

Reporting con Macros

1.- Índice

- [2.- Introducción](#)
- [2.1.- Funcionalidades](#)
- [3.- Implementación](#)
- [3.1.- Formato](#)
- [3.1.1.- Hoja menu](#)
- [3.1.2.- Hoja definicion cuentas](#)
- [3.1.3.- Hoja definicion reports](#)
- [3.1.4.- Hoja saldos](#)
- [3.1.5.- Hoja resultado](#)
- [3.2.- Programación](#)
- [3.2.1.- Lectura de saldos](#)
- [3.2.2.- Ejecución de reports](#)
- [4.- Bibliografía](#)

2.- Introducción

Por lo general, las empresas disponen de una herramienta contable (léase contaplus, SAP R3,...) que les provee de las operaciones standard contables (apuntes contables, cuentas de mayor, acreedores...). Estas herramientas poseen la habilidad de generar reports pero se quedan 'cojos' en la manejabilidad de los mismos.

Al final, el departamento contable realiza descargas de los datos y manualmente realiza macros para suplir las carencias del reporting.

En este manual vamos a realizar una pequeña macro que resuelva las necesidades globales de reporting mensual de una pequeña/mediana empresa.

2.1.- Funcionalidades

Con esta solución no pretendemos dar cobertura a todas las necesidades de reporting de todas las empresas, si no ser una macro a partir de la cual y mediante ampliaciones, cada empresa pueda añadir sus necesidades específicas.

Funcionalidades:

- ✓ Lectura de los saldos por fichero, en este fichero se encontrará la información de la sociedad (solución multisociedad), cuenta contable, año, mes y saldo, dispondrá de los siguientes campos:

- 1.- Sociedad
- 2.- Año
- 3.- Mes
- 4.- Cuenta Contable
- 5.- Saldo
- 6.- Indicador de fin de fichero

Ejemplo de fichero:

```
PPSL2008020019000001000000000012,00X
PPSL20080200190000020000000004000,00X
PPSL200802001900000300000000500000,00X
PPSL2008020019000004-000000200000,00X
```

Layout del fichero:

Campo	Tamaño
Sociedad	4
Año	4
Mes	2
Cuenta Contable	10
Saldo	16
Indicador de fin	1

- ✓ Ejecución de los reports indicando:

- 1.- Sociedad
- 2.- Año
- 3.- Mes
- 4.- Tipo report (solución con varios reports).

3.- Implementación

Para realizar la implementación dividiremos las tareas en 2:

- ✓ Formato: esta tarea engloba crear todas las hojas necesarias para la solución, incluyendo los textos y los botones
- ✓ Programación: desarrollar las macros para ejecutar las funcionalidades que requieran programación, como:

- 1.- Lectura de saldos
- 2.- Ejecución de reports

3.1.- Formato

Inicialmente creamos 5 hojas:

menu: desde donde se ejecutarán las funcionalidades

definicion_cuentas: donde estarán definidas las cuentas contables y a que epígrafes de los diferentes reports se mapean.

definicion_reports: donde se indican el orden de los epígrafes y los literales de los diferentes reports

saldos: donde se graban los saldos leídos por fichero.

resultado: donde se muestra el report.

3.1.1.- Hoja menu

En esta hoja se definen los siguientes nombres (Insertar>Nombre>Definir):

- ✓ **fichero:** celda B3, donde se debe indicar el nombre del fichero que contiene los saldos.
- ✓ **sociedad:** celda B9, donde el usuario debe indicar la sociedad.
- ✓ **anno:** celda B10, donde se indica el año de los datos del report.
- ✓ **mes:** celda B11, donde se indica el mes del report.
- ✓ **report:** celda B12, donde se indica el report a ejecutar.



3.1.2.- Hoja definicion_cuentas

En la primera columna se indicara las cuentas, y en las columnas siguientes se indicaran los epigrafes de cada report
Es muy importante que en la fila 2 se marque el nombre del report ya que nos servirá para identificarlos y ejecutarlo.



En la primera columna está la cuenta.

No todas las cuentas deben aparecer en todos los reports

3.1.3.- Hoja definicion_reports



Los nombres de la fila 2 deben ser iguales a los de la fila 2 de la hoja de definicion_cuentas, ya que son los nombres de los reports.

3.1.4.- Hoja saldos

En esta hoja estará la información de las cuentas con sus dimensiones (léase sociedad, año, mes) y el saldo.



3.1.5.- Hoja resultado

El resultado será el siguiente:



3.2.- Programación

Antes de empezar a programar se aconseja crear 2 módulos de programación:

El primer modulo (MCONSTANTES) dispone de constantes y de definición de tipos. Se definen las siguientes constantes, con el fin de hacer más legible el código:

Definición de hojas

```
Global Const CONST_HOJA_MENU = "menu"  
Global Const CONST_HOJA_DEF_CUENTAS = "definicion_cuentas"  
Global Const CONST_HOJA_DEF_REPORTS = "definicion_reports"  
Global Const CONST_HOJA_SALDOS = "saldos"  
Global Const CONST_HOJA_RESULTADO = "resultado"
```

Adicionalmente se crean las constantes de error que seguro utilizaremos:

Definición de errores

```
Global Const CONST_ERROR_FALTA_FICHERO = 1  
Global Const CONST_ERROR_FALTA_SOCIEDAD = 2  
Global Const CONST_ERROR_FALTA_ANNO = 3  
Global Const CONST_ERROR_FALTA_MES = 4  
Global Const CONST_ERROR_FALTA_REPORT = 5  
Global Const CONST_ERROR_FICHERO_NOEXISTE = 6  
Global Const CONST_ERROR_REPORT_NOEXISTE = 7
```

Las constantes de mensajes:

Definición de mensajes

```
Global Const CONST_MENSAJE_FIN_LECTURA = 1  
Global Const CONST_MENSAJE_FIN_REPORT = 2
```

Se indican las constantes de filas:

Definición de filas y columnas

```
Global Const CONST_FILA_INICIO_SALDO = 3  
Global Const CONST_FILA_INICIO_DEF_CUENTAS = 3  
Global Const CONST_FILA_INICIO_DEF_REPORTS = 3
```

Y por último se define un tipo que nos servirá para leer el fichero:

Definición del fichero

```
Public Type TFICHERO  
    sociedad As String * 4  
    anno As String * 4  
    mes As String * 2  
    ctas As String * 10  
    saldo As String * 16  
    fin As String * 1  
    intro As String * 2  
End Type
```

Fíjate que se ha creado un campo que se llama intro que ocupa 2 posiciones; equivale al intro de cambio de línea del fichero.

Una vez definido un Tipo se puede utilizar en la definición de variables de la siguiente manera:

```
Dim nombre_variable as tipo_creado
```

El segundo módulo (MMOSTRAR) dispone de las funciones que muestran información por pantalla, como puede ser mensajes de error o mensajes de fin de proceso:

```
1. Muestra mensaje de error  
2. Public Sub mostrar_error(codigo As Long, Optional demas As Variant)  
3. Dim mensaje As String  
4.  
5. Select Case codigo  
6. Case CONST_ERROR_FALTA_FICHERO  
7.     mensaje = "Para leer el fichero debe indicar el nombre"  
8. Case CONST_ERROR_FALTA_SOCIEDAD  
9.     mensaje = "Para ejecutar el report debes indicar la sociedad"  
10. Case CONST_ERROR_FALTA_ANNO  
11.     mensaje = "Para ejecutar el report debes indicar el año"  
12. Case CONST_ERROR_FALTA_MES  
13.     mensaje = "Para ejecutar el report debes indicar el mes"  
14. Case CONST_ERROR_FALTA_REPORT  
15.     mensaje = "Para ejecutar el report debes indicar el tipo de report"  
16. Case CONST_ERROR_FICHERO_NOEXISTE  
17.     mensaje = "El fichero " & CStr(demas) & " no existe"  
18.     Case CONST_ERROR_REPORT_NOEXISTE19.     mensaje = "El tipo de report " &  
CStr(demas) & " no existe"  
20. End Select  
21. MsgBox mensaje, vbCritical, "Mensaje de error!!"
```

22.End Sub

```
1. 'Muestra el mensaje de finalizacion de proceso
2. Public Sub mostrar_mensaje(codigo As Long, Optional demas As Variant)
3. Dim mensaje As String
4.
5. Select Case codigo
6. Case CONST_MENSAJE_FIN_LECTURA
7. mensaje = "Se realizo la lectura del fichero"
8. Case CONST_MENSAJE_FIN_REPORT
9. mensaje = "Se realizo la ejecución del report"
10. End Select
11. MsgBox mensaje, vbInformation, "Mensaje de fin proceso"
12.End Sub
```

3.2.1.- Lectura de saldos

En el menú se han creado dos botones asociado a la lectura de saldos:

- ✓ **Leer saldos borrando saldos antiguos:** borra toda la información que hay en la pestaña saldos y procede a la lectura del fichero.
- ✓ **Leer saldos:** sólo lee el fichero de saldos y los acumula a los ya existentes en la pestaña de saldos.

El código asociado a estos dos botones es el siguiente:

```
1. 'Leer fichero sin borrar datos actuales
2. Private Sub CommandButton1_Click()
3. Call leer_fichero(False)
4. End Sub

5. 'Leer fichero borrando datos actuales
6. Private Sub CommandButton2_Click()
7. Call leer_fichero(True)
8. End Sub
```

3.2.1.1.- Código principal

El procedimiento leer_fichero tiene un parámetro que le indica si se deben borrar los datos actuales de la hoja de saldos o no.

El algoritmo general de lectura es el siguiente:

```
si validacion_fichero() entonces
  si borrar_saldos entonces
    borrar_informacion_saldos()
    fila=inicio_fila()
  si no
    fila=buscar_ultima_fila()
  finsi
  fd=abrir_fichero()
  mientras no EOF(fd)
  leer_fichero()
  escribir_informacion(fila)
  finmientras
  cerrar_fichero(fd)
  mostrar_mensaje_fin_lectura()
finsi
```

El código es el siguiente:

```
1. Public Sub leer_fichero(borrar_info As Boolean)
2. Dim fila As Long, fd As Long
3. Dim informacion As TFICHERO
4.
5. If validar_fichero Then
6. If borrar_info Then
7. Call borrar_informacion_saldos
8. fila = CONST_FILA_INICIO_SALDO
9. Else
10. fila = buscar_ultima_fila
11. End If
12. fd = abrir_fichero()
13.
14. While EOF(fd) = False
15. Get #fd, informacion
16. Sheets(CONST_HOJA_SALDOS).Cells(fila, 1) = informacion.sociedad
17. Sheets(CONST_HOJA_SALDOS).Cells(fila, 2) = informacion.anno
18. Sheets(CONST_HOJA_SALDOS).Cells(fila, 3) = informacion.mes
19. Sheets(CONST_HOJA_SALDOS).Cells(fila, 4) = informacion.ctas
20. Sheets(CONST_HOJA_SALDOS).Cells(fila, 5) = CDb1(informacion.saldo)
21. fila = fila + 1
22. Wend
23. Call cerrar_fichero (fd)
24. Call mostrar_mensaje(CONST_MENSAJE_FIN_LECTURA)
25. End If
26.End Sub
```

En la línea 14 se ve el uso de la función EOF (End Of File), que nos indica si hemos acabado de leer el fichero.

La función Get sirve para leer un fichero abierto en modo lectura y random.

En la variable fila indicamos en qué fila se va a ir escribiendo la información leída en el fichero, por eso en la fila 21 se incrementa en uno una vez insertada la información.

3.2.1.2.- Función validar fichero

El código de la validación del fichero es:

```
1. Public Function validar_fichero() As Boolean
2. Dim ret As Boolean
3.
4. ret = True
5. If Range("fichero") = "" Then 'si no se ha escrito nada
6. Call mostrar_error(CONST_ERROR_FALTA_FICHERO)
7. ret = False
8. Else
9. If Dir(Range("fichero"), vbArchive) = "" Then 'si el fichero no existe
10. Call mostrar_error(CONST_ERROR_FICHERO_NOEXISTE)
11. ret = False
12. End If
13. End If
14. validar_fichero = ret
15. End Function
```

Esta función retorna True si no hay ningún error; las validaciones que realiza son:

- ✓ Que el nombre del fichero esté informado
- ✓ Que el fichero exista

Para acceder a una celda definida con nombres se utiliza la función Range, que equivaldría a poner cells (fila, columna), donde fila y columna indica que celda se desea leer.

La función Dir se comporta como el DIR de MS-DOS, de tal manera que retorna una cadena de caracteres diferente de "" si ha encontrado un fichero.

Fijate que para retornar un valor en una función se realiza la asignación del tipo:

[nombre de funcion] = valor

3.2.1.3.- Procedimiento borrar_informacion_saldos

```
1.Public Sub borrar_informacion_saldos()
2. Sheets(CONST_HOJA_SALDOS).Select
3. Rows(CONST_FILA_INICIO_SALDO & ":65535").Select
4. Selection.ClearContents
5. Range("A3").Select
6. Sheets(CONST_HOJA_MENU).Select
7.End Sub
```

Este procedimiento borra la información que hay en la pestaña de saldos. Para poder borrar la información se debe cambiar de hoja por ello la línea 2.

Es muy importante que, si cambiamos de pestaña, volvamos a la pestaña original con el fin de no despistar al usuario.

3.2.1.4.- Función buscar_ultima_fila

```
1. Public Function buscar_ultima_fila() As Long
2. Dim ret As Long
3.
4. ret = CONST_FILA_INICIO_SALDO
5. While Sheets(CONST_HOJA_SALDOS).Cells(ret, 1) <> ""
6. ret = ret + 1
7. Wend
8. buscar_ultima_fila = ret
9. End Function
```

Esta función busca en qué fila no hay nada escrito en la hoja de saldos, sólo comprueba la primera columna

3.2.1.5.- Función abrir_fichero

```
1.Public Function abrir_fichero() As Long
2.Dim fd As Long, eo1 As TFICHERO
3.
4. fd = FreeFile
5. Open Range("fichero") For Random As #fd Len = Len(eo1)
6. abrir_fichero = fd
7.End Function
```

Abre un fichero en modo lectura y retorna el identificador que sirve para después acceder al fichero.

Existen varias maneras de abrir ficheros, en este caso hemos elegido en modo lectura y con random.

Es común que en la programación reciba el nombre de fd al fichero, referenciándose a "field descriptor".

Para más información de como abrir ficheros accede al help de Excel de la función Open y encontrarás todos los detalles.

3.2.1.6.- Procedimiento cerrar_fichero

```
1.Public Sub cerrar_fichero(fd As Long)
2. Close #fd
3.End Sub
```

3.2.2.- Ejecución de reports

En este punto explicaré como realizar el reporting, algoritmo que se puede hacer de muchas maneras. Intentaremos hacerla de manera sencilla y, como no, por pasos:

Paso 1: validación de los datos de entrada

Paso 2: copiar TODO el contenido de los saldos a la pestaña de resultado

Paso 3: filtrar los datos que hay en la pestaña de resultado

Paso 4: filtrar información que no se mostrará en el report elegido

Paso 5: Asignar con la información restante a cada epígrafe el saldo

Paso 6: poner el report en bonito ajustando el formateo

Veamos un ejemplo de como va evolucionando la hoja de resultados por pasos:

Paso	Estado
Paso 2, es posible que haya información en la hoja de resultado de otras ejecuciones; por eso esta hoja no está vacía	
Paso 2, copia de la información a partir de la fila 3	
Paso 2, borrar las 2 primeras filas de otras ejecuciones	
Paso 3, se quitan todos los registros que no cumplen la selección (sociedad, año y mes)	
Paso 4, Poner la formula de buscarv asociado a la hoja de cuentas y de definicion del report	
Paso 4, se copia por valor y se elimina todo aquello que no sirva para el report	
Paso 5, ordenar la información por la columna G	
Paso 5, borrar las 2 primeras columnas	
Paso 5, copiar la estructura del report en la primera columna	
Paso 5, asignar a los epígrafes de la columna A sus importes	
Paso 6, borrar las columnas C a la G y ajustar anchos y altos de celda	
Paso 6, ajustar formato de la columna B, con 2 decimales y el punto separador de miles	

El código principal queda así:

```
1. Public Sub ejecutar_report()
2. Dim columna_cuentas As Long, columna_reports As Long
3.
4. If validar_ejecutar(columna_cuentas, columna_reports) Then
5. Call copiar_saldos
6. Call filtrar_seleccion
7. Call filtrar_epigrafes(columna_cuentas, columna_reports)
8. Call asignar_importes(columna_reports)
9. Call formateo_final
10. Call mostrar_mensaje(CONST_MENSAJE_FIN_REPORT)
11. End If
```

12.End Sub

Veamos el detalle de cada paso.

3.2.2.1.- Función validar_ejecutar (paso 1)

```
1. 'Validacion de que se puede realizar la ejecucion del report
2. Public Function validar_ejecutar(columna_cuentas As Long,
3.   columna_reports As Long) As Boolean
4. Dim ret As Boolean
5.
6. ret = True
7. If Range("sociedad") = "" Then
8.   ret = False
9.   Call mostrar_error(CONST_ERROR_FALTA_SOCIEDAD)
10. Else
11.   If Range("anno") = "" Then
12.     ret = False
13.     Call mostrar_error(CONST_ERROR_FALTA_ANNO)
14.   Else
15.     If Range("mes") = "" Then
16.       ret = False
17.       Call mostrar_error(CONST_ERROR_FALTA_MES)
18.     Else
19.       If Range("report") = "" Then
20.         ret = False
21.         Call mostrar_error(CONST_ERROR_FALTA_REPORT)
22.       Else
23.         Call existe_report(columna_cuentas, columna_reports)
24.         If columna_cuentas = -1 Or columna_reports = -1 Then
25.           ret = False
26.           Call mostrar_error(CONST_ERROR_REPORT_NOEXISTE)
27.         End If
28.       End If
29.     End If
30.   End If
31. End If
32. validar_ejecutar = ret
33.End Function
```

Primero se valida que la sociedad, año, mes y report se han escrito, en el siguiente punto se comprueba que el report está definido en las hojas de definición de cuentas y de reports utilizando el procedimiento existe_report.

El procedimiento existe_report nos retorna en qué columna está definido el report en cada una de las 2 hojas antes mencionadas (definición de cuentas y definición de reports). Si retorna -1 quiere decir que no lo ha encontrado. Estas columnas se utilizarán en otros pasos.

3.2.2.2.- Procedimiento copiar_saldos (paso 2)

```
1. 'Realizar la copia del contenido de la tabla de saldos a la hoja de resultado
2. Public Sub copiar_saldos()
3.
4. Sheets(CONST_HOJA_SALDOS).Select
5. Range(CONST_FILA_INICIO_SALDO & ":" & "65535").Select
6. Selection.Copy
7. Sheets(CONST_HOJA_RESULTADO).Select
8. Rows(CONST_FILA_INICIO_SALDO & ":" & CONST_FILA_INICIO_SALDO).Select
9. ActiveSheet.Paste
10.
11. Range("1:2").Select 'borrar las 2 primeras filas
12. Range("A2").Activate
13. Selection.ClearContents
14.
15. Sheets(CONST_HOJA_MENU).Select
16.End Sub
```

De la línea 4 a la 9 se realiza el copiado de la información de la hoja de saldos a la hoja de resultados; las siguientes líneas borran el contenido de las 2 primeras filas.

3.2.2.3.- Procedimiento filtrar_seleccion (paso 3)

```
1. 'Eliminar todas los datos que no concuerden con sociedad, año o mes
2. Public Sub filtrar_seleccion()
3. Dim sociedad As String, anno As Long, mes As Long
4. Dim fila As Long
5.
6. sociedad = Range("sociedad")
7. anno = Range("anno")
8. mes = Range("mes")
9. Sheets(CONST_HOJA_RESULTADO).Select
10. fila = CONST_FILA_INICIO_SALDO
11. While Cells(fila, 1) <> ""
12.   If Cells(fila, 1) <> sociedad Or Cells(fila, 2) <> anno Or Cells(fila, 3) <> mes Then
13.     Rows(fila & ":" & fila).Select
14.     Selection.Delete Shift:=xlUp
15.   Else
16.     fila = fila + 1
17.   End If
18. Wend
19. Sheets(CONST_HOJA_MENU).Select
20.End Sub
```

En el bucle (línea 11 a la 18) recorre todos los registros y si no son iguales a la sociedad/año/mes seleccionado los borra con el siguiente código:

```
13. Rows(fila & ":" & fila).Select
14. Selection.Delete Shift:=xlUp
```

3.2.2.4.- Procedimiento filtrar_epigrafes (paso 4)

```
1. 'Se ponen los buscarv para realizar un borrado de lo que realmente no es necesario
2. Public Sub filtrar_epigrafes(columna_cuentas As Long, columna_reports As Long)
3. Dim fila As Long, auxi As Long
4.
5. auxi = -7 + columna_reports
6. fila = CONST_FILA_INICIO_SALDO
```

```

7. Sheets(CONST_HOJA_RESULTADO).Select
8. While Sheets(CONST_HOJA_RESULTADO).Cells(fila, 1) <> ""
9. 'se pone la formula de buscarv sobre la hoja de def. cuentas y de def. reports
10. Cells(fila, 6).Formula = "=VLOOKUP(RC[-2], " & CONST_HOJA_DEF_CUENTAS & "[C[-5]:C[20], " & columna_cuentas & ",)"
11. Cells(fila, 7).Formula = "=VLOOKUP(RC[-1], " & CONST_HOJA_DEF_REPORTS & "[C[" & auxi & "],1,)"
12. fila = fila + 1
13. Wend
14. 'copiar por valor
15. Cells.Select
16. Selection.Copy
17. Selection.PasteSpecial Paste:=xlValues, Operation:=xlNone, SkipBlanks:= _
18. False, Transpose:=False
19.
20. fila = CONST_FILA_INICIO_SALDO
21. 'borrar lo sobrante
22. While Sheets(CONST_HOJA_RESULTADO).Cells(fila, 1) <> ""
23. If IsError(Sheets(CONST_HOJA_RESULTADO).Cells(fila, 6)) Or _
24. IsError(Sheets(CONST_HOJA_RESULTADO).Cells(fila, 7)) Then
25. Rows(fila & ":" & fila).Select
26. Selection.Delete Shift:=xlUp
27. Else
28. If Sheets(CONST_HOJA_RESULTADO).Cells(fila, 6) = 0 Or _
29. Sheets(CONST_HOJA_RESULTADO).Cells(fila, 7) = 0 Then
30. Rows(fila & ":" & fila).Select
31. Selection.Delete Shift:=xlUp
32. Else
33. fila = fila + 1
34. End If
35. End If
36. Wend
37. Sheets(CONST_HOJA_MENU).Select
38.End Sub

```

Este procedimiento es el más complejo, así que separémoslos en operatividad:

Fila 5 a 13: se recorre la información y para cada registro pone:

En la columna 6 la función BUSCARV para confirmar que esa cuenta está definida en la hoja de definición de cuentas.

En la columna 7 la función BUSCARV para confirmar que el epígrafe definido en la columna 6 está en la definición del report escogido.

Fila 14 a 19: copiar por valor la información que hay en la hoja

Fila 20 a 36: eliminar los registros en que la columna 6 ó 7 ha retornado #N/A (utilizando la función IsError) ó 0.

3.2.2.5- Procedimiento asignar_importes (paso 5)

```

1. 'Realizar la asignacion de los importes a los epigrafes del report
2. Public Sub asignar_importes(columna_reports)
3. Dim fila As Long, resultado As Double, fila_busqueda As Long
4. Dim fila_report As Long
5.
6. 'ordenar la hoja de resultados para que los espacios
7. 'en blanco esten por debajo
8. Sheets(CONST_HOJA_RESULTADO).Select
9. Cells.Select
10. Selection.Sort Key1:=Range("G1"), Order1:=xlAscending, Header:=xlGuess, _
11. OrderCustom:=1, MatchCase:=False, Orientation:=xlTopToBottom
12.
13. 'copiar estructura del report
14. Range("A1:B65535").Select 'borrar la primera y la segunda columna
15. Range("A2").Activate
16. Selection.ClearContents
17.
18. fila = CONST_FILA_INICIO_DEF_REPORTS
19. fila_report = 1
20. While Sheets(CONST_HOJA_DEF_REPORTS).Cells(fila, columna_reports) <> ""
21. Cells(fila_report, 1) = Sheets(CONST_HOJA_DEF_REPORTS).Cells(fila, _
22. columna_reports)
23. fila_report = fila_report + 1
24. fila = fila + 1
25. Wend
26. 'asignar los importes
27. fila = 1
28. While Cells(fila, 1) <> ""
29. resultado = 0
30. fila_busqueda = 1
31. 'acumular el importe que concuerda el epigrafe
32. While Cells(fila_busqueda, 7) <> ""
33. If Cells(fila_busqueda, 7) = Cells(fila, 1) Then
34. resultado = resultado + Cells(fila_busqueda, 5)
35. End If
36. fila_busqueda = fila_busqueda + 1
37. Wend
38. Cells(fila, 2) = resultado
39. fila = fila + 1
40. Wend
41.End Sub

```

Para asignar los epígrafes resultantes a los registros que nos quedan en esta hoja se deben realizar los siguientes pasos:

Fila 6 a 11: ordenar por la columna 7

Fila 13 a 25: borrar las 2 primeras columnas de la hoja de resultados y copiar la estructura del report de la hoja de definición de reports en la columna 1 de la hoja de resultado

Fila 26 a 40: para cada registro que hay en la columna 1 buscar el importe con el mismo epígrafe (comparar primera columna con la séptima)

3.2.2.6.- Procedimiento formateo_final (paso 6)

```

1. 'Poner en bonito el resultado
2. Public Sub formateo_final()
3. 'borrar informacion innecesaria
4. Sheets(CONST_HOJA_RESULTADO).Select
5. Columns("C:G").Select
6. Selection.ClearContents
7. 'ajustar el alto y el ancho de las 2 primeras columnas
8. Cells.Select
9. Cells.EntireColumn.AutoFit
10. Cells.EntireRow.AutoFit
11.
12. 'asignar formato al numero con 2 decimales
13. Columns("B:B").Select
14. With Selection
15. .HorizontalAlignment = xlRight

```

```
16. .VerticalAlignment = xlBottom
17. .WrapText = False
18. .Orientation = 0
19. .AddIndent = False
20. .ShrinkToFit = False
21. .MergeCells = False
22. End With
23. Selection.NumberFormat = "##,##0.00"
24. Range("A1").Select
25. Sheets(CONST_HOJA_MENU).Select
26.End Sub
```

Fila 4 a la 7: borra el contenido de las columnas C a G.

Fila 8 a la 11: define el ancho y alto óptimo de todas las celdas

Fila 12 a la 25: define el formato de la segunda columna, el formato es numérico con 2 decimales y el punto separador de miles.

4.- Bibliografía

<http://www.asesores.com/contabilidad/conta1.htm>