



FACULTAD DE INGENIERÍA UNAM  
DIVISION DE EDUCACIÓN CONTINUA

# AUTOMATIZACION DE PROCESOS CON MACROS DE EXCEL

Material bibliográfico

Curso: CC46

OCTUBRE, 2006

## TABLA DE CONTENIDO

AUTOMATIZACIÓN DE PROCESOS CON MACROS DE EXCEL.....	
1. Cómo escribir una macro utilizando el Editor de Visual Basic.....	2
2. Cómo ejecutar una Macro.....	4
3. Macros con libros .....	5
4. Movernos por un libro .....	6
5. Movernos por un libro realizando una acción .....	7
6. Elementos repetidos.....	9
7. Elementos repetidos en lista con registro .....	10
8. Macro - Comparación entre columnas.....	12
9. Barra de herramientas personalizada .....	15
10. Crear botones en la barra de herramientas personalizada.....	16
11. Asignar macro a botón.....	17
12. Macro para convertir cantidad a letras (Dólares o Pesos) .....	18
13. Impresión de Formularios con Macros.....	22

## 1. *Cómo 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 del menú **Herramientas - Macro** deberemos escoger la opción: **Editor de Visual Basic**.

Una vez seleccionada se abrirá una nueva ventana con dos ventanas acopladas a la izquierda. La superior es la ventana que llamamos: de **Proyecto** y la inferior la de **Propiedades**. 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 superior 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 creemos.

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
```

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.

## MACROS EN EXCEL

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 la siguiente lección veremos como podemos poner en funcionamiento una Macro creada por nosotros.

## 2. **Cómo ejecutar una Macro**

En la lección anterior vimos cómo podemos crear y guardar una macro. En esta lección veremos cómo ejecutarla y ver lo que esta realiza.

Es recomendable poner el nivel de **Seguridad** a **Alto**, de esta forma evitaremos que se ejecuten macros de libros sin nuestro permiso. El nivel de seguridad lo podremos modificar desde el menú: **Herramientas - Macro - Seguridad...**, escoger la opción: **Alta**.

1.- Abre el libro donde tienes guardada la Macro que creamos en la lección anterior.

2.- Despliega el menú: **Herramientas - Macro**

Una vez allí podrás ver como aparece un pequeño submenú.

3.- Selecciona la opción **Macros...**

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 hora 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 esta lección 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

A partir de este momento vamos a ver pequeñas macros que realizan tareas muy concretas. Para poder ver cómo funcionan simplemente tienes que copiarlas en nuestro **Editor de Visual Basic**, según vimos en la primera lección y después ejecutarla, según vimos en la segunda.

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.

**Abrir un libro existente:** para abrir un libro utilizaremos la siguiente Macro.

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

Observa cómo 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.

**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 AActivarLibro()  
  
    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.

**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
```

Si modificásemos este libro después Excel nos preguntaría si deseamos guardar los cambios.

#### **4. Movernos por un libro**

En esta lección veremos algunas macros que nos permitirán mover por las celdas que contiene un libro. Es recomendable trabajar con libros que contengan datos para así poder ver el efecto que la macro hace sobre la hoja.

**Buscar el final de una lista:** podemos utilizar esta Macro para podernos situar 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.

```
Sub Final()  
  
While ActiveCell.Value <> ""  
  
ActiveCell.Offset(1, 0).Range("A1").Select  
  
Wend  
  
End Sub
```

Deberemos observar que con esta Macro siempre queda activada la primera celda que está en blanco después de toda la lista.

**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 <> ""  
  
ActiveCell.Offset(1, 0).Range("A1").Select  
  
Wend  
  
ActiveCell.Offset(-1, 0).Range("A1").Select  
  
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.

## 5. Movernos por un libro realizando una acción

En esta lección continuaremos con las macros que nos permiten movernos por nuestro documento.

**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 fila que se encuentra la población.

```
Sub BorrarFilas()  
  
While ActiveCell.Value <> ""  
  
If ActiveCell.Value <> "Barcelona" Then  
  
ActiveCell.Offset(1, 0).Range("A1").Select  
  
Else  
  
Selection.EntireRow.Delete  
  
End If  
  
Wend  
  
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.

**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 espacios en blanco, como mucho un espacio en blanco entre dato y dato. Si utilizásemos la primera macro que hemos creado esta interpretaría como final de la lista el primer espacio en blanco que encontrara. Realizaremos una macro que interpretará el final de la lista cuando encuentre 2 espacios en blanco seguidos.



## MACROS EN EXCEL

*Sub FinalListaEspecial()*

*Salir = "No"*

*While Salir = "No"*

*While ActiveCell.Value <> ""*

*· ActiveCell.Offset(1, 0).Range("A1").Select*

*Wend*

*ActiveCell.Offset(1, 0).Range("A1").Select*

*If ActiveCell.Value <> "" Then*

*Salir = "No"*

*Else*

*Salir = "Si"*

*End If*

*Wend*

*End Sub*

## 6. Elementos repetidos

En lecciones anteriores hemos visto cómo movernos por una lista y cómo eliminar algunos de ellos según nos conviniera.

Ahora vamos a ver cómo podríamos utilizar estas dos cosas vistas en lecciones anteriores para poder eliminar elementos repetidos de una lista.

**Eliminar elementos repetidos en una lista:** vamos a realizar una macro 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
```

```
    valor = ActiveCell.Value
```

```
    ActiveCell.Offset(1, 0).Range("A1").Select
```

```
    While ActiveCell.Value <> ""
```

```
        If ActiveCell.Value = valor Then
```

```
            Selection.Delete Shift:=xlUp
```

```
            contador = contador + 1
```

```
        Else
```

```
            valor = ActiveCell.Value
```

```
            ActiveCell.Offset(1, 0).Range("A1").Select
```

```
        End If
```

```
    Wend
```

```
    Respuesta = 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.

## **7. Elementos repetidos en lista con registro**

En esta lección seguiremos con la creación de macros que nos servirán para eliminar elementos repetidos de una lista.

**Eliminar elementos repetidos en una lista y realizar un registro:** en la primera macro que hemos mostrado en esta página eliminamos todos los elementos que aparecen repetidos dentro de una lista quedando uno solo de todos los elementos repetidos. Según para qué 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

    valor = ActiveCell.Value

    ActiveCell.Offset(1, 0).Range("A1").Select

    While ActiveCell.Value <> ""

        If ActiveCell.Value = valor Then

            ActiveSheet.Next.Select

            If ActiveCell.Value <> valor Then

                ActiveCell.Offset(1, 0).Range("a1").Select

                ActiveCell.Value = valor

            End If

        End If

        ActiveSheet.Previous.Select

        Selection.Delete Shift:=xlUp
    
```

## MACROS EN EXCEL

```
contador = contador + 1

Else

  If contador <> 1 Then

    ActiveSheet.Next.Select

    ActiveCell.Offset(0, 1).Range("a1").Select

    ActiveCell.Value = contador

    ActiveCell.Offset(0, -1).Range("a1").Select

    ActiveSheet.Previous.Select

  End If

  contador = 1

  valor = ActiveCell.Value

  ActiveCell.Offset(1, 0).Range("A1").Select

End If

Wend

If contador <> 1 Then

  ActiveSheet.Next.Select

  ActiveCell.Offset(0, 1).Range("a1").Select

  ActiveCell.Value = contador

  ActiveCell.Offset(0, -1).Range("a1").Select

  ActiveSheet.Previous.Select

End If

End Sub
```

## 8. Macro - Comparación entre columnas

En las lecciones anteriores hemos visto como eliminar elementos repetidos existentes en una misma fila. En este ejemplo realizaremos comparaciones entre diferentes columnas, eliminando de la primera los elementos que existan en la segunda.

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 para eliminar de la primera.

A continuación escribiremos la macro completa y después explicaremos que realiza cada una de las líneas. El número que aparece al principio de cada una de las líneas nos servirá como guía en la explicación, no debes copiarlos en el Editor.

```

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).Range("A1").Select

```

## MACROS EN EXCEL

```
17   End If
18   Wend
19   Posicion = Posicion + 1
20   Range("b1").Select
21   ActiveCell.Offset(Posicion - 1, 0).Range("a1").Select
22   Wend
23 End Sub
```

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

En esta lección realizaremos la explicación paso a paso de la macro que hemos escrito anteriormente para la comparación entre dos columnas.

**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 hasta 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**.

## MACROS EN EXCEL

**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 **Si** 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 recomendamos repasarla cuantas veces creas oportunas hasta que quede completamente clara. No hace falta que entiendas completamente las instrucciones, pero sí cuál es su función. Para probar la macro simplemente debes copiar una serie de elementos en la primera columna y en la segunda poner algunos de los elementos que están en la primera.

## 9. Barra de herramientas personalizada

Hasta esta lección hemos visto cómo copiar, guardar y poner en funcionamiento una macro utilizando el menú de herramientas **Herramientas - Macro**.

En muchas ocasiones puede ser que tengamos una macro la cual necesitemos 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 esta lección 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.

### 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. En la siguiente lección aprenderemos cómo podemos crear un botón dentro de esta barra para que al pulsarlo se ejecute una de nuestras macros.



## **10. Crear botones en la barra de herramientas personalizada**

En la lección anterior hemos visto cómo crear una barra de herramientas personalizada para colocar nuestros botones, los cuales ejecutarán una macro.

En esta lección veremos cómo crear estos botones.

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 cómo 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**.

En la siguiente lección asignaremos la macro a este botón.

## **11. Asignar macro a botón**

En lecciones anteriores hemos visto cómo crear una barra de herramientas personalizada y un botón en ella.

En esta lección veremos cómo asignar una macro a este 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**.

A partir de este momento ya puedes utilizar este botón en el momento que desees ejecutar la macro.

## 12. Macro para convertir cantidad a letras (Dólares o Pesos)

'Funciones para convertir de números a letras

'Llamada : Letras(Número,Formato) - Formato 1-Pesos, 2-Dólares

\*\*\*\*\*

Function Unidades(num, UNO)

Dim U

Dim Cad

U = Array("UN", "DOS", "TRES", "CUATRO", "CINCO", "SEIS", "SIETE", "OCHO", "NUEVE")

Cad = ""

If num = 1 Then

If UNO = 1 Then

Cad = Cad & "UNO"

Else

Cad = Cad & "UN"

End If

Else

Cad = Cad & U(num - 1)

End If

Unidades = Cad

End Function

\*\*\*\*\*

Function Decenas(num1, res)

Dim D1

D1 = Array("ONCE", "DOCE", "TRECE", "CATORCE", "QUINCE", "DIECISEIS", "DIECISIETE",

"DIECIOCHO", "DIECINUEVE")

D2 = Array("DIEZ", "VEINTE", "TREINTA", "CUARENTA", "CINCUENTA", "SESENTA",

"SETENTA", "OCHENTA", "NOVENTA")

If num1 > 10 And num1 < 20 Then

Cad1 = D1(num1 - 10 - 1)

Else

Cad1 = D2((num1 \ 10) - 1)

If (num1 \ 10) <> 2 Then

If res > 0 Then

Cad1 = Cad1 & " Y "

Cad1 = Cad1 & Unidades(num1 Mod 10, 0)

End If

## MACROS EN EXCEL

```
Else
  If res = 0 Then
    Cad1 = Cad1 & "E"
  Else
    Cad1 = Cad1 & "I"
    Cad1 = Cad1 & Unidades(num1 Mod 10, 0)
  End If
End If
End If
Decenas = Cad1
End Function
```

\*\*\*\*\*

```
Function Cientos(num2)
  num3 = num2 \ 100
  Select Case num3
    Case 1
      If num2 = 100 Then
        cad2 = "CIEN "
      Else
        cad2 = "CIENTO "
      End If
    Case 5
      cad2 = "QUINIENTOS "
    Case 7
      cad2 = "SETECIENTOS "
    Case 9
      cad2 = "NOVECIENTOS "
    Case Else
      cad2 = Unidades(num3, 0) & "CIENTOS "
  End Select
```

```
  num2 = num2 Mod 100
  If num2 > 0 Then
    If num2 < 10 Then
      cad2 = cad2 & Unidades(num2, num2)
    Else
      cad2 = cad2 & Decenas(num2, num2 Mod 10)
    End If
  End If
  Cientos = cad2
End Function
```

\*\*\*\*\*

## MACROS EN EXCEL

```
Function Miles(num4)
  If (num4 >= 100) Then
    cad3 = Cientos(num4)
  Else
    If (num4 >= 10) Then
      cad3 = Decenas(num4, num4 Mod 10)
    Else
      cad3 = Unidades(num4, 0)
    End If
  End If
  cad3 = cad3 & " MIL "
  Miles = cad3
End Function
```

\*\*\*\*\*

```
Function Millones(cant)
  If cant = 1 Then
    ter = " "
  Else
    ter = "ES "
  End If
  If (cant >= 1000) Then
    cantl = cantl & Miles(cant \ 1000)
    cant = cant Mod 1000
  End If
  If cant > 0 Then
    If cant >= 100 Then
      cantl = cantl & Cientos(cant)
    Else
      If cant >= 10 Then
        cantl = cantl & Decenas(cant, cant Mod 10)
      Else
        cantl = cantl & Unidades(cant, 0)
      End If
    End If
  End If
  Millones = cantl & " MILLON" & ter
End Function
```

\*\*\*\*\*

```
Function decimales(numero As Single) As Integer
  Dim iaux As Integer
  iaux = numero - Application.Round(numero, 2)
  decimales = iaux
End Function
```

## MACROS EN EXCEL

Function letras(cantm As Variant, ByVal mon As Integer) As String  
Dim cents1 As String, num1 As Variant, num2 As Variant

```
num1 = cantm \ 1000000
num2 = cantm - (num1 * 1000000)

cents = (num2 * 100) Mod 100
If cents = 0 Then
    cents1 = "00"
Else
    cents1 = Format(cents)
End If
cantm = cantm - (cents / 100)
If cantm >= 1000000 Then
    cantlm = Millones(cantm \ 1000000)
    cantm = cantm Mod 1000000
End If
If cantm > 0 Then
    If (cantm >= 1000) Then
        cantlm = cantlm & Miles(cantm \ 1000)
        cantm = cantm Mod 1000
    End If
End If
If cantm > 0 Then
    If cantm >= 100 Then
        cantlm = cantlm & Cientos(cantm)
    Else
        If cantm >= 10 Then
            cantlm = cantlm & Decenas(cantm, cantm Mod 10)
        Else
            cantlm = cantlm & Unidades(cantm, 1)
        End If
    End If
End If
If mon = 1 Then
    letras = cantlm & " PESOS " & cents1 & "/100 M.N."
Else
    letras = cantlm & " DOLARES " & cents1 & "/100 U.S.D."
End If
End Function
```

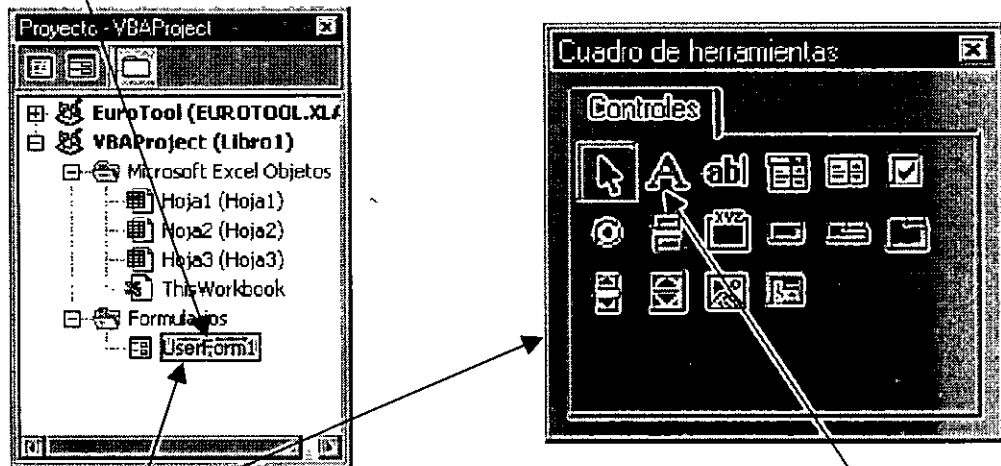
\*\*\*\*\*

```
Sub prueba()
    Dim res As String, num As Single
    num = 50899697.51
    res = letras(num, 1)
End Sub
```

## 13. Impresión de Formularios con Macros

A continuación Muestro como crear un formulario y como programarlo:

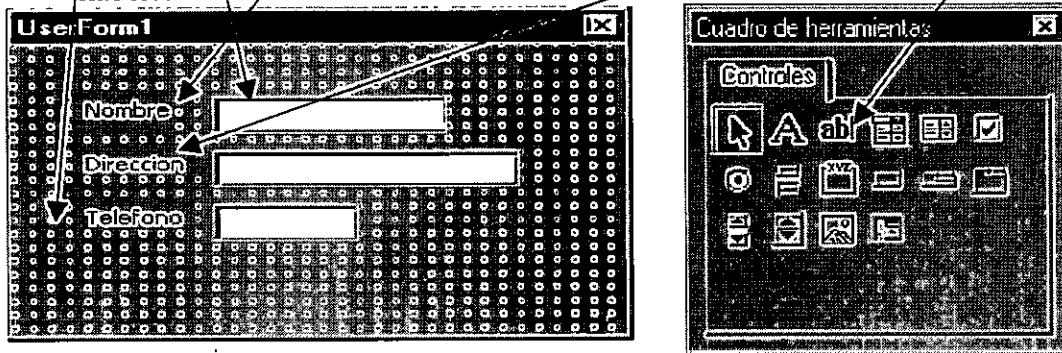
1. Presione La Teclas **Alt + F11**, para entrar al editor de **Visual Basic**.
2. Activa las siguientes opciones:
  - De clic en el **Menú Ver** y elija la opción **Explorador de Proyectos**
  - De clic en el **Menú ver** y elija la opción **Ventana Propiedades**
3. Del **Menú Insertar** elija la Opción **UserForm**. Esto inserta el Formulario que programaremos con controles. En el **Explorador de Proyecto** se observara que se inserto el **UserForm**.



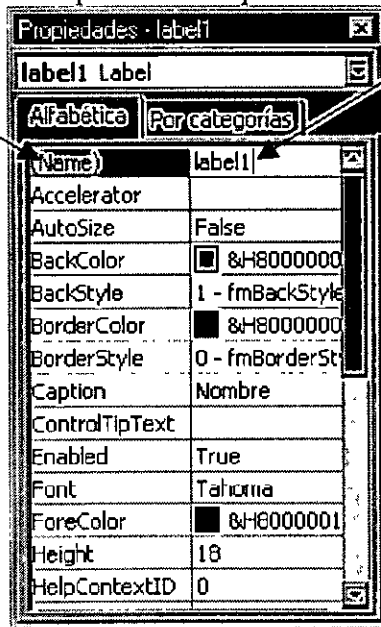
También cuando de clic en el Formulario **USERFORM1** se debe de activar el **Cuadro de Herramientas**, si no se activa de clic en el **Menú Ver** y elija la opción **Cuadro de Herramientas**.

4. Elija del **Cuadro de Herramientas** el Control **Etiqueta** el que tiene la **A** y Arrastre dibujando en el Formulario **USERFORM1** la etiqueta. Quedara el nombre **Label1**, después de un clic en la etiqueta dibujada y podrá modificar el nombre de adentro y pondremos ahí **Nombre**. Si por error da doble clic en la etiqueta y lo manda a la pantalla de programación de la etiqueta, solo de doble clic en **UserForm1** que se encuentra en el **Explorador de Proyecto**.

5. Elija del **Cuadro de Herramientas** el control **Cuadro de Texto** el que tiene **ab** y arrastre dibujando en el formulario **USERFORM1** el cuadro de texto a un lado de la etiqueta que dice **Nombre**. El cuadro de texto debe de estar vacío y su nombre será **Textbox1**, el nombre solo aparecerá en el control.
6. Haga los dos pasos anteriores igualmente poniendo **Dirección** en la **Label2** y **Teléfono** en la **Label3** y también dibújales su **Textbox**. Esto quedara así después de haberlo hecho



*Si tiene algún problema* al dibujar las etiquetas o los cuadros de texto, solo cámbiele el nombre a la etiqueta o el cuadro de texto en la **Ventana Propiedades** la opción se llama (**Name**). El Error que marque puede ser **Nombre Ambiguo**, pero si le cambias el Nombre al control se quitara el error. Puedes ponerle cualquier nombre en lugar de Label1.

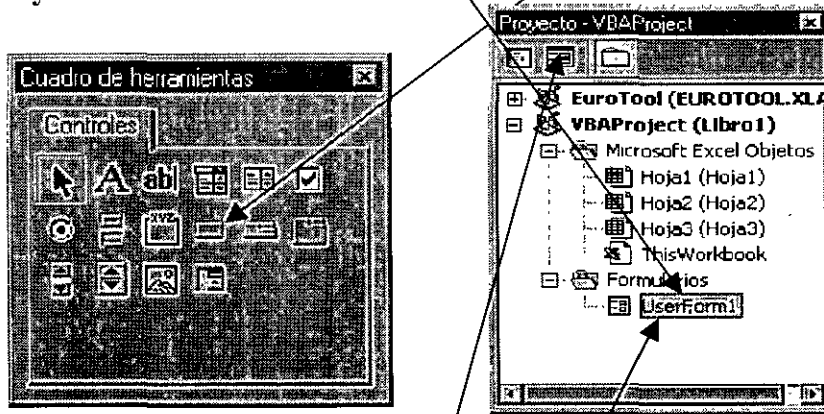


**Solo altera esto si te marca error, si NO déjalo así.**

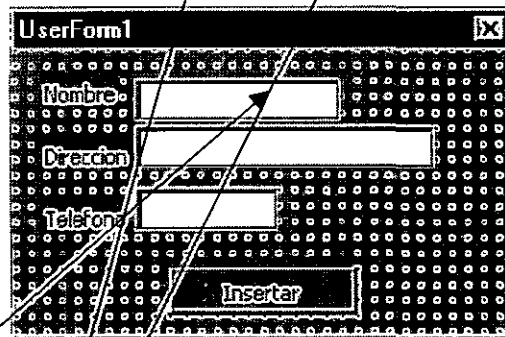
Los controles como las Etiquetas y Cuadros de Textos pueden modificárseles algunas opciones en la Ventana Propiedades Para hacer esto es necesario tener conocimiento sobre las propiedades de los controles. No altere las propiedades si no las conoce.



7. Elija del **Cuadro de Herramientas** el control **Botón de Comando** y Arrastre dibujando en el Formulario **USERFORM1** el Botón, después de un clic en el nombre del Botón dibujado y podrá modificar el nombre y pondremos ahí **Insertar**. Si por error da doble clic en la Botón y lo manda a la pantalla de programación de la etiqueta, solo de doble clic en **UserForm1** que se encuentra en el **Explorador de Proyecto**.



Así quedara el Formulario formado por los controles:



8. Ahora de doble clic sobre el control **Textbox1** para programarlo y después inserte el siguiente código:

```
Private Sub TextBox1_Change()  
Range("A9").Select  
ActiveCell.FormulaR1C1 = TextBox1  
End Sub
```

Esto indica que se valla a **A9** y escriba lo que hay en el **Textbox1**

**Nota.**-Lo que esta en azul lo genera Excel automáticamente, usted solo escribirá lo que esta en **Negrita**.

Para volver al **Formulario** y programar el siguiente Textbox de doble clic en **UserForm1** que se encuentra en el **Explorador de Proyecto**, o simplemente de clic en **Ver Objeto** en el mismo **Explorador de Proyecto**.

9. Ahora de doble clic sobre el control **Textbox2** para programarlo y después inserte el siguiente código:

```
Private Sub TextBox2_Change()  
Range("B9").Select  
ActiveCell.FormulaR1C1 = TextBox2  
End Sub
```

Esto indica que se vaya a **B9** y escriba lo que hay en el **Textbox2**

Para volver al **Formulario** y programar el siguiente Textbox de doble clic en **UserForm1** que se encuentra en el **Explorador de Proyecto**, o simplemente de clic en **Ver Objeto** en el mismo **Explorador de Proyecto**.

10. Ahora de doble clic sobre el control **Textbox3** para programarlo y después inserte el siguiente código:

```
Private Sub TextBox3_Change()  
Range("C9").Select  
ActiveCell.FormulaR1C1 = TextBox2  
End Sub
```

Esto indica que se vaya a **C9** y escriba lo que hay en el **Textbox3**

Para volver al **Formulario** y programar el **Botón de Comando Insertar** de doble clic en **UserForm1** que se encuentra en el **Explorador de Proyecto**, o simplemente de clic en **Ver Objeto** en el mismo **Explorador de Proyecto**.

11. Ahora de doble clic sobre el control **Botón de Comando** para programarlo y después inserte el siguiente código:

```
Private Sub CommandButton1_Click()  
Rem inserta un renglón  
Selection.EntireRow.Insert  
Rem Empty Limpia Los Textbox  
TextBox1 = Empty  
TextBox2 = Empty  
TextBox3 = Empty  
Rem TextBox1.SetFocus Envía el cursor al Textbox1 para volver a capturar los  
datos  
TextBox1.SetFocus  
End Sub
```

## MACROS EN EXCEL

**Nota.**-El comando **Rem** es empleado para poner comentarios dentro de la programación, el comando **Empty** es empleado para vaciar los Textbox.

12. Ahora presione el botón **Ejecutar User/Form** que se encuentra en la barra de herramientas o simplemente la tecla de función **F5**



Se activara el **Userform1** y todo lo que escriba en los Textbox se escribirá en Excel y cuando presione el botón Insertar, se insertara un renglón y se vaciaran los Textbox y después se mostrara el cursor en el **Textbox1**.