

MACROS EN EXCEL 97

- 1.- Como escribir una macro utilizando el Editor de Visual Basic
- 2.- Como ejecutar una macro
- 3.- Macros con libros
- 4.- Movernos por un libro
- 5.- Elementos repetidos
- 6.- Comparación de columnas
- 7.- Crear una barra de herramientas personalizada
- 8.- Macro para convertir Pesetas a Euros y viceversa

1.- Como escribir una macro utilizando el Editor de Visual Basic.

Las macros que vamos a ver a lo largo de este curso las vamos a ir programando directamente con el Editor de *Visual Basic* que viene incorporado con el mismo *Excel*.

Para ponerlo en funcionamiento deberemos realizar los siguientes pasos:

1.- Dentro de **Herramientas - Macro** deberemos escoger la opción: **Editor de Visual Basic (Alt + F11)**. Una vez seleccionada se abrirá una nueva ventana con dos ventanas acopladas a la izquierda. La superior es la ventana que llamamos: de **Proyecto (Ctrl+R** para verla) y la inferior la de **Propiedades (F4** para verla). En la primera aparecerán los elementos que forman parte de cada **Proyecto** (grupo de macros y hojas de Excel) y la segunda son las propiedades de los objetos que se pueden incorporar a nuestras macros. Más adelante veremos como se utilizan estas dos ventanas y todas sus características. Vamos a prepararnos para poder escribir nuestra primera macro.

2.- Escoge la opción **Módulo** del menú **Insertar**. Observa como en la ventana de Proyecto aparece una nueva carpeta llamada **Módulos** y en su interior un nuevo elemento llamado **Módulo1**. Dentro de este módulo será donde guardemos las *macros* que creamos.



También podrás ver como la parte derecha de la ventana ahora es completamente blanca. Aquí es donde podemos escribir las instrucciones que formarán parte de nuestra *Macro*. Vamos a crear una *Macro* y esta nos servirá de ejemplo para ver como se deben escribir. Crearemos una *Macro* muy sencilla la cual nos servirá para que la página activa pase a ser la segunda.

3.- En la página en blanco de la derecha escribe lo siguiente:

```
Sub CambiarDeHoja()  
Worksheets(2).Activate  
End Sub
```

Ya tenemos escrita nuestra primera *Macro*.

El comando **Sub** indica el principio de la *Macro*, mientras que **End Sub** marca el final de esta. A continuación del comando **Sub** hemos puesto el nombre que deseamos dar a nuestra *Macro*. Observa como después del nombre aparece un paréntesis que se abre y otro que se cierra. Entre el principio y el final de la *Macro* escribimos las instrucciones que deseamos se realicen en esta *Macro*. Si deseamos guardar la *Macro* simplemente deberemos guardar el libro activo con el nombre que deseamos. Ambos elementos se guardarán juntos.

En el siguiente capítulo veremos como podemos poner en funcionamiento una *Macro* creada por nosotros.

2.- Como ejecutar una Macro

Una Macro escrita por nosotros se ejecuta exactamente igual que si se tratase de cualquier otra *Macro*.

- 1.- Abre el libro donde tienes guardada la Macro que creamos en el capítulo anterior.
- 2.- Despliega el menú: **Herramientas - Macro**. Una vez allí podrás ver como aparece un pequeño submenú.
- 3.- Selecciona la opción **Macros...** (**Alt+F8**). Te aparecerá una nueva ventana en la que podrás ver el nombre que nosotros escribimos detrás del comando **Sub**.
- 4.- Haz un clic sobre este nombre y después pulsa el botón **Ejecutar**.

La Macro se pone en funcionamiento, podemos observar como la hoja activa pasa a ser la segunda. Ya hemos visto como crear, guardar y ejecutar *Macros* creadas utilizando el *Editor de Visual Basic*. A partir de este capítulo iremos viendo pequeñas *Macros* que realizarán cosas muy concretas. Estas *Macros* se podrán ir entrelazando para así poder conseguir lo que nosotros deseamos.

3.- Macros con libros.

Vamos a ver tres *Macros* que trabajarán directamente con libros. Con ellas podremos abrir un libro existente, activar un libro y por último crear un libro nuevo.

3.a.- Abrir un libro existente.

Para abrir un libro utilizaremos la siguiente *Macro*:

```
Sub AbrirLibro()
    Workbooks.Open ("C:\Mis documentos\Ejemplo.xls")
End Sub
```

Observa como entre paréntesis y comillas dobles hemos escrito el camino y el nombre del archivo que deseamos abrir. Este archivo debe existir, si no fuera así esta *Macro* nos daría error.

3.b.- Activar un libro ya abierto.

Con esta *Macro* podemos cambiar el libro activo. Es muy importante que el libro ya esté abierto, si no fuera así, la *Macro* daría error.

```
Sub ActivarLibro()
    Workbooks("Ejemplo.xls").Activate
End Sub
```

Observa como en esta *Macro* no indicamos el camino donde se encuentra el archivo, no hace falta ya que este debe estar abierto.

3.c.- Crear un libro nuevo.

Con la siguiente *Macro* podremos crear un libro nuevo. Esta *Macro* es exactamente igual que si activásemos la opción **Archivo - Nuevo**.

```
Sub NuevoLibro()
    Workbooks.Add
End Sub
```

Los nombres de las *Macros* no tienen porqué ser necesariamente los mismos que hemos puesto nosotros. Te recomiendo escribas estas *Macros* y las ejecutes para ver como funcionan.

4.- Movernos por un libro.

Vamos a ver Macros que nos permitan movernos por las celdas que contiene un libro.

4.a.- Buscar el final de una lista.

Podemos utilizar esta Macro para situarnos al final de una lista de elementos. Interpretaremos que el final de la misma está en el momento en el que encontramos una celda vacía. Antes de ejecutar la Macro tenemos que seleccionar la primera celda de la lista.

```
Sub Final()
    While ActiveCell.Value <> ""           ' Mientras la Celda Activa no esté vacía
        ActiveCell.Offset(1, 0). Select    ' Selecciona la Celda de Abajo
    Wend                                     ' Vuelve al While y si la condición sigue siendo Cierta, vuelve a ejecutar su contenido
End Sub
```

' Todo lo que aparece en este color, que sigue a un apóstrofe (') son comentarios que el Compilador de Visual Basic Ignora y no hace falta que los escribáis.

Debemos observar que cuando finaliza esta Macro siempre queda seleccionada la primera celda que está en blanco después de toda la lista.

4.b.- Buscar el final de una lista y situarnos en la última entrada.

Esta Macro es exactamente igual que la anterior pero con la única diferencia que la celda activada es la última entrada de la lista.

```
Sub FinalLista()
    While ActiveCell.Value <> ""           ' Mientras la Celda Activa no esté vacía
        ActiveCell.Offset(1, 0). Select    ' Selecciona la Celda de Abajo
    Wend                                     ' Vuelve al While y si la condición sigue siendo Cierta, vuelve a ejecutar su contenido
    ActiveCell.Offset(-1, 0). Select       ' Selecciona la Celda de Arriba
End Sub
```

En esta Macro se realiza el bucle hasta que se encuentra la celda vacía y después subimos una posición para situarnos en la última entrada de la tabla.

4.c.- Buscar una entrada en una lista y borrar toda la fila.

Vamos a imaginar que tenemos una tabla de datos en la cual nos interesa buscar un elemento determinado y borrar todos los datos que hay en la misma fila. Pongamos el caso que tenemos una tabla con nombre, población y teléfono y deseamos buscar todos los datos de una población determinada y borrar el nombre y la población. Para hacer la búsqueda deberemos situarnos en la columna que se encuentra la población.

```
Sub BorrarFilas()
    While ActiveCell.Value <> ""           ' Mientras la Celda Activa no esté vacía
        If ActiveCell.Value <> "Barcelona" Then ' Si la Celda Activa no es "Barcelona" entonces
            ActiveCell.Offset(1, 0). Select    ' Selecciona la Celda de Abajo
        Else                                     ' Si n(que sí es "Barcelona")
            Selection.EntireRow.Delete         ' Borra la Fila Entera de la Celda Seleccionada
        End If
    Wend                                     ' Vuelve al While y si la condición sigue siendo Cierta, vuelve a ejecutar su contenido
End Sub
```

Es importante hacer notar que si ejecutamos esta Macro no se podrá deshacer el borrado de las filas que se han eliminado con la Macro.

4.d.- Buscar el final de una lista con espacios en blanco dentro de ella.

Imaginemos que tenemos una lista de datos dentro de la que hay celdas en blanco, como mucho una celda en blanco entre dato y dato. Si utilizásemos la primera macro que hemos creado (4.a), ésta interpretaría como final de la lista la primera celda en blanco que encontrara. Realizaremos una macro que interpretará el final de la lista cuando encuentre 2 celdas en blanco seguidas.

```

Sub FinalListaConBlancos()
    Salir = "No"           ' Salir es una variable donde se guarda un valor
    While Salir = "No"
        While ActiveCell.Value <> ""           ' Buscamos la primera celda vacía
            ActiveCell.Offset(1, 0). Select
        Wend
        ActiveCell.Offset(1, 0). Select         ' Seleccionamos la de abajo
        If ActiveCell.Value <> "" Then         ' Si no está vacía
            Salir = "No"                       ' No salimos
        Else                                    ' Pero si no
            Salir = "Si"                       ' Sí salimos
        End If
    Wend
End Sub

```

En esta ocasión la Macro no finalizará hasta que encuentre dos celdas en blanco seguidas.

5.- Elementos repetidos.

En capítulos anteriores hemos visto como movernos por una lista y como eliminar algunos de ellos según nos conviniera. Pues bien, ahora vamos a ver como podríamos utilizar estas dos cosas para poder eliminar elementos repetidos de una lista.

5.a.- Eliminar elementos repetidos en una lista.

Vamos a realizar una macro en la cual nos vayamos desplazando por una lista, hasta encontrar un elemento vacío e ir comprobando si un elemento es igual al anterior, si esto es así lo eliminaremos y "subiremos" un lugar la lista para que así no existan lugares vacíos en la lista.

El único requisito que hay que cumplir para realizar esta macro es que la lista debe estar completamente ordenada.

```

Sub EliminarRepetidos()
    contador = 0           ' En la variable contador vamos a guardar el nº de repetidos encontrados
    valor = ActiveCell.Value ' En la variable valor almacenamos el contenido de la Celda Actual
    ActiveCell.Offset(1, 0). Select ' Seleccionamos la Celda de abajo
    While ActiveCell.Value <> "" ' Mientras no llegue al final de la lista (Celda vacía)
        If ActiveCell.Value = valor Then ' Si el valor de la celda actual es igual al que tengo guardado
            Selection.Delete Shift:=xlUp ' Borro la celda seleccionada y desplazo hacia arriba el resto
            contador = contador + 1      ' Incremento el contador
        Else                               ' Si no era igual al guardado
            valor = ActiveCell.Value     ' Guardo el Actual
            ActiveCell.Offset(1, 0). Select ' Y sigo avanzando por la lista
        End If
    Wend
    MsgBox("Se han encontrado " & contador & " elementos repetidos",1,"Número de repetidos")
End Sub

```

En esta macro estamos utilizando una variable llamada **contador** la cual nos servirá para contabilizar el número de elementos repetidos que encontramos en la lista. Este valor lo mostramos al finalizar la macro y lo incluimos dentro de un mensaje. La función MsgBox nos presenta una ventanita con el mensaje que queramos y el operador & sirve para concatenar cadenas de texto.

5.b.- Eliminar elementos repetidos en una lista y realizar un registro.

En la primera macro que hemos mostrado en esta capítulo eliminamos todos los elementos que aparecen repetidos dentro de una lista quedando uno solo de todos los elementos repetidos. Según para que realizásemos este proceso nos podría interesar llevar un control de la cantidad de elementos que estaban repetidos, por esto sería conveniente crear en una hoja nueva un registro de cual es el elemento repetido y el número de veces que aparecía dentro de la lista.

La macro siguiente necesita que los elementos estén ordenados en una primera hoja y que la segunda hoja esté vacía, ya que será aquí donde se realizará el registro de los elementos repetidos.

```

Sub EliminarRepetidosYRegistro()
    contador = 1           ' Asumo que tengo al menos un elemento de cada (Si hay repetidos serán más)
    valor = ActiveCell.Value ' Continuo como la Macro anterior
    ActiveCell.Offset(1, 0). Select
    While ActiveCell.Value <> ""
        If ActiveCell.Value = valor Then           ' Si encuentro un duplicado
            ActiveSheet.Next.Select                ' Selecciono la hoja siguiente (la de la estadística)
            If ActiveCell.Value <> valor Then       ' Si todavía no está escrito el valor duplicado
                ActiveCell.Offset(1, 0). Select    ' Selecciono la celda de abajo
                ActiveCell.Value = valor          ' Escribo el valor duplicado que voy a eliminar
            End If
            ActiveSheet.Previous.Select            ' Vuelvo a la hoja anterior
            Selection.Delete Shift:=xlUp          ' Borro la celda y desplazo hacia arriba el resto
            contador = contador + 1              ' Incremento el contador de duplicados
        Else                                       ' Si ya no hay más duplicados
            If contador > 1 Then                  ' Y tengo más de uno (vamos, que ha habido duplicados)
                ActiveSheet.Next.Select          ' Selecciono la hoja siguiente (la de la estadística)
                ActiveCell.Offset(0, 1). Select   ' Selecciono la celda de la derecha
                ActiveCell.Value = contador      ' Escribo el valor de repetidos almacenado en contador
                ActiveCell.Offset(0, -1). Select  ' Vuelvo a la celda de la izquierda
                ActiveSheet.Previous.Select      ' Vuelvo a la hoja anterior
            End If
            contador = 1                          ' Como ya no hay más duplicados de este valor, pongo el contador a 1 para contar
            valor = ActiveCell.Value              ' duplicados de siguientes valores
            ActiveCell.Offset(1, 0). Select
        End If
    Wend
    If contador <> 1 Then                          ' Cuando se acaba la lista y tengo duplicados tengo que escribir la estadística
        ActiveSheet.Next.Select                    ' Selecciono la hoja siguiente (la de la estadística)
        ActiveCell.Offset(0, 1). Select            ' Selecciono la celda de la derecha
        ActiveCell.Value = contador              ' Escribo el valor de repetidos almacenado en contador
        ActiveCell.Offset(0, -1). Select          ' Vuelvo a la celda de la izquierda
        ActiveSheet.Previous.Select              ' Vuelvo a la hoja anterior
    End If
End Sub

```

Sería conveniente prestarle mucha atención a esta macro para poder ver y entender como está realizada.

6.- Comparación entre columnas.

En este capítulo vamos a ver una macro la cual nos permitirá comparar dos columnas de elementos, eliminando de la primera los elementos que existan en la segunda. Esta macro me la pidieron para hacer borrar muchos elementos de una lista sin tener que irlos buscando uno a uno. En la primera columna están todos los elementos que forman parte de la lista y en la segunda columna se van colocando los elementos que se desean buscar y eliminar de la primera.

A continuación escribiremos la macro completa y después explicaremos que realiza cada una de las líneas.

```

1  Sub Repetidos()
2      Range("b1").Select
3      Posicion = 1
4      While ActiveCell.Value <> ""
5          valorcomparacion = ActiveCell.Value
6          Range("a1").Select
7          Salir = "no"
8          While ActiveCell.Value <> "" And Salir = "no"
9              If ActiveCell.Value = valorcomparacion Then
10                 respuesta = MsgBox("¿Deseas borrar esta entrada?", 4, "¡¡Encontrado!!")
11                 If respuesta = vbYes Then
12                     Selection.Delete Shift:=xlUp
13                 End If
14                 Salir = "si"
15             Else
16                 ActiveCell.Offset(1, 0). Select
17             End If
18         Wend
19         Posicion = Posicion + 1
20         Range("b1").Select
21         ActiveCell.Offset(Posicion - 1, 0). Select
22     Wend
23 End Sub

```

Antes de realizar la explicación tenemos que dejar muy claro que la **columna A**, será donde tengamos todos los datos de nuestra lista y la **columna B**, los datos que deseamos buscar y eliminar de la primera columna.

Línea 2: Con esta instrucción nos situamos en la primera celda de la columna B para empezar a buscar los datos que deseamos borrar.

Línea 3: Creamos una variable llamada **Posición** para controlar en que fila nos encontramos de la segunda columna.

Línea 4: Aquí creamos un bucle que se repetirá mientras que no se terminen los elementos a buscar.

Línea 5: Creamos una variable llamada **valorcomparacion** con la que trabajaremos para ir comparando los elementos de la primera y segunda columna. El valor de la celda en la que nos encontramos en la columna B, pasa a estar en la variable **valorcomparacion**.

Línea 6: Cambiamos a la columna A para iniciar el proceso de comparación.

Línea 7: Creamos una nueva variable llamada **Salir** con la que controlaremos si debemos salir del bucle o no. Solamente saldremos en el momento en el que se encuentre un elemento que está en la segunda y primera columna.

Línea 8: Aquí iniciamos un bucle que se repetirá hasta que se llegue al final de la primera columna, (caso que se dará cuando no existan elementos comunes) o hasta que se encuentre un elemento común, (esto nos lo indicará la variable **Salir**).

Línea 9: En esta línea preguntaremos si la celda en la que nos encontramos (primera columna) es igual que el valor que tenemos dentro de la variable: **valorcomparacion**.

Línea 10: Si se ha producido la coincidencia de elementos procedemos a preguntar si realmente deseamos borrar este elemento.

Línea 11: Miramos si la respuesta del usuario ha sido afirmativa.

Línea 12: Si el usuario contesta afirmativamente eliminamos la celda y desplazamos la lista hacia arriba, de esta forma no tenemos huecos en medio de la lista.

Línea 13: Terminamos la estructura **If** que hemos iniciado en la línea 11.

Línea 14: Como se ha localizado un elemento coincidente pondremos la variable **Salir** con valor **Si**.

Línea 15: Aquí empezamos el caso en el que el valor que tenemos seleccionado en la primera columna con el valor de la variable **valorcomparacion** no sea coincidente.

Línea 16: Si no coinciden los datos lo que hacemos es avanzar una posición dentro de la primera columna.

Línea 17: Aquí terminamos la estructura **If** iniciada en la línea 9.

Línea 18: Terminamos el bucle iniciado en la línea 8.

Línea 19: Incrementamos en uno el valor de la variable **Posicion** ya que estamos avanzando una posición dentro de la segunda columna.

Línea 20: Pasamos a la segunda columna para continuar con la comparación de los elementos.

Línea 21: Avanzamos en la segunda columna tantas veces como se nos indique en la variable **Posicion** para iniciar nuevamente el proceso de comparación entre los elementos de ambas columnas.

Línea 22: Terminamos el bucle principal de esta macro.

Línea 23: Terminamos la macro.

Esta macro es bastante completa, te recomiendo repasarla cuantas veces creas oportunas hasta que quede completamente clara. Para probar la macro simplemente deberás copiar una serie de elementos en la primera columna y en la segunda poner algunos de los elementos que están en la primera.

7.- Barra de herramientas personalizada.

Hasta este momento hemos visto como poner en funcionamiento una macro utilizando el menú de herramientas **Herramientas - Macro** tal y como explicamos en capítulos anteriores. En muchas ocasiones puede ser que realicemos una macro la cual tengamos que poner en funcionamiento muchas veces con lo que el hecho de abrir el menú y buscar dicha macro nos puede ser un poco engorroso.

En este capítulo vamos a ver una forma de crear una nueva barra de herramientas en la que pondremos un botón para que al pulsarlo se ponga en funcionamiento la macro.

Explicaremos los pasos necesarios para conseguir esto.

7.a.- Crear una barra de herramientas nueva.

- 1.- Abrir el menú **Ver - Barras de herramientas** y seleccionar la opción: **Personalizar**. A continuación aparecerá una nueva ventana llamada: **Personalizar**.
- 2.- De esta ventana pulsar sobre el botón: **Nueva...** Nos aparecerá una ventana llamada: **Nueva barra de herramientas** en la que deberemos especificar el nombre que le deseamos dar a nuestra barra para poderla diferenciar del resto.
- 3.- Para nuestro ejemplo podríamos poner como nombre: **Mis macros**.
- 4.- A continuación pulsar **Aceptar**. Podremos observar como en la lista de barras de herramientas disponibles aparece el nombre que acabamos de poner. A la derecha del nombre aparecerá una indicación conforme esta barra está visible. Si no la ves puedes mover la ventana **Personalizar**. Ahora ya tenemos nuestra nueva barra de herramientas creada. Veamos como podemos crear un botón dentro de esta barra para que al pulsarlo se ejecute una de nuestras macros.

7.b.- Crear un botón en la nueva barra de herramientas.

- 5.- Dentro de la ventana **Personalizar** deberemos activar la pestaña: **Comandos**.
- 6.- En la lista de Categorías deberíamos hacer un clic sobre: **Macros**. En la parte derecha de esta misma ventana aparecerán dos opciones: **Personalizar elemento de menú** y **Personalizar botón**, en esta ocasión utilizaremos la segunda de las opciones.
- 7.- Deberemos hacer un clic sobre la opción **Personalizar botón** para seleccionarla.
- 8.- Seguidamente deberemos desplazar este botón a la nueva barra de herramientas que hemos creado. Para ello simplemente deberemos situarnos encima, pulsaremos el botón izquierdo del ratón y mientras lo tenemos pulsado nos desplazaremos hasta situarnos encima de la barra de herramientas que hemos creado anteriormente.
- 9.- Una vez situado el cursor dentro de la barra de herramientas soltaremos el botón. Podremos ver como dentro de la barra: **Mis macros** aparece el botón que hemos arrastrado. Ahora que ya tenemos creado el botón en nuestra barra de herramientas, vamos a personalizarlo para que cumpla con su función. Observa como en la ventana actual está activado el botón: **Modificar selección**.
- 10.- Pulsa este botón, desplázate hasta la opción: **Cambiar imagen del botón**.
- 11.- De la lista, selecciona el icono que más te guste. Observa como la imagen del botón de nuestra barra de herramientas ha cambiado.
- 12.- Pulsa nuevamente el botón: **Modificar selección**. Ahora lo que cambiaremos es el texto que aparecerá en el momento en el que pongamos el ratón encima de este botón, así de esta forma siempre podremos recordar que función hace dicho botón.
- 13.- Selecciona la opción **Nombre**: y escribe: **Repetir con informe**. Ahora lo único que nos falta es asignar la macro a este botón.

7.c.- Asignar una macro a un botón personalizado.

14.- Abre nuevamente el botón: **Modificar selección**.

15.- Selecciona la opción: **Asignar macro...** Aparecerá una nueva ventana llamada: **Asignar macro**.

16.- De esta nueva ventana escoge la macro que deseas asignar a este botón. En nuestro caso seleccionaremos: **EliminarRepetidosYRegistro**, macro que servía para eliminar elementos repetidos de una lista y crear un informe con el número de repeticiones.

17.- Una vez seleccionada la macro, pulsa **Aceptar**.

18.- Cierra la ventana **Personalizar**. Ahora ya puedes poner la barra en el lugar que más te convenga y utilizar este botón siempre que quieras.

7.d.- Peculiaridades a saber sobre las macros.

Para que una Macro pueda ejecutarse (que aparezca en la lista de macros, Alt + F8) ha de estar en un libro abierto. Pero si está asignada a un botón, no es necesario. El botón sabe en que libro se encuentra, y de no estar abierto el libro que la contiene, lo abriría automáticamente, quedando abierto cuando finaliza la ejecución de la Macro.

8.- Macro para convertir Euros a pesetas y viceversa.

Desde hace unos meses tenemos necesidad de convertir nuestras hojas de cálculo, normalmente realizadas en **pesetas**, a **euros**, con lo que deberemos realizar multitud de conversiones.

Las macros que explicaremos a continuación nos facilitarán mucho este trabajo ya que nos permitirán convertir la celda en la que nos encontramos a **Pesetas** o **Euros** según decidamos y a la vez cambiará su formato.

Estas macros se pueden realizar para cualquier otra moneda, lo único que deberemos cambiar será el **tipo de conversión** y el formato con el que deseamos trabajar.

Antes de escribir las macros necesarias recordemos que para pasar de pesetas a euros deberemos dividir por **166,386** y para pasar de euros a pesetas deberemos multiplicar por esta cantidad.

8.a.- Conversión de pesetas a euros.

Sub Euro()

valor = ActiveCell.Value / 166.386

ActiveCell.Value = valor

Selection.NumberFormat = "#,##0.00 € ;[Red]-#,##0.00 € "

End Sub

En la primera línea de esta Macro lo único que hacemos es mirar el valor que hay en la celda activa y dividirlo por **166.386**, seguidamente asignamos este valor a una variable. A continuación esta variable la asignamos a la misma celda en la que nos encontramos. De esta forma lo que hacemos es reemplazar un valor por otro. Seguidamente lo único que hacemos es cambiar el formato de la celda pasándola de un formato de pesetas a un formato de Euro, con dos decimales y con el símbolo €.

8.b.- Conversión de euros a pesetas.

```
Sub Peseta()
    ActiveCell.Value = ActiveCell.Value * 166.386
    Selection.NumberFormat = "#,##0 ""pts"" ;[Red]-#,##0 ""pts"" "
End Sub
```

Esta segunda Macro funciona exactamente igual que la primera, podríamos decir que el único cambio que hay es que en esta se realiza una multiplicación y que el formato de celda que se selecciona es de pesetas, sin decimales y con **pts** al final de la cifra. Se ha prescindido de la variable, pues se puede asignar el valor de forma directa.

También hay que destacar que en la máscara del formato se ha encerrado **pts** entre comillas dos veces: **""pts""**. Esto es así porque siempre que aparece un delimitador (las ") dentro de lo delimitado, hay que poner el delimitador dos veces, para indicarle al compilador que no es el final de lo delimitado. Observa que toda la máscara está encerrada entre comillas (las azules ").

Las divisas reconocidas (€, \$, pta, £, ¥, etc.) **no** hacen falta encerrarlas entre comillas dobles, pero las no reconocidas (pts, peras, coches) sí que hay que encerrarlas.

8.c.- Conversión de pesetas a euros, en muchas celdas de golpe.

Las dos Macros anteriores sólo convierten una sola celda cada vez. Como eso sería muy tedioso para un rango de muchas celdas, diseñaremos una Macro que convierta de una sola vez todas las celdas que tenga seleccionadas el usuario. Para ello vamos a emplear un bucle que se realizará tantas veces como elementos existan en la colección (tantas veces como celdas haya en la colección de celdas seleccionadas).

```
Sub Euro_Varios()
    For Each celda In Selection.Cells
        celda.Value = celda.Value / 166.386
    Next
    Selection.NumberFormat = "#,##0.00 € ;[Red]-#,##0.00 € "
End Sub
```

En cada iteración del bucle, la variable *celda* hace referencia a una celda distinta cada vez. Como si tengo *n* celdas seleccionadas, el bucle se realiza *n* veces; las acciones que lleva a cabo el bucle se habrán repetido para cada una de las *n* celdas seleccionadas.

8.d.- Conversión de euros a pesetas , en muchas celdas de golpe.

De forma análoga a la Macro anterior, podemos realizar esta otra que realiza la operación inversa.

```
Sub Peseta_Varios()
    For Each celda In Selection.Cells
        celda.Value = celda.Value * 166.386
    Next
    Selection.NumberFormat = "#,##0 ""pts"" ;[Red]-#,##0 ""pts"" "
End Sub
```

Para facilitar el uso de estas Macros os recomiendo creéis unos botones en vuestra barra de herramientas y les asignéis estas Macros. Así las conversiones serán mucho más cómodas y rápidas.