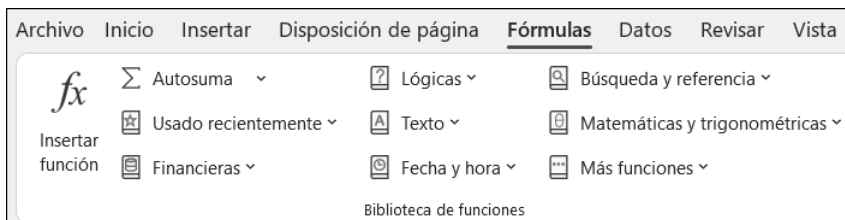
La función *SI* permite crear una fórmula condicional: en este ejemplo, si el valor de la celda C4 es superior o igual a 1000, aparecerá el texto "Presupuesto superado" en las celdas; en caso contrario, es el texto "Presupuesto conforme" el que se mostrará.



☐ Haga clic en **Aceptar** cuando haya definido todos los argumentos.

☞ Es posible insertar funciones dentro de fórmulas o de otras funciones. Para ello, empiece por introducir la fórmula y, cuando le parezca oportuno, haga clic en  (de la zona **Nombre**, situado a la izquierda de la barra de fórmulas) para ver la lista con las últimas funciones usadas y la opción **Más funciones** para acceder a la lista completa de funciones. También puede usar el botón **Recientes** del grupo **Biblioteca de funciones** (pestaña **Fórmulas**).

☞ Para insertar funciones utilizando el asistente, puede activar la pestaña **Fórmulas**, hacer clic en uno de los botones del grupo **Biblioteca de funciones** (agrupan las funciones por tipo) y hacer clic sobre la función en cuestión.



A. Objetivos del capítulo

Cuando quiera ejecutar un programa según ciertas condiciones, por ejemplo, para poner un renglón en amarillo 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.

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.