

	A	B	C
1	Año	Area	Valor
2	Ultimo año	Direccion Administrativa	26,8
3	Ultimo año	Financiero	28,2
4	Ultimo año	Sistemas	32,1
5	Ultimo año	Recursos humanos	85,6
6	Ultimo año	Mantenimiento	33,8
7	Ultimo año	Juridico	23,8
8	Ultimo año	Planeación	38
9	Ultimo año	Auditoría	99,6
10	Ultimo año	Revisoría Fiscal	13
11	Año anterior	Direccion Administrativa	61
12	Año anterior	Financiero	43,9
13	Año anterior	Sistemas	62,4
14	Año anterior	Recursos humanos	48,1
15	Año anterior	Mantenimiento	59,7
16	Año anterior	Juridico	76,7
17	Año anterior	Planeación	48,9
18	Año anterior	Auditoría	2,6
19	Año anterior	Revisoría Fiscal	6,6

Podemos llegar con ELEMENTOS calculados a esto:

Suma de Valor		Año		
grupo	Area	Año anterior	Ultimo año	Total general
Diferencia	Auditoría	-2,6	99,6	97
	Direccion Administrativa	-61	26,8	-34,2
	Financiero	-43,9	28,2	-15,7
	Juridico	-76,7	23,8	-52,9
	Mantenimiento	-59,7	33,8	-25,9
	Planeación	-48,9	38	-10,9
	Recursos humanos	-48,1	85,6	37,5
	Revisoría Fiscal	-6,6	13	6,4
	Sistemas	-62,4	32,1	-30,3
Total Diferencia		-409,9	380,9	-29
Total general		-409,9	380,9	-29

Si nos pasan la info así:

	A	B	C
1	Area	Ultimo año	Año anterior
2	Direccion Administrativa	26,8	61
3	Financiero	28,2	43,9
4	Sistemas	32,1	62,4
5	Recursos humanos	85,6	48,1
6	Mantenimiento	33,8	59,7
7	Juridico	23,8	76,7
8	Planeación	38	48,9
9	Auditoría	99,6	2,6
10	Revisoría Fiscal	13	6,6

Con CAMPOS CALCULADOS Podemos llegar a esto:

Rótulos de fila	Suma de Ultimo año	Suma de Año anterior	Suma de Variacion
Auditoría	99,6	2,6	97
Direccion Administrativa	26,8	61	-34,2
Financiero	28,2	43,9	-15,7
Juridico	23,8	76,7	-52,9
Mantenimiento	33,8	59,7	-25,9
Planeación	38	48,9	-10,9
Recursos humanos	85,6	48,1	37,5
Revisoría Fiscal	13	6,6	6,4
Sistemas	32,1	62,4	-30,3
Total general	380,9	409,9	-29

Más de tablas dinámicas:

Cruces de información.

Auditoría o Contabilidad, Costos, Producción o cualquier profesional que necesite comparar dos bases, siempre tendrá necesidad de cruzar información.

Los elementos y campos calculados nos pueden ayudar con esta tarea, sin embargo, es mi deber comentarle que es una gran ayuda tener un nuevo Excel con capacidad para manejar más de un millón de registros pero, cuando usted monta elementos o campos calculados en un par de bases que le midan más de 500 mil registros, el Excel de su máquina, se le puede quedar literalmente MUERTO. Esto es debido al enorme esfuerzo que comienza a efectuar el aplicativo haciendo la operación campo por campo. Incluso con un Computador con las mejores especificaciones pasa.

Qué hacer entonces?

Pues ayudarle a Excel evitando que se esfuerce tanto. Cómo? NO utilizando ni campos ni elementos calculados. Lo que Excel si hace MUY RAPIDAMENTE, es la Suma total, por filas y por columnas. Entonces, el truco radica en cambiar el signo al campo valor en una de las bases o una de las tablas.

Supongamos que tenemos un Extracto bancario, que lo hemos podido descargar de internet, gracias a los servicios del banco. Este puede ser un ejemplo:

	A	B	C	D
1	Ref	fecha	Ref	valor
2	Transferencia	16/11/2009	sdfdsfds	\$ 801.680
3	Transferencia	16/11/2009	nfgnbds	\$ 346.117
4	Transferencia	16/11/2009	dfgdfggb	\$ 885.762
5	Cheque pagado	16/11/2009	yuiyyujf	\$ 198.936
6	Cheque pagado	16/11/2009	ldffasqw	\$ 936.976

Para el caso, sirve. (Es un ejemplo inventado por mí, el propósito es transmitirle la idea...).

Ahora, en otra hoja de cálculo Usted tiene el libro auxiliar en archivo plano, y ya lo ha logrado depurar, esto es Limpiarle la basura de los encabezados, o a lo mejor su aplicativo contable exporta los datos de una sola vez a Excel, y digamos que tiene la info así:

	A	B	C	D
1	Consignación	16/11/2009	Consignación del día	\$ 801.980
2	Consignación	16/11/2009	Consignación del día	\$ 346.117
3	Consignación	16/11/2009	Consignación del día	\$ 885.792
4	Pagos	16/11/2009	Paguitos del día...	\$ 198.936
5	Pagos	16/11/2009	Paguitos del día...	\$ 939.679

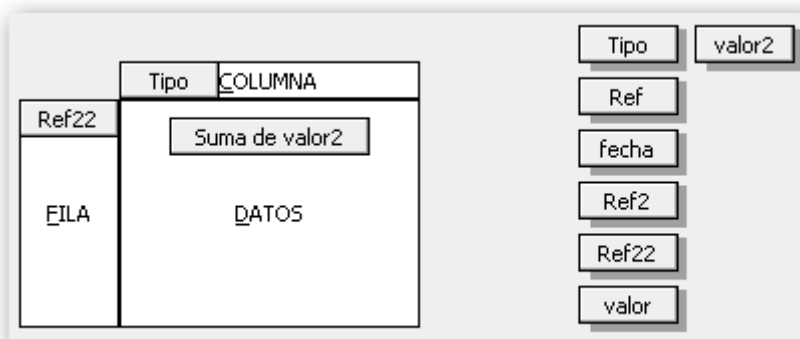
Para cruzar la información, lo que le sugiero es:

1. Unifique las dos bases de datos, una debajo de la otra.
2. Identifíquelas. Esto es, coloque un nombre en la columna A, a cada una de las tablas, cosa que le permita a Excel distinguir que la tabla de arriba es el Extracto y la tabla de abajo es la contabilidad.
3. Si puede identificar más cosas en común, inserte una nueva columna y coloque el mismo nombre. Por ejemplo, por decir algo, las transferencias recibidas pueden ser equivalentes a las consignaciones. Entonces, a esos dos conceptos, transferencias y consignaciones, llámelos de la misma manera.
4. Cámbiele el signo a una de las tablas, en una nueva columna de Valor.

	A	B	C	D	E	F	G
1	Tipo	Ref	fecha	Detalle	Ref2	valor	valor2
2	Extracto Banco	Transferencia	16/11/2009	sdfdsfds	Abonos	\$ 801.680	\$ -801.680
3	Extracto Banco	Transferencia	16/11/2009	nfgnbds	Abonos	\$ 346.117	\$ -346.117
4	Extracto Banco	Transferencia	16/11/2009	dfgdfggb	Abonos	\$ 885.762	\$ -885.762
5	Extracto Banco	Cheque pagado	16/11/2009	yuiyyujf	Pagos	\$ 198.936	\$ -198.936
6	Extracto Banco	Cheque pagado	16/11/2009	ldffasqw	Pagos	\$ 936.976	\$ -936.976
7	Auxiliar	Consignación	16/11/2009	Consignación del día	Abonos	\$ 801.980	\$ 801.980
8	Auxiliar	Consignación	16/11/2009	Consignación del día	Abonos	\$ 346.117	\$ 346.117
9	Auxiliar	Consignación	16/11/2009	Consignación del día	Abonos	\$ 885.792	\$ 885.792
10	Auxiliar	Pagos	16/11/2009	Paguitos del día...	Pagos	\$ 198.936	\$ 198.936
11	Auxiliar	Pagos	16/11/2009	Paguitos del día...	Pagos	\$ 939.679	\$ 939.679

Mire que aquí, en la columna A, e identificado que el bloque superior es el extracto bancario y el inferior el libro auxiliar de contabilidad. También, en la columna REF2, unifiqué el criterio de que lo que es similar en ambas tablas. Finalmente, dupliqué la columna de Valor en Valor2, y le cambié el signo a la tabla de arriba.

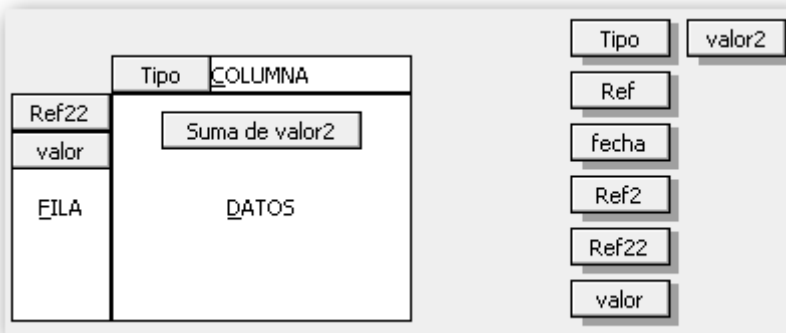
Con la información organizada de esta manera, Excel no tiene que calcular nada, salvo hacer la operación de suma. La idea de esto es, que si existen 100 pesos en una de las tablas, y esos 100 pesos existen en la otra tabla, pero negativos (porque le cambiamos el signo), Excel sumará $+100 -100 = 0$. Entonces, TOODO lo que nos dé CERO, significará que está cuadrado. (Ojo, En principio, pues te puede ocurrir que hayan varios valores de 100!!). Organice la tabla dinámica así:



La tabla dinámica le quedará así:

	A	B	C	D	E
1					
2					
3	Suma de valor2	Tipo			
4	Ref22	Auxiliar	Extracto Banco	Total general	
5	Abonos	2033889,06	-2033559,06	330	
6	Pagos	1138615,31	-1135912,31	2703	
7	Total general	3172504,37	-3169471,37	3033	
8					

Como puede ver, aquí está la conciliación.... Hay unos casos que le llegan a uno en los cuales a veces es muy difícil hacer un match (match = coincidencia), y no hay una llave (campo clave) sobre el cual efectuar la conciliación. Entonces, una salida, puede ser, CRUZAR EL MISMO VALOR. Como los valores son positivos en ambas BASES, pues vamos a rastrear o cruzar a nivel del mismo valor, ASI:



La tabla dinámica nos mostrará esto:

	A	B	C	D	E
1					
2					
3	Suma de valor2		Tipo		
4	Ref22	valor	Auxiliar	Extracto Banco	Total general
5	Abonos	\$ 346.117	346117,15	-346117,15	0
6		\$ 801.680		-801680,37	-801680,37
7		\$ 801.980	801980,37		801980,37
8		\$ 885.762		-885761,54	-885761,54
9		\$ 885.792	885791,54		885791,54
10	Total Abonos		2033889,06	-2033559,06	330
11	Pagos	\$ 198.936	198935,87	-198935,87	0
12		\$ 936.976		-936976,44	-936976,44
13		\$ 939.679	939679,44		939679,44
14	Total Pagos		1138615,31	-1135912,31	2703
15	Total general		3172504,37	-3169471,37	3033

Ahora, podemos cruzar por fechas a ver qué pasa:

Obtenemos esto:

Suma de valor2			Tipo		
fecha	Ref22	valor	Auxiliar	Extracto Banco	Total general
16/11/2009	Abonos	\$ 346.117	346117,15	-346117,15	0
		\$ 801.680		-801680,37	-801680,37
		\$ 801.980	801980,37		801980,37
		\$ 885.762		-885761,54	-885761,54
		\$ 885.792	885791,54		885791,54
	Total Abonos		2033889,06	-2033559,06	330
	Pagos	\$ 198.936	198935,87	-198935,87	0
		\$ 936.976		-936976,44	-936976,44
		\$ 939.679	939679,44		939679,44
	Total Pagos		1138615,31	-1135912,31	2703
Total 16/11/2009		3172504,37	-3169471,37	3033	
Total general		3172504,37	-3169471,37	3033	

Podemos ver que en los abonos y en los pagos hay dos valores que cuadran. Y que la diferencia total de 3.033 viene de 330 pesos en abonos y 2703 en pagos, y cada uno de ellos, son de una inconsistencia en valores, por ejemplo, el auxiliar dice tener 801.980,37 y el extracto dice que tiene 801.680. Un error de digitar un 980 en lugar de un 680, además de otros dos casos.

Cruzar la información de esta manera es sumamente VELOZ para Excel, mientras que si lo pones a hacer campos o elementos calculados, tu computadora se esfuerza más.

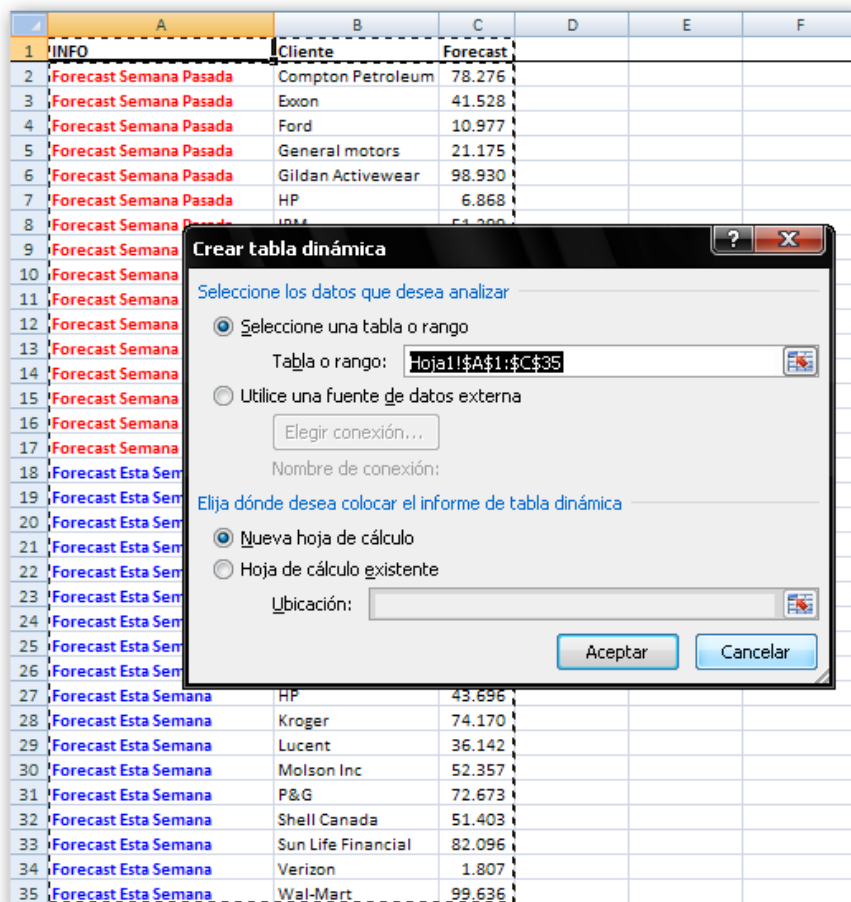
Más de Tablas dinámicas: Desactivar la opción GETPIVOTDATA.

Estoy seguro, que habrá notado que cuando se desean hacer cálculos u operaciones sobre los datos de la tabla dinámica, Excel crea una fórmula RARISIMA, incluso para una SIMPLE DIFERENCIA DE CELDAS.

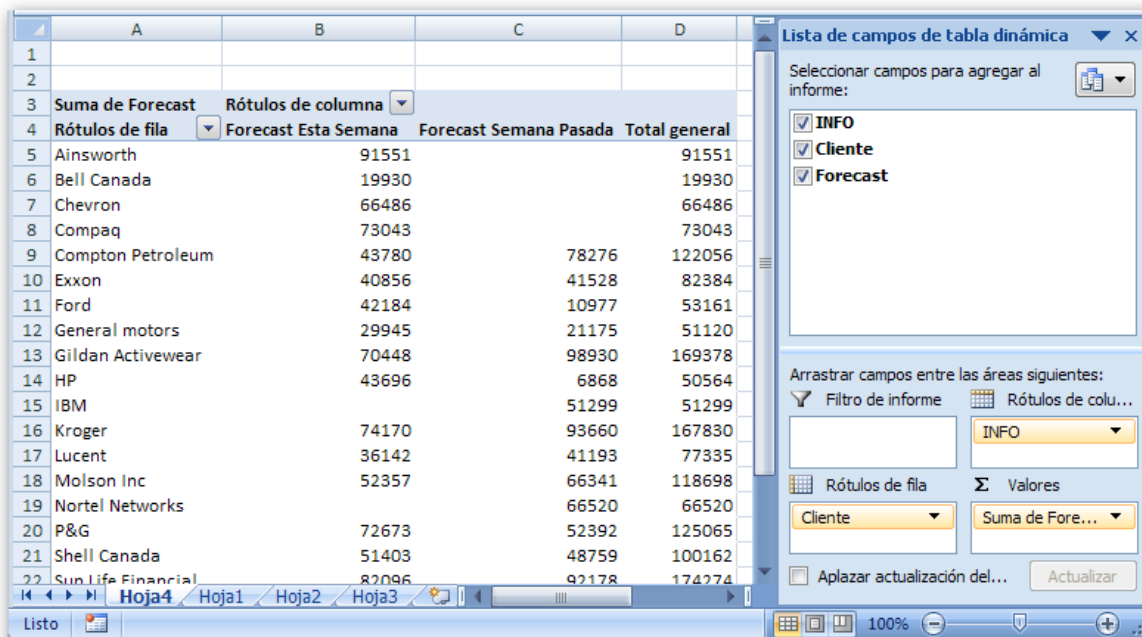
Hagamos nuevamente un ejemplo con cruces de información. Supongamos que tenemos un par de rangos, cada uno de ellos con el Forecast por cliente de la semana actual versus la semana anterior.

	A	B	C	D	E	F
1	Forecast Semana Pasada				Forecast Esta Semana	
2						
3	Cliente	Forecast			Cliente	Forecast
4	Compton Petroleum	78.276			Ainsworth	91.551
5	Exxon	41.528			Bell Canada	19.930
6	Ford	10.977			Chevron	66.486
7	General motors	21.175			Compaq	73.043
8	Gildan Activewear	98.930			Compton Petroleum	43.780
9	HP	6.868			Exxon	40.856
10	IBM	51.299			Ford	42.184
11	Kroger	93.660			General Motors	29.945
12	Lucent	41.193			Gildan Activewear	70.448
13	Molson Inc	66.341			HP	43.696
14	Nortel Networks	66.520			Kroger	74.170
15	P&G	52.392			Lucent	36.142
16	Shell Canada	48.759			Molson Inc	52.357
17	Sun Life Financial	92.178			P&G	72.673
18	Verizon	57.660			Shell Canada	51.403
19	Wal-Mart	82.523			Sun Life Financial	82.096
20					Verizon	1.807
21					Wal-Mart	99.636

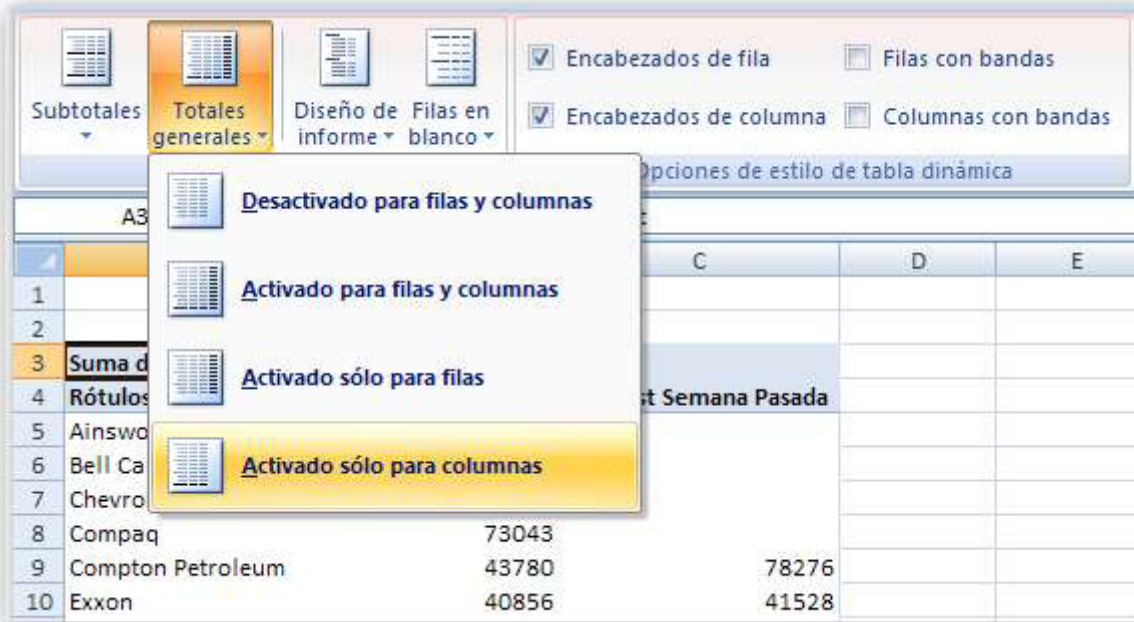
Bueno, de manera similar a como se acaba de explicar, se unen las dos tablas y en una nueva columna se identifica cada una de ellas, así:



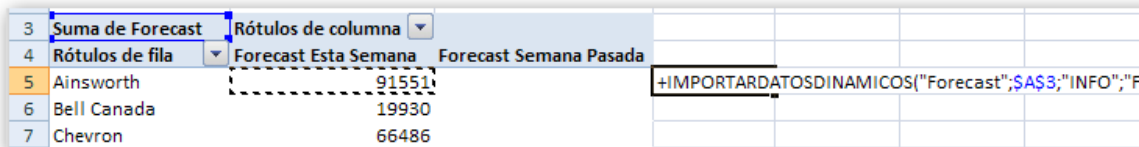
Con la INFO organizada, comienza a construir la tabla dinámica y dispone los campos así:



Por la opción Diseño de tabla dinámica, desactiva los totales de columna, así:



Verá como se quita ese total. Ahora viene el problemita que le comentaba: Si se quisiera restar la columna Forecast Esta semana MENOS Forecast semana pasada, nos sale una fórmula que nada tiene que ver con la tradicional, además, si probamos a copiarla, veremos cómo NO FUNCIONA:

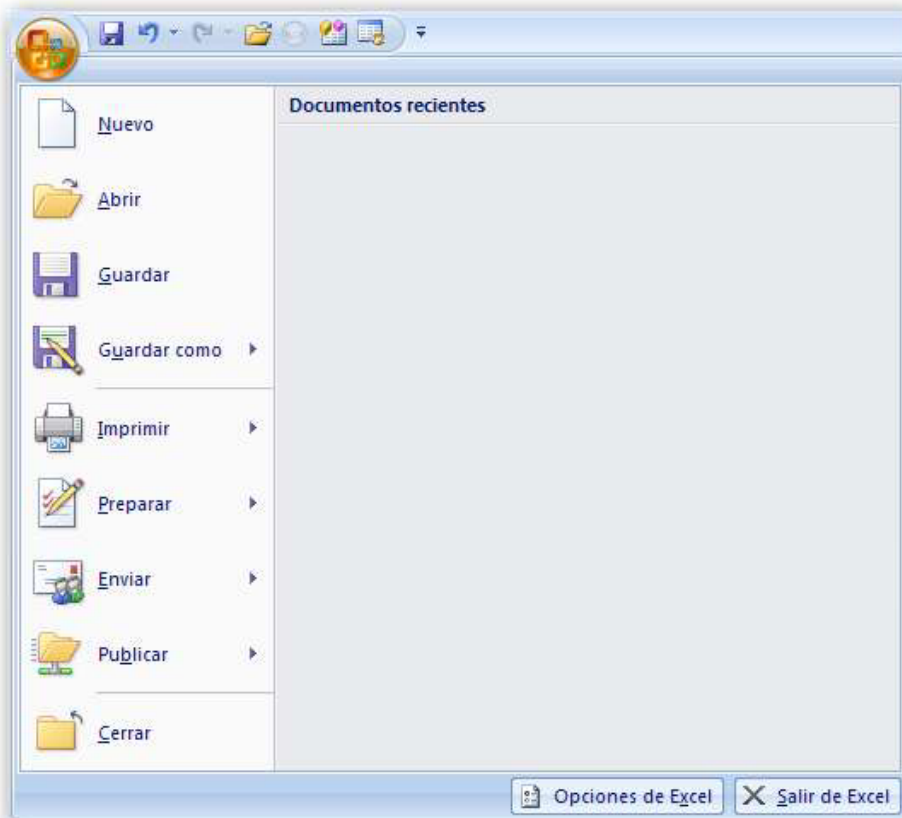


Como QUITAR esto? ASI:

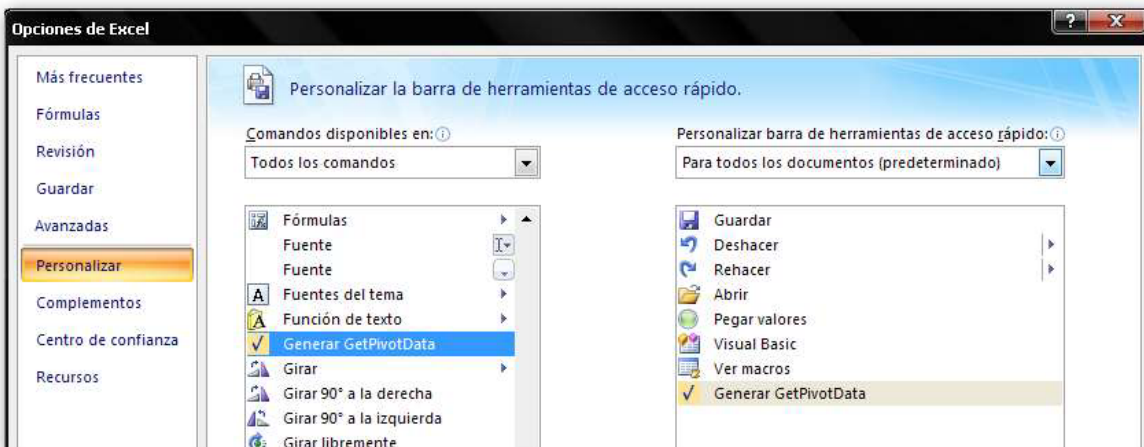
En Excel 2007, Vaya por el ícono de Office en el extremo superior izquierdo:



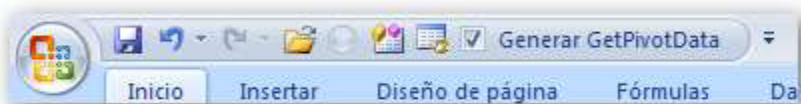
Haga Clic en ese ICONO, luego vaya por OPCIONES DE EXCEL:



En opciones de Excel, escoja PERSONALIZAR, y en comandos más utilizados, escoja TODOS LOS COMANDOS, y en ellos, dé Adjuntar el botón GENERAR GETPIVOTDATA:



La opción le quedará Disponible en los botones de acceso rápido:



Si desactiva la opción, podrá hacer operaciones SOBRE la tabla dinámica:

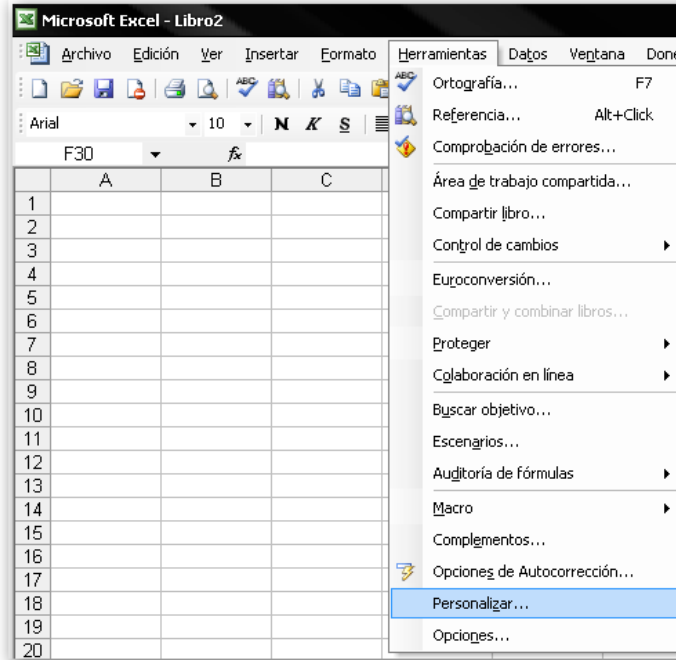
	A	B	C	D
1				
2				
3	Suma de Forecast	Rótulos de columna		
4	Rótulos de fila	Forecast Esta Semana	Forecast Semana Pasada	
5	Ainsworth	91551		+B5-C5
6	Bell Canada	19930		
7	Chevron	66486		

Y copiar la formula hacia abajo:

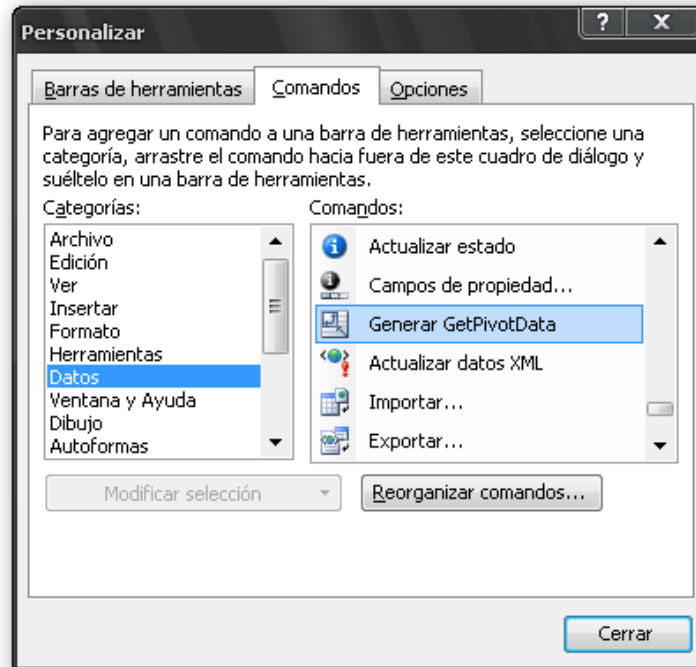
3	Suma de Forecast	Rótulos de columna		
4	Rótulos de fila	Forecast Esta Semana	Forecast Semana Pasada	
5	Ainsworth	91551		91.551
6	Bell Canada	19930		19.930
7	Chevron	66486		66.486
8	Compaq	73043		73.043
9	Compton Petroleum	43780	78276	(34.496)
10	Exxon	40856	41528	(672)
11	Ford	42184	10977	31.207
12	General motors	29945	21175	8.770
13	Gildan Activewear	70448	98930	(28.482)
14	HP	43696	6868	36.828
15	IBM		51299	(51.299)
16	Kroger	74170	93660	(19.490)
17	Lucent	36142	41193	(5.051)
18	Molson Inc	52357	66341	(13.984)
19	Nortel Networks		66520	(66.520)
20	P&G	72673	52392	20.281
21	Shell Canada	51403	48759	2.644
22	Sun Life Financial	82096	92178	(10.082)
23	Verizon	1807	57660	(55.853)
24	Wal-Mart	99636	82523	17.113
25	Total general	992203	910279	

Como activar la opción en Excel 2003:

Vaya por la opción Herramientas – Personalizar:



En la ventana, seleccione la ficha Comandos, y dentro de ella, DATOS, Generar GetPivotData. Arrastre el botón hacia la barra de herramientas:



Ok, es una forma que incluso es más sencilla que ponerse a cambiar el signo a una de las tablas fuente. A veces es mejor ser práctico que sofisticado. No lo cree?

Armando tablas dinámicas que leen más de 1.048.576 filas

En varias partes del libro he mencionado que, pasar de 65 mil filas al millón y pico de filas en el nuevo Excel me hizo muy feliz, y es verdad, pero cada vez que te enfrentas a mayores retos, te puedes encontrar que incluso el millón de filas puede no ser suficiente. Ya me ha pasado. Tuve que analizar Kardex demasiado grandes y te cuento que la cosa se me complicó un poco. Hasta que me acordé de la aplicación hermanita de Excel: El Todopoderoso Access.

La cosa fue así: baje el kardex de todo un año en archivo plano. La idea era cruzarlo con el auxiliar de la cuenta de inventarios y en un dos por tres saber cuáles eran las transacciones problema. Lo abrí en Excel 2007. **Qué pasó? Excel NO LO ABRIO COMPLETAMENTE.**

Como le mencionaba en los primeros temas del libro, en el apartado relacionado con arreglar los archivos planos con la opción Ordenar los datos, el 1.048.576 filas de Excel corresponde a una hoja de papel VIRTUAL de 5 kilómetros de largo. Por eso, en principio, se me hace difícil creer que haya un volumen mayor de datos... pero, como se dice, *"De todo se ve en la viña del Señor"*. Entonces, pues nada, generé el listado en bloques de 6 meses, lo cual pues también es aceptable, los volví base de datos e intenté unirlos. **Y qué pasó? No pude! Al unirlos, sobrepasaban el 1.048.576 filas....**

RAYOS. Entonces? Necesitaba presentar la conciliación que contuviera un año. No me servía presentar 6 meses.... Solución?

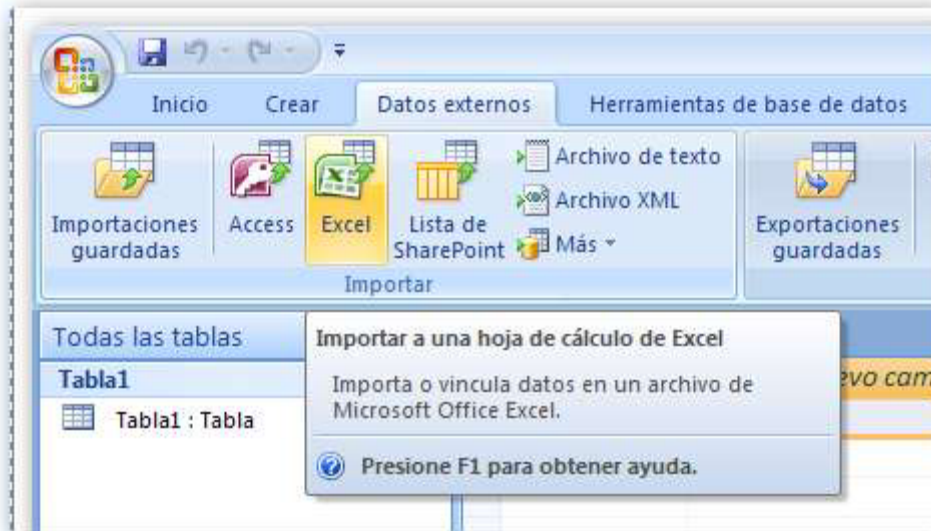
Utilizando Microsoft Access para crear una sola base de datos.

Lo que hice fue UNIFICAR los dos archivos en uno solo, con la ayuda de Microsoft Access. Cómo? Así:

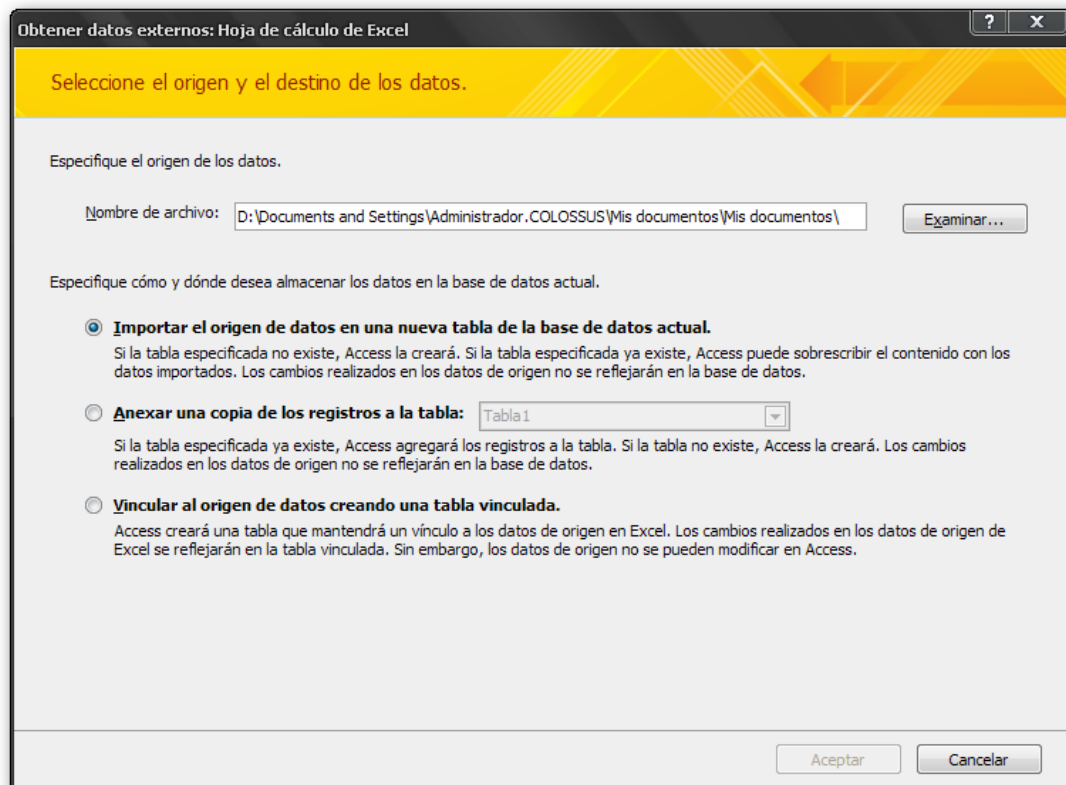
- 1) Usted ya tiene los dos archivos que corresponden a las dos bases de datos en Excel. Ojo, Deben ser IDENTICOS EN ESTRUCTURA.
- 2) Abre el Access.



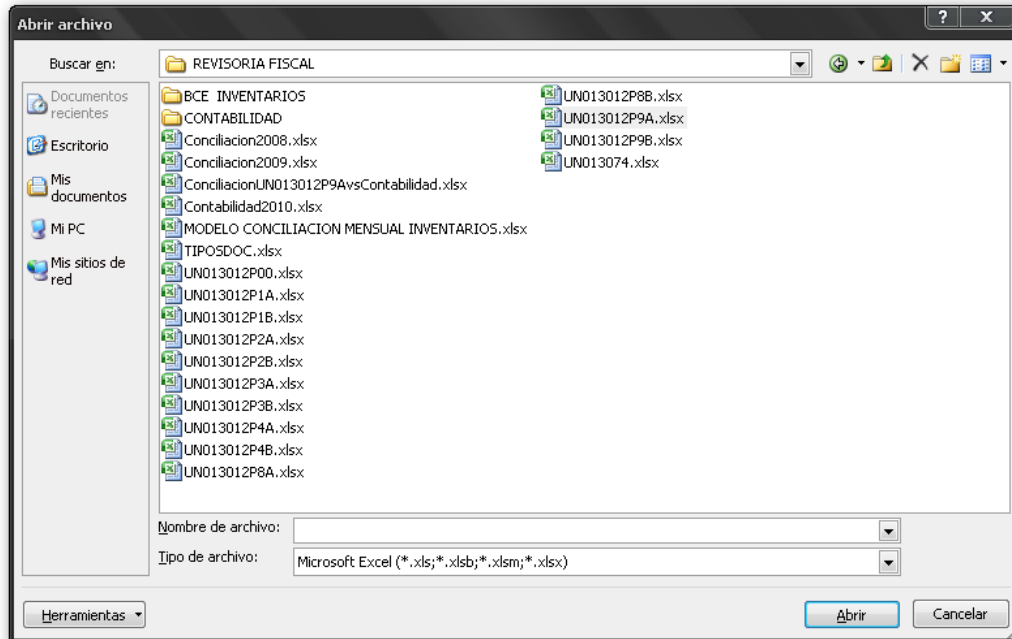
- 3) Abre una nueva base de datos en blanco, clic en crear.
- 4) Clic en Ficha, datos Externos, Excel:



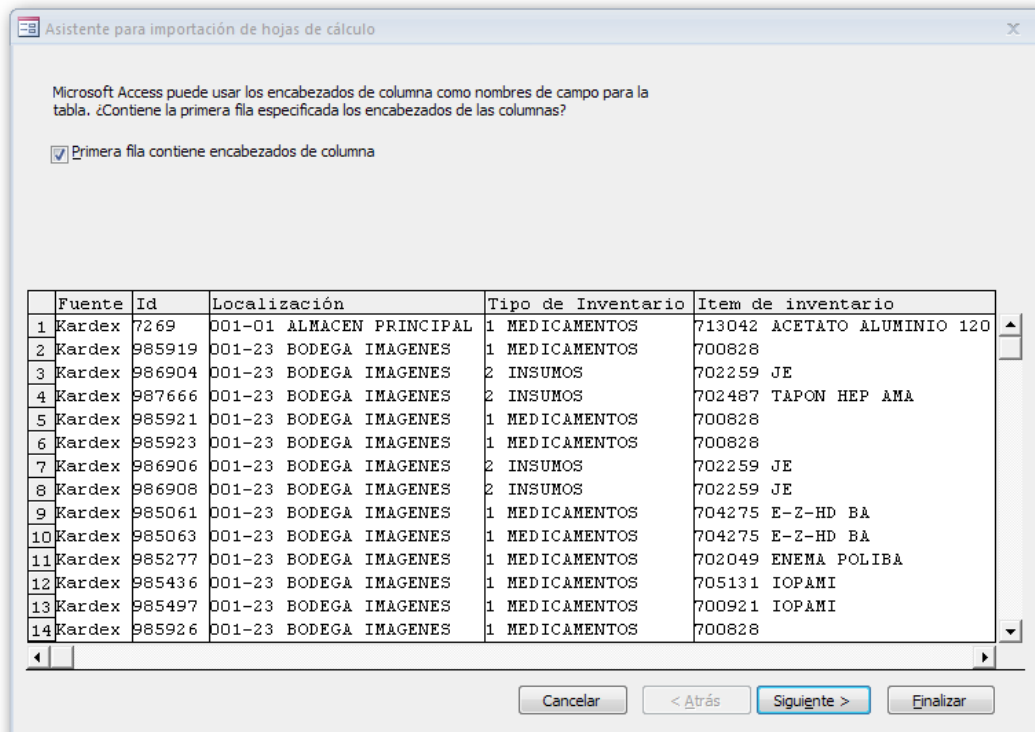
5) Te saldrá este cuadro de diálogo:



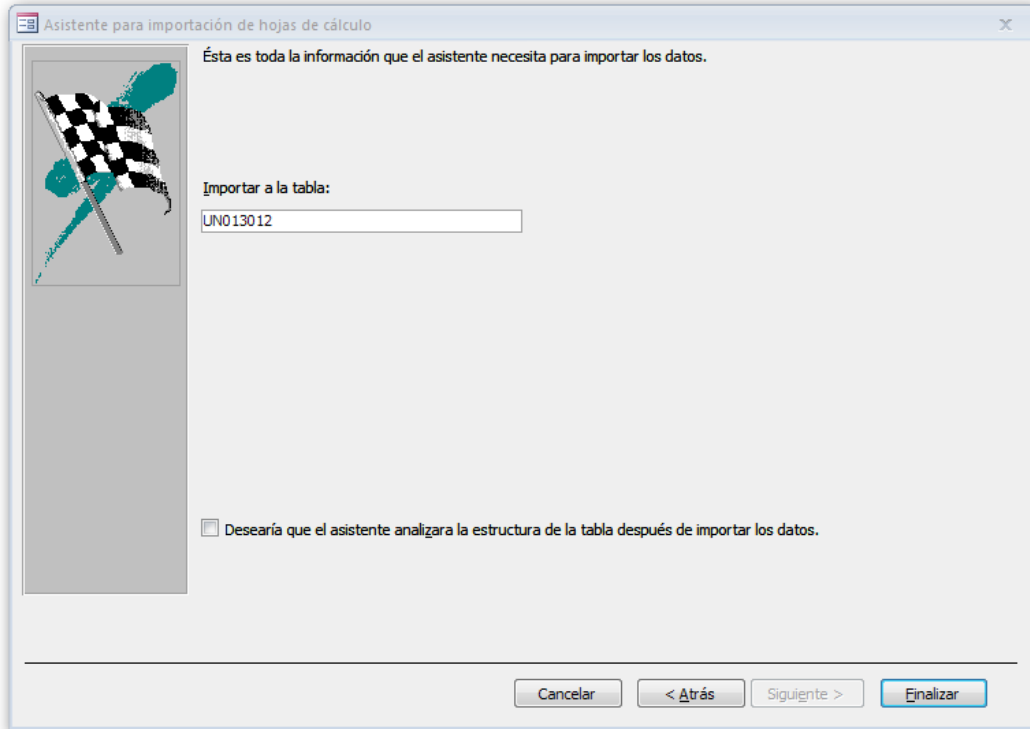
6) En el botón examinar, que es una especie de EXPLORADOR, busque la primera de las bases de datos EXCEL:



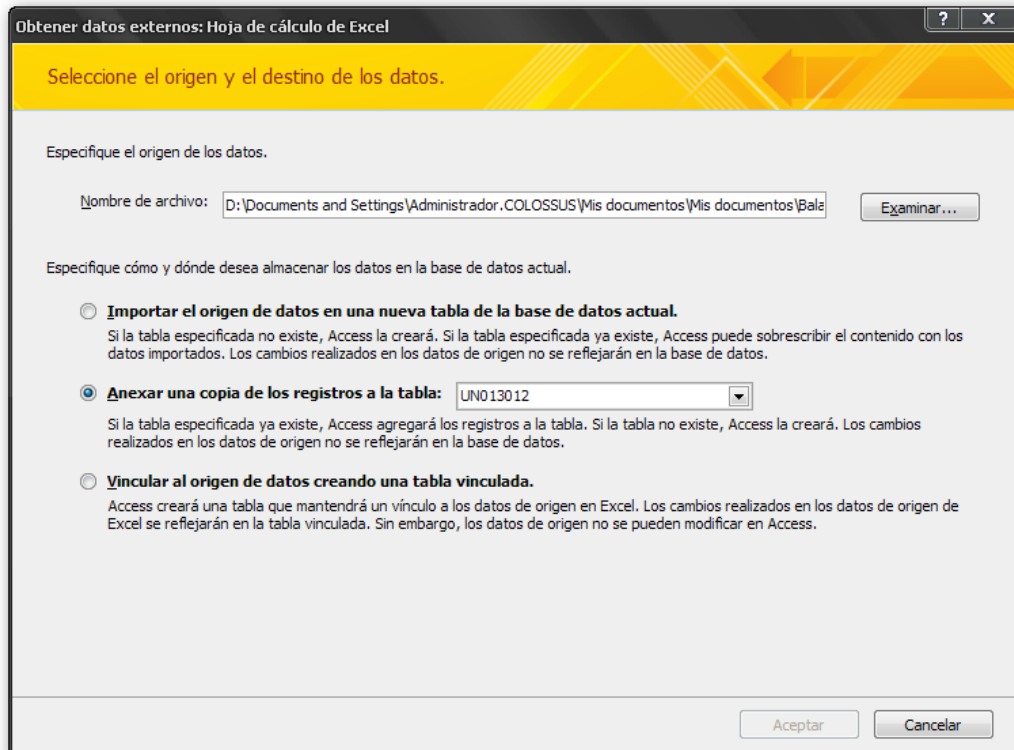
- 7) Seleccione uno de los archivos que corresponden a la primera parte de uno de los años del kardex, y haga clic en abrir. Regresará al momento anterior que veíamos en Access, (la imagen anterior a esta) clic en aceptar. Access comenzará a importar el archivo. Al finalizar tendrá algo así:



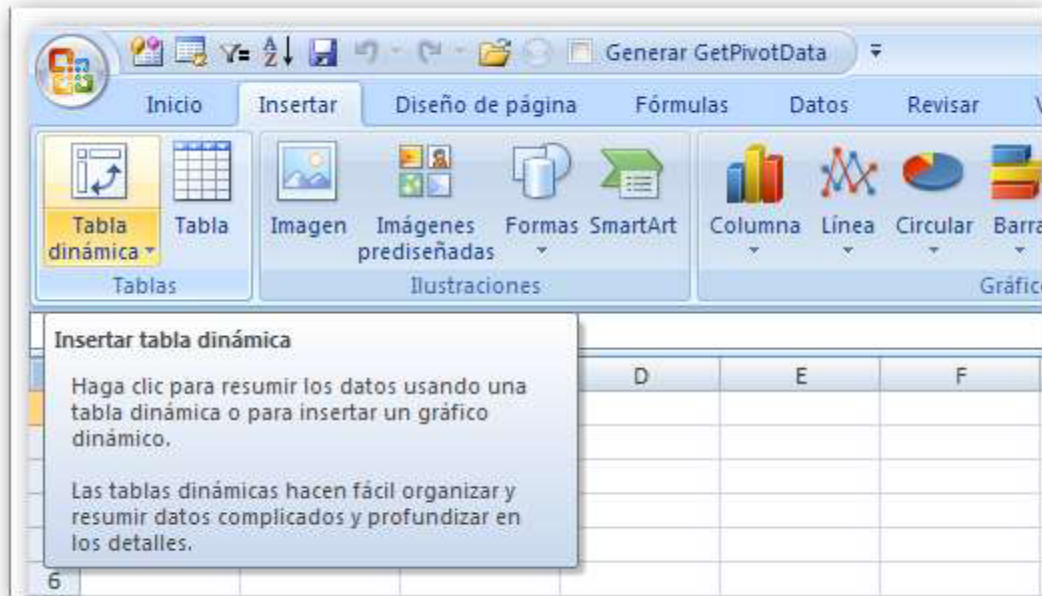
- 8) Siga las instrucciones, acepte las sugerencias de Access hasta llegar a la opción finalizar:



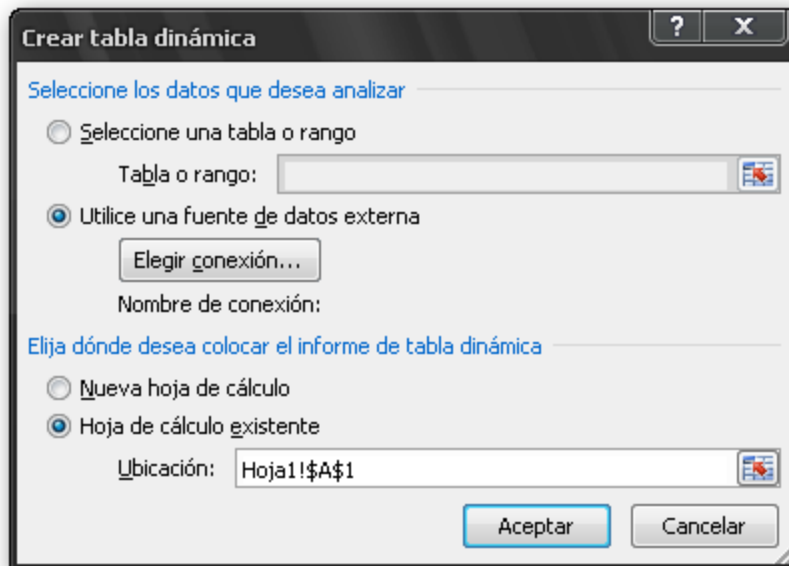
- 9) En el último paso, clic en cerrar. Nuevamente importe la otra tabla cuidando que quede formando parte de la tabla que acabó de importar:



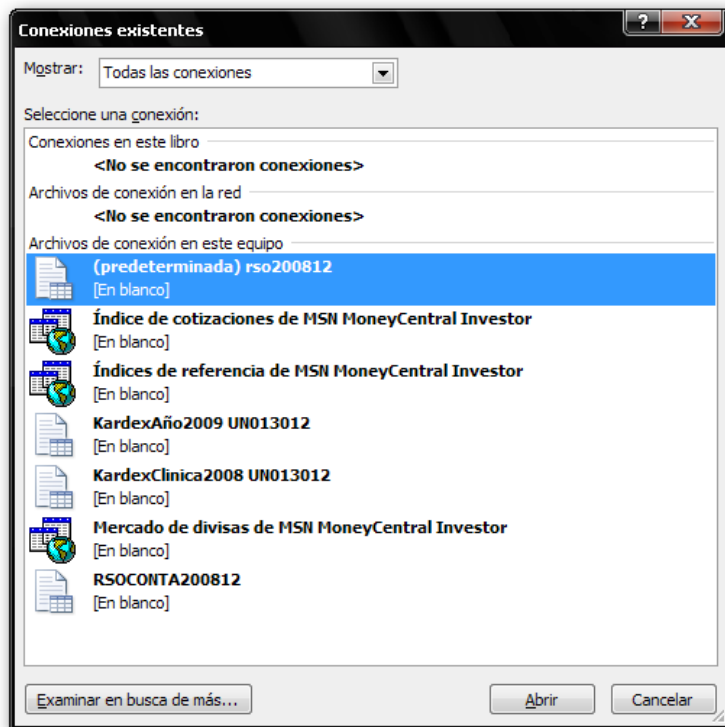
- 10) Cuando ya tenga ambas tablas unidas, guarde el archivo en Access. Es decir, como un archivo Access... o sea un archivo con extensión .accdb
11) Cierre la aplicación Access y abra Excel.
12) En un libro nuevo, vaya por la ficha insertar y escoja tabla dinámica:



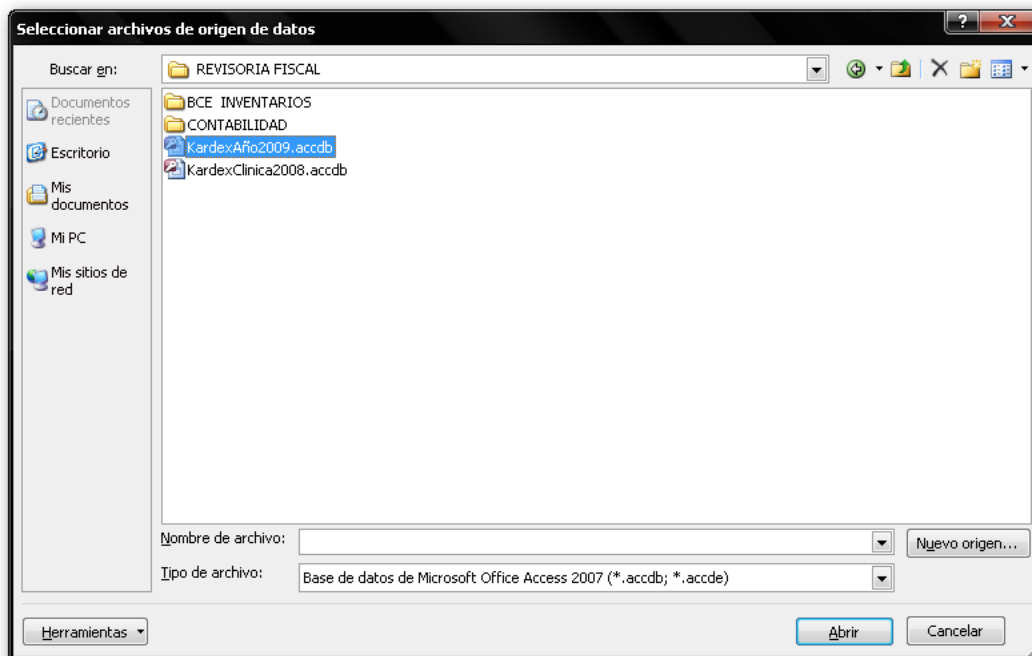
13) En crear tabla dinámica, vaya por la opción Elegir fuente de datos externa:



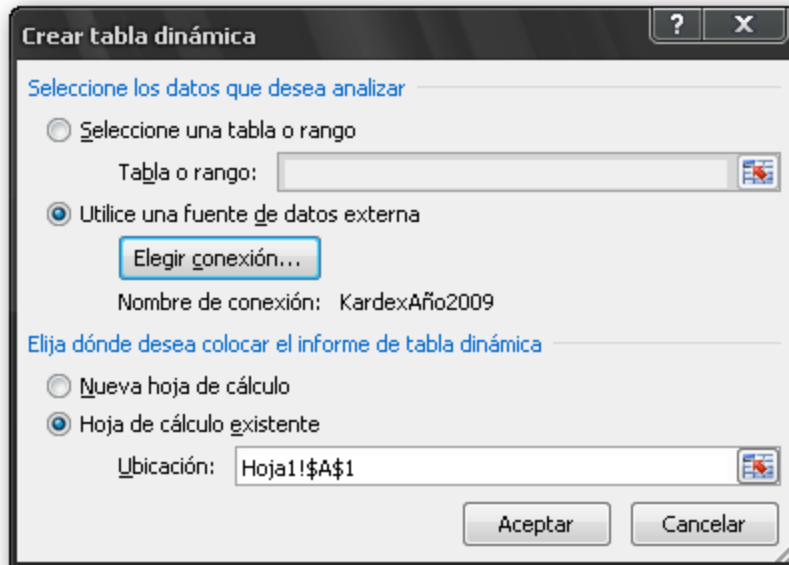
14) Haga clic en elegir conexión



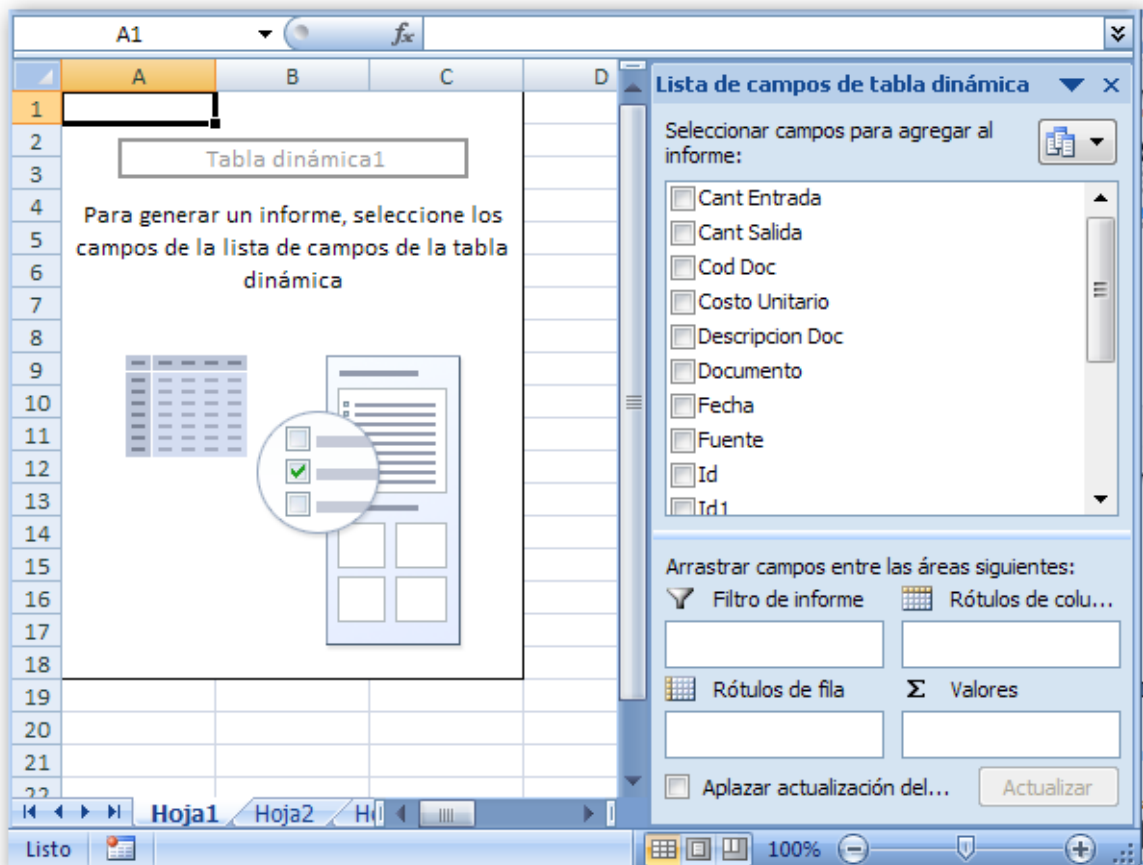
15) Clic en "Examinar en busca de mas" y busque el archivo que unificó con la ayuda de Access, dele Abrir:



16) Una vez seleccionado el archivo, haga clic en aceptar:



La tabla dinámica comenzará a leer los datos, podrá ver como supera sin problema el 1.048.576 filas, y podrá armar la tabla dinámica a su conveniencia:



Tenga en cuenta que está manejando una cantidad impresionante de datos, así que tenga paciencia si su equipo se pone RELENTO. Paciencia, mientras se compra un equipo de cómputo mejor.

Ahora, si quiere hacer el cruce de información, del kardex y el auxiliar, acuérdesese de cómo realizar el cruce de información que le explique en uno de los temas de tablas dinámicas, concretamente como organizar las dos tablas y únalas en un archivo MADRE con la ayuda de Access. En cuanto comience a montar la tabla dinámica, verá que cuenta con el campo "origen", que es el que identifica cada una de las bases unidas y tendrá que la tabla dinámica le hará la SUPERCONCILIACION, basada en el archivo externo de Access.

De manera similar, la tabla dinámica puede leer cualquier fuente de datos, Oracle, SQL, Fox Pro, etc. No necesita cargar en Excel la supertabla o base entera, puede hacer que Excel la lea de manera Externa, y de esta manera no se verá detenido por la limitante del 1.048.576 filas.

Si tiene dificultades, pídale ayuda al Ingeniero de sistemas, ellos están muy familiarizados con el manejo de tablas en SQL, Oracle, Fox o aún el viejo Débase.

Automatizando las tareas con Excel VBA

No está de más recordarle que el Excel 2007 trae más de UN MILLON CUARENTA MIL FILAS. Le cuento, por experiencia propia, que hasta movimientos contables mensuales de empresas grandes se pueden manejar con este nuevo Excel. ...Es que uno siempre quedaba limitado por las viejas 65 mil filas de Excel 2003, solo con esta mejora, me hicieron muy feliz. Ya sabe, si leyó el tema anterior, que incluso Excel puede trabajar directamente en las bases de datos con lo cual puede trascender el límite del millón cuarenta mil filas...

En números: Diferencias entre Excel 2003 versus Excel 2007:

	2003	2007
Numero de Filas	65.536	1.048.576
Numero de Columnas	256	16.384
Cantidad de memoria usada	1 Giga	El Limite es fijado por Windows
Numero de colores	256	4.3 Billones
Numero de formatos condicionales por celda	3	Ilimitados
Numero de niveles al ordenar	3	64
Numero de niveles para deshacer	16	100
Numero de ítems que se muestran en el autofiltro	1.000	10.000
Numero de caracteres que se puede mostrar en una celda	1.000	32.000
Numero de estilos únicos en un libro	4.000	64.000
Máximo número de caracteres en una formula	1.000	8.000
Numero de anidaciones en una formula	7	64
Numero máximo de argumentos en una formula	30	255

Fuente: pulse el botón F1 de ayuda de su Excel 2007

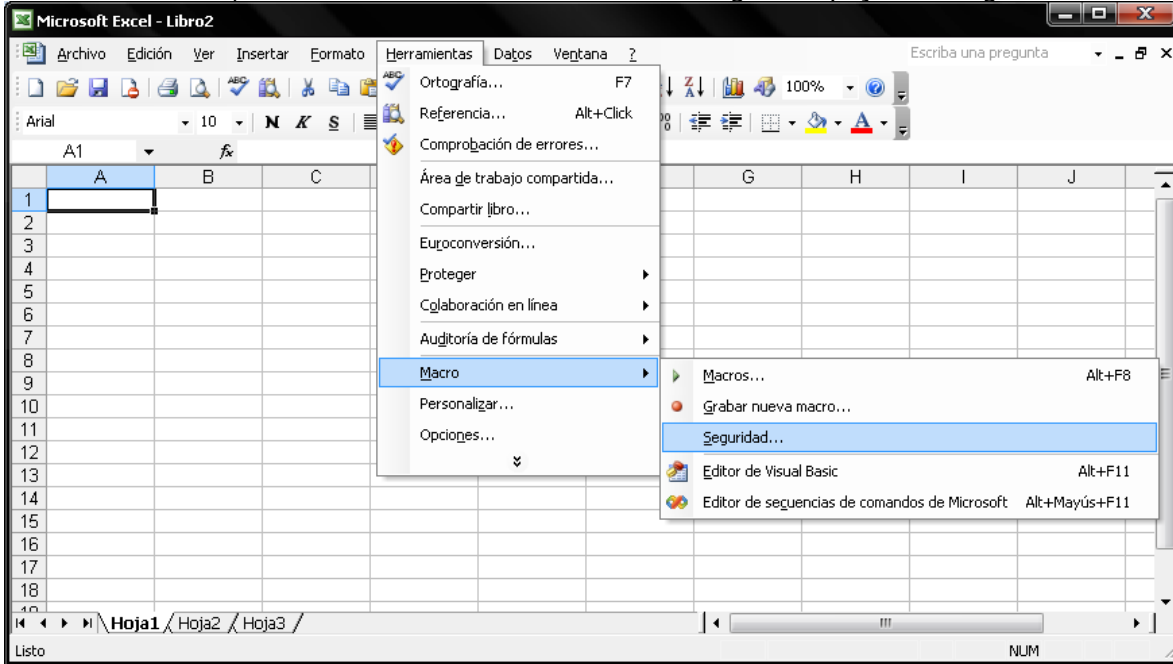
El tema de las macros es pletórico. Esta parte pretende enseñarle lo mínimo necesario para que pueda crear macros por lo menos funcionales. Pero muchos temas se quedan por fuera, si le gusta el tema, deberá comprar libros especializados en el tema. Si después de esto, le coge gusto, estoy seguro que no será tan difícil abordar libros técnicos especializados.

Yo creo que, si Usted leyó el tema relacionado con "como rellenar los espacios en blanco", allá tuvo que ver como con una pequeña macro, nos podemos deshacer de una gran cantidad de trabajo EN SEGUNDOS!!! Es una cosa INCREIBLE, MONTAÑAS de trabajo pueden ser evacuadas EN SEGUNDOS!!! Razón más que suficiente para aprender el tema.

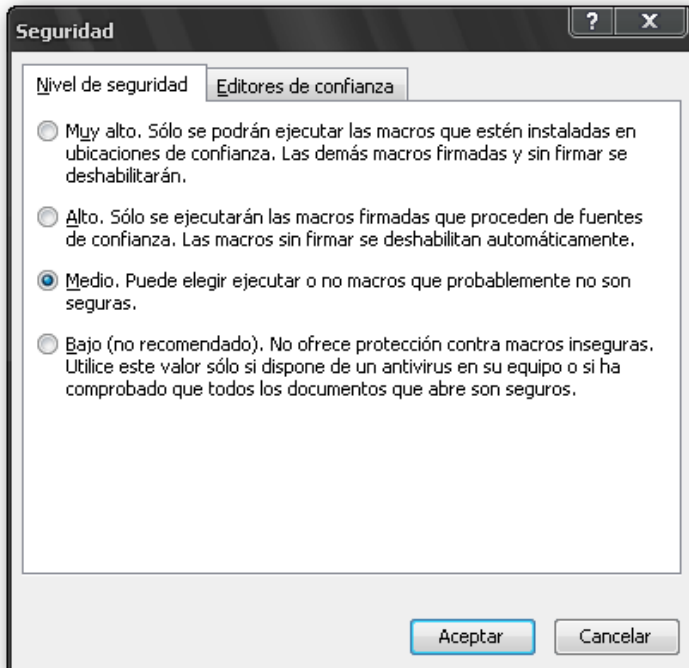
Quiero que sea consiente de como en este siglo XXI, el concepto angular de la dificultad se vuelve al revés. Por el momento es al nivel Informático de Bytes. Pronto también será realidad a nivel atómico y con ello me refiero a los tremendos adelantos que se llevan a cabo en la manufactura molecular. El planeta avanza a pasos agigantados, no pierda de vista los adelantos y no se resista al desarrollo tecnológico, o será arrasado de la arena profesional. "una palanca y moveré al mundo", decía aquel sabio griego.

Otorgando el permiso necesario para que el código de las macros corran o se ejecuten en su computadora

El primer paso antes de usar macros es activar el Visual Basic en el Excel de su máquina computadora. En Excel 2000-2003, utilice la ruta Herramientas – Macro – Seguridad y ajuste la seguridad a MEDIO.



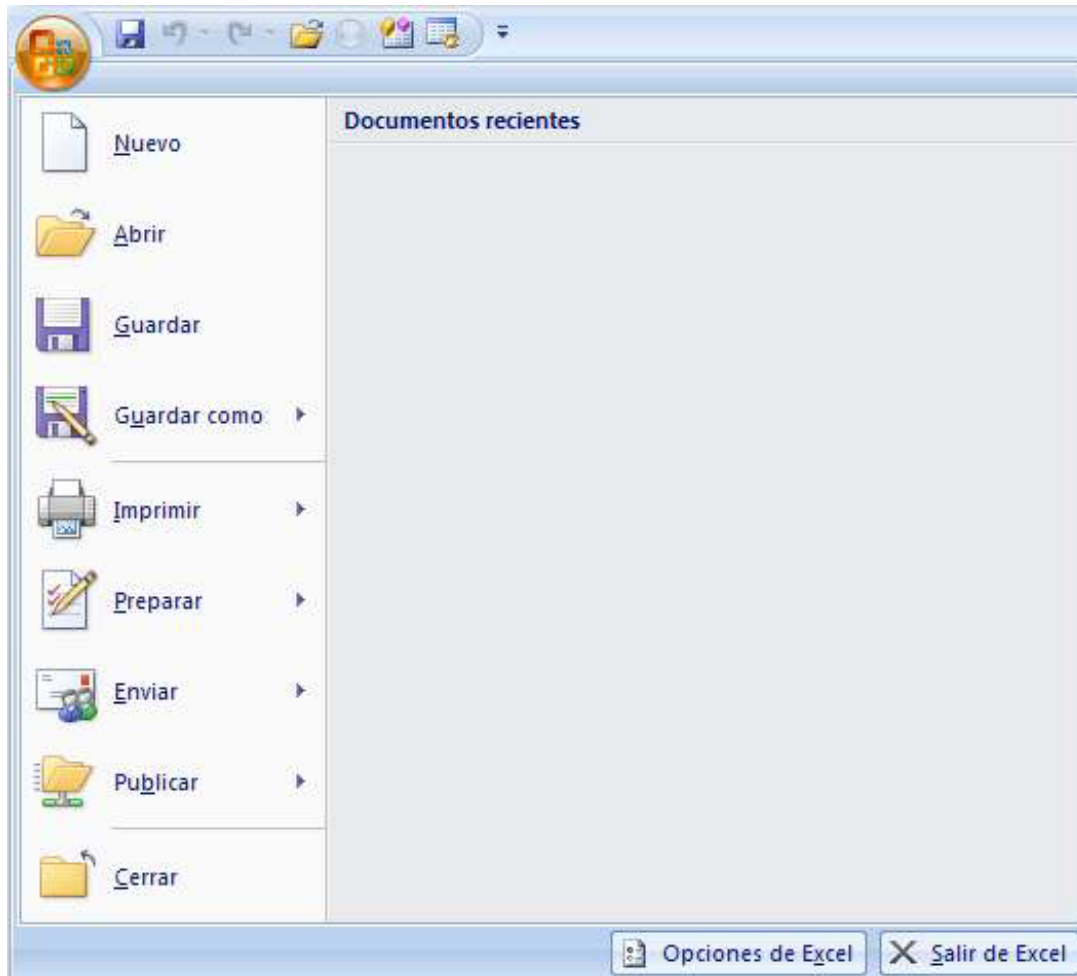
Cuando se da clic en Seguridad, como se ve en la imagen, aparece el cuadro de dialogo de seguridad:



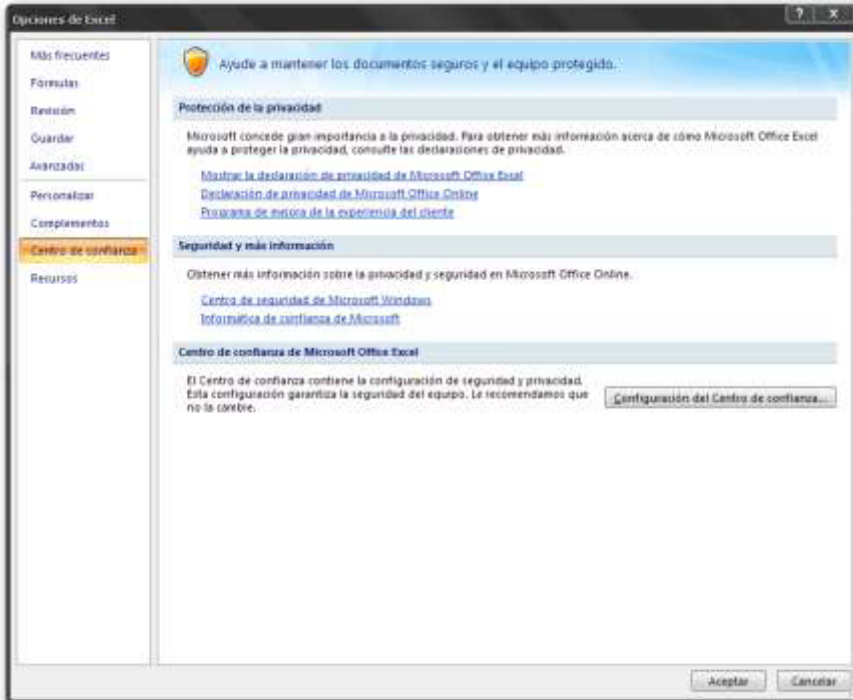
Ok. Ahora, en Excel 2007, vaya por el icono de Office que se encuentra en el extremo superior izquierdo:



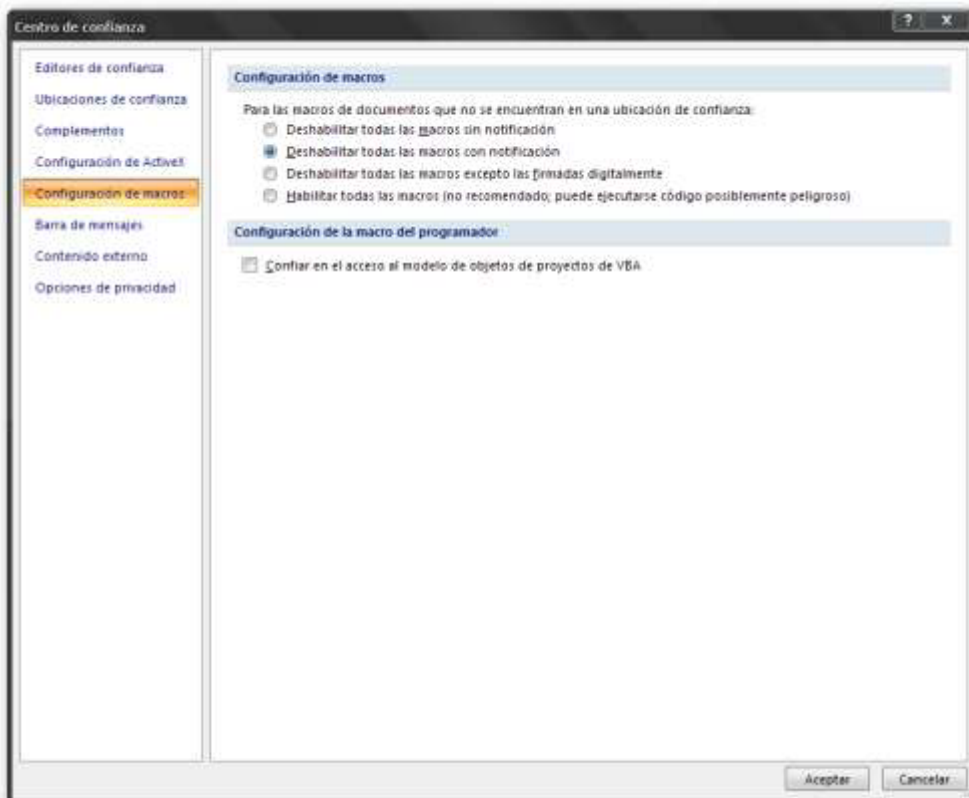
Osea, al hacer clic allí, se desplegará esto:



Aquí, hace clic en Opciones de Excel, y allí, en centro de confianza. Llegará aquí:



De clic en configuración de centro de confianza. Llegará aquí:



Si no llega, vaya por la ficha programador

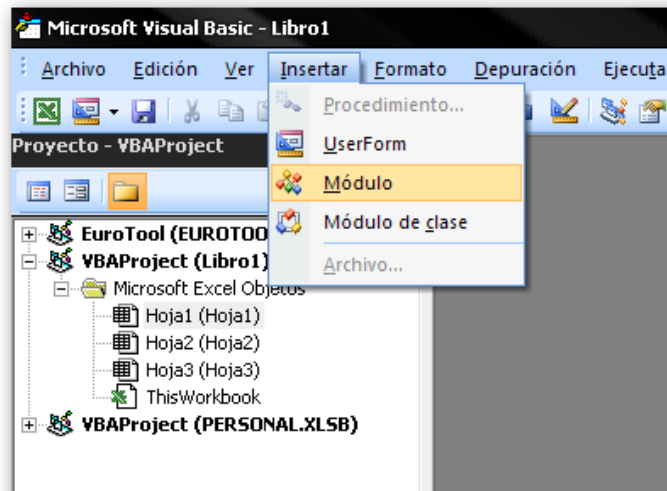
Insertando el modulo VBA en su libro de trabajo

Las macros normalmente se escriben en un modulo VBA de su libro de trabajo. Un libro puede contener muchas hojas de calculo así como hojas de gráficos, también puede contener muchos módulos VBA.

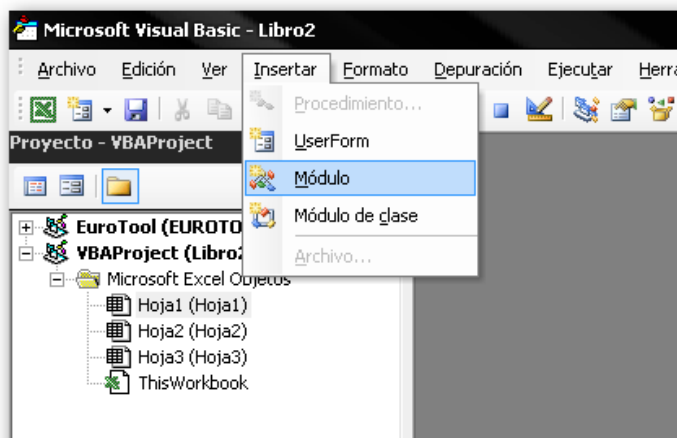
En un proyecto complejo, es recomendable colocar las macros que crean reportes en un modulo y las que se refieren a seguridad o captura de info en otro modulo. Cada módulo puede contener muchas macros.

Pero macros pequeñas y sencillas se pueden guardar en un solo módulo.

En la macro copiar vimos un poco de esto, pero no está de mas recordar la ruta para insertar un modulo: Ficha programador – Visual basic. Una vez allí, Menú insertar – módulo.



En Excel 2003 la cosa es muy similar: Herramientas – Macro - Visual basic Editor. Una vez allí, Menú insertar – módulo.



Escribiendo y corriendo una macro

Una vez que inserte el modulo, verá un panel blanco al lado derecho, el cual se llama Panel de Código. Allí es en donde se escriben las macros.

Para comenzar a escribir una macro, escriba la palabra Sub, un espacio, un nombre para la macro y un par de paréntesis, uno que abre y otro que cierra. Por ejemplo:

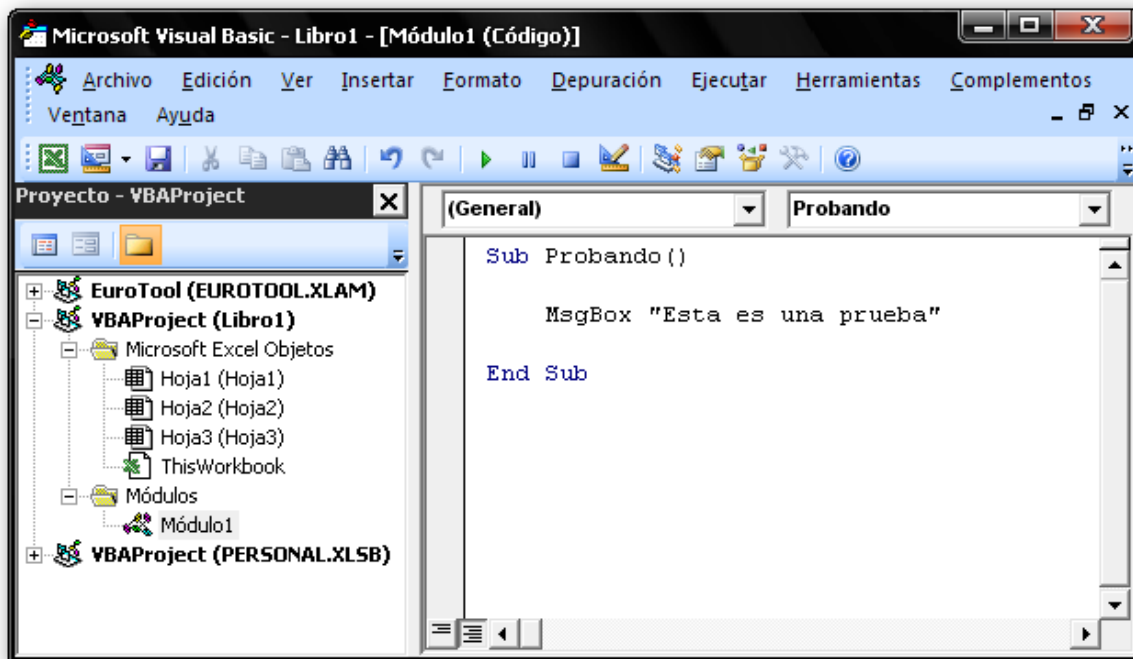
```
Sub Probando()
```

Cuando escriba esta línea, presione ENTER. Excel crea una nueva macro llamada Probando. Visual Basic adiciona una línea en blanco, y la línea final de la macro con la palabra End Sub.

Cualquier instrucción que escriba entre las líneas Sub y End Sub se ejecutan cuando usted corre la macro.

Haga clic en la línea en blanco entre Sub y End Sub, y escriba esta línea:

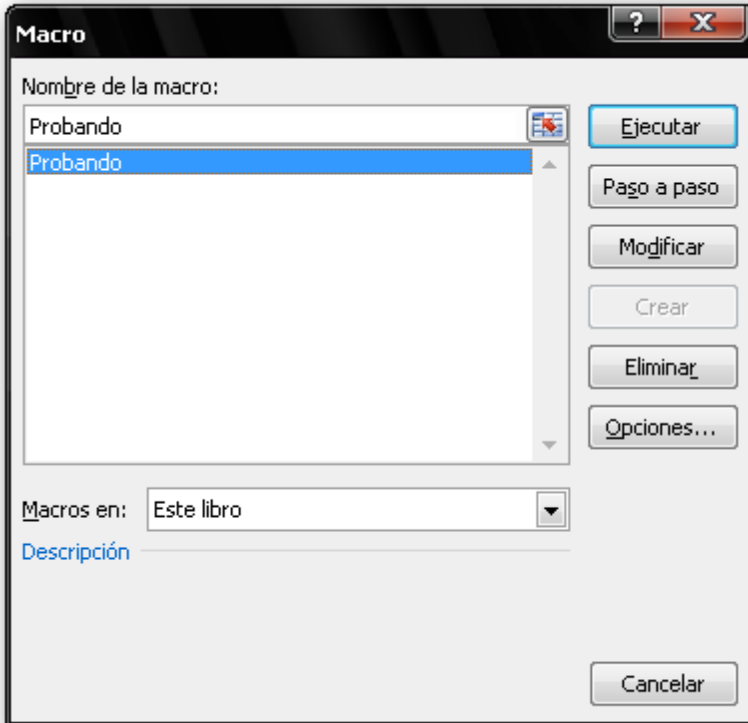
```
Msgbox "Esta es una prueba"
```



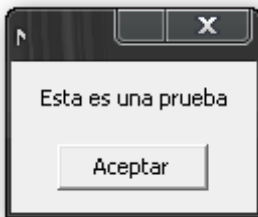
Haga clic en el ícono de Excel y regrese a la hoja de cálculo de Excel. Si está siguiendo estas instrucciones paso a paso, ahora va a probar si la macro funciona.

Una manera de correr una macro es llamar las macros disponibles, lo cual puede hacer presionando Alt+F8 en cualquier versión de Excel. En Excel 2007 las puede llamar a través de la ficha programador. En las versiones anteriores, yendo por el menú Herramientas – Macro – Macros.

Tal como se muestra en la siguiente imagen, el cuadro de dialogo de las macros lista todas las macros disponibles en su libro. Haga clic en la macro y presione el botón ejecutar:



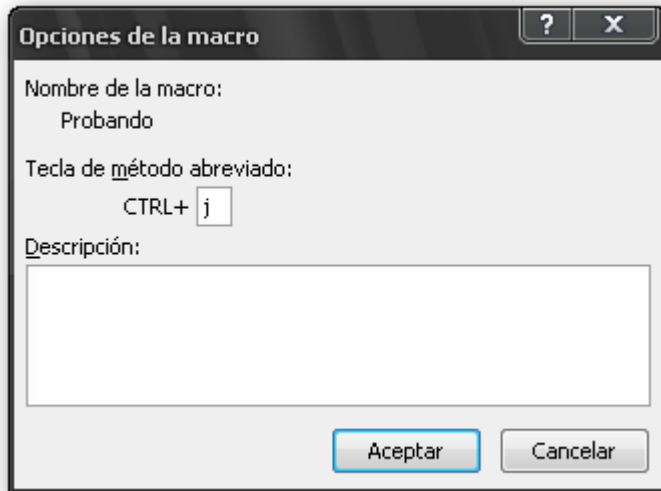
Esta macro sencillita le muestra el texto en una caja de mensajes:



Puede no ser muy efectivo tener que traer la ventana de las macros, sobretodo si se tiene que ejecutar periódicamente. Es mejor asignar la macro a un atajo de teclado, una combinación de letras que al oprimirlas simultáneamente, ejecuten la macro.

Para asignar una combinación de letras que sirva para correr la macro, haga lo siguiente:

1. Llame las macros.
2. Haga clic en una de las macros.
3. Escoja el botón Opciones
4. En la ventana de las opciones de macro, escriba una combinación de teclas. Ctrl+J y Ctrl+K son buenas opciones, pues no hay comandos de Excel asignados a esta combinación de teclas.



5. Haga clic en Aceptar para cerrar la ventana opciones de macro. (esta que ve aquí)
6. Clic en Cancelar para cerrar LA OTRA ventana de Macros.

Ahora puede correr la macro si presiona Ctrl+J al mismo tiempo.

Una macro que recorra todos los registros de la hoja de Excel

Por lo general, una macro sigue la siguiente estructura:

Comienza en el primer renglón de información de la hoja de cálculo.
Examina un valor en una de las columnas.
Basado en este valor, decide hacer algo.
Se mueve al siguiente renglón en el rango de datos y repite.

Para crear esta estructura de macro siga los siguientes pasos:

1. En su Excel, identifique la primera fila con información. También identifique la última fila de información.
2. Llame el Editor de Visual Basic con Alt+F11
3. Si no está listo el modulo en el cual escribir la macro, insértelo usando el menú Insertar-Modulo.
4. Escriba la palabra Sub, dele un nombre a su macro y después un doble paréntesis de estos: (). En este ejemplo estoy denominando la macro así:

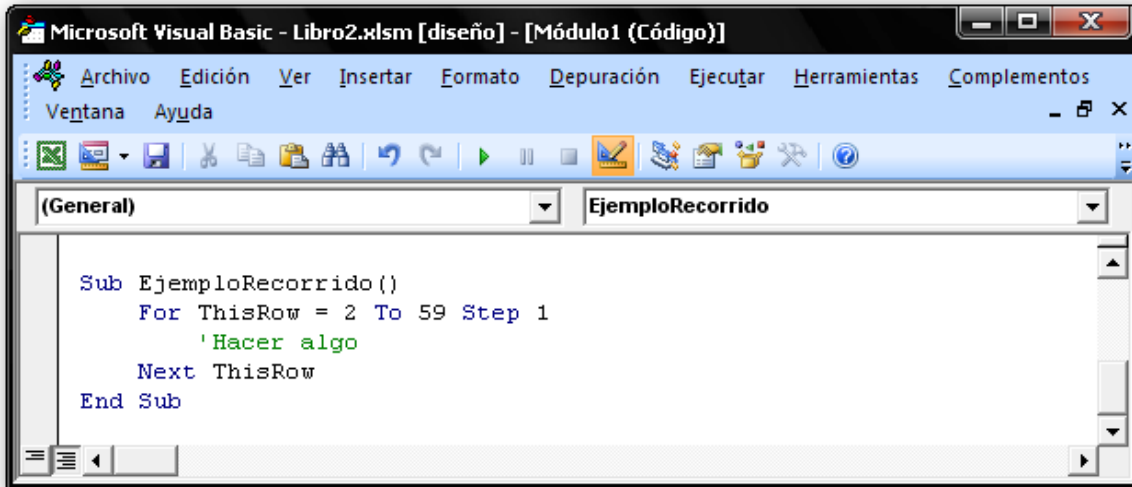
```
Sub EjemploRecorrido()
```

5. El comando para un recorrido es la palabra "For" seguido de un nombre variable. Los nombres variables pueden ser I o X, o nombres tales como This row (Esta fila) Usted debe especificar la primera y la ultima fila del recorrido, para el ejemplo el recorrido será especificado así: (Para EstaFila = 2 a 59, saltando de a uno)

```
For ThisRow = 2 to 59 Step 1
```

6. En la siguiente línea o renglón escriba un apostrofo y las palabras "Hacer algo"
7. En la siguiente línea, finalice el recorrido o Loop con el comando Next ThisRow.

Esta macro recorrerá todos los registros de información de su hoja de calculo "Step 1" (paso 1) significa que la macro revisará cada fila. Si se usa "Step 2" revisará la segunda fila distinta a la actual. En este momento, la macro no hará nada aparte de recorrer la hoja de cálculo, pues la instrucción se indica en donde colocamos el comentario del apostrofo "Hacer algo". Seguidamente le diremos que haga o ejecute alguna cosa, con lo cual le dará una idea de lo que la macro puede hacer.



Como referenciar las celdas en el recorrido

Uno puede referirse a una celda de una fila y columna particular usando el comando

`CELLS (Row, Column)` *quiere decir CELDA(Fila, Columna)*

Por ejemplo, la celda J2 esta en la segunda fila y la décima columna (si verifica que numero de columna es la columna J, se dará cuenta que es la número 10), entonces `CELLS(2,10)` se refiere a la celda J2.

Nota: Es importante indicar que el grabador de macros generaría en lugar de `Cells(2,10)`, la celda J2, esta otra notación que es igualmente válida, aunque fija, lo cual es una desventaja como veremos a continuación:

```
Range("J2").Select
```

Lo bueno de referirse a las celdas mediante la técnica `CELLS(Fila,Columna)`, es que las coordenadas de la celda pueden ser una variable, o sea, no tienen que definirse fijamente sino que pueden ser cambiantes. Por ejemplo, para referirnos a la columna D en la fila corriente de la hoja de cálculo, puede utilizar `CELLS(ThisRow, 4)`. Para referirse a la columna A en la fila corriente de la hoja de calculo, puede usar `CELLS(ThisRow, 1)`. Para referirse a la columna E en la fila anterior de la hoja de calculo, puede utilizar `CELLS(ThisRow -1, 5)`.

Referenciar la fila completa usando CELLS

En algunas ocasiones, mientras la macro hace el recorrido uno puede necesitar referirse a la fila entera de la hoja de cálculo. Por ejemplo, ocultarla o colorearla. Utilice el comando `.EntireRow` (Fila entera).

Cualquiera de estos comandos ocultarán la fila corriente en el recorrido:

```
CELLS(ThisRow, 5)EntireRow.Visible = False
CELLS(ThisRow, 1)EntireRow.Visible = False
```

Referenciar un rango rectangular usando CELLS

Si quisiéramos aplicar un color o un formato negrita, podemos hacerlo a un rango. Podemos hacerlo por ejemplo, a 6 columnas, no a la fila entera.

El comando `.RESIZE(#Rows,#Columns)` hace que el comando `CELLS` que se refiere a una sola celda, se refiera a un rango rectangular comenzando por la Celda como el comienzo de la esquina superior izquierda.

Nota: Resize significa cambiar la medida o cambiar el tamaño Rows Filas, Columns Columnas

Por ejemplo, para referirse a un rango de una fila por seis columnas, puede usar:

```
CELLS(ThisRow, 1).RESIZE(1, 6)
```

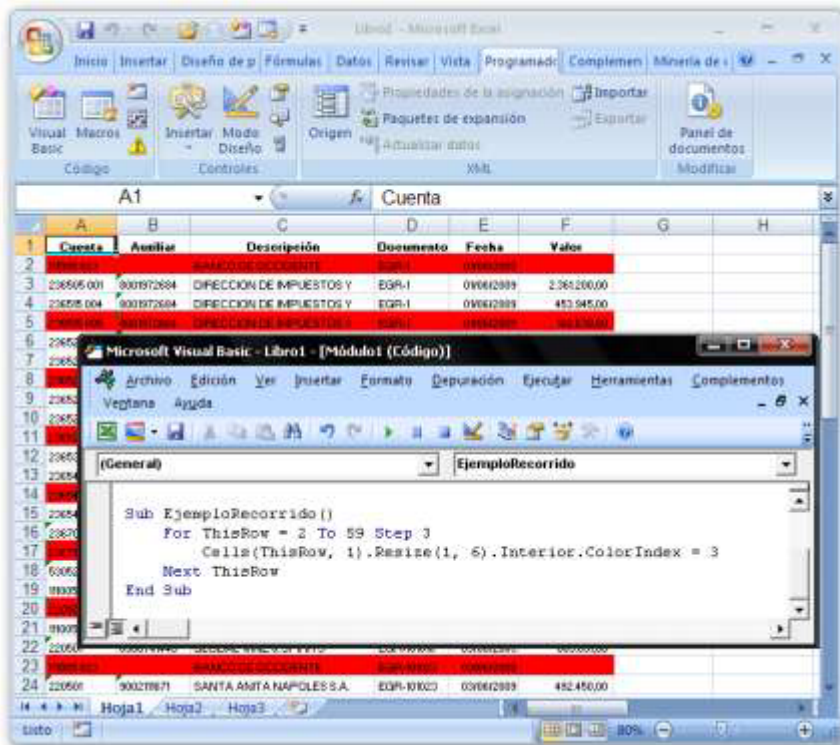
Este otro comando colorea las columnas A hasta la F en la fila actual en rojo:

```
CELLS(ThisRow, 1).RESIZE(1, 6).Interior.ColorIndex = 3
```

Macro que nos colorea cada tercera fila de rojo

Bueno, con lo visto anteriormente ya podemos montar una macro que recorra nuestra hoja de calculo, y cambiando el comando `STEP` para que haga el paso cada tercera fila. En la macro que estábamos escribiendo, en lugar del "Hacer Algo" vamos a cambiar el color de las primeras seis columnas en esa fila a rojo.

La macro que se muestra a continuación corre un pocos segundos (o milisegundos) y es mucho mas rápida que si se creara un nuevo estilo de tabla o formato condicional.



Tomando decisiones basados en los valores hallados en la fila

Avanzando un poco mas en las técnicas vistas, podemos hacer que dependiendo del valor de la celda, se ejecute una acción o no. Si Usted conoce la función Si, esta estructura le será mas fácil de entender. Se puede hacer que una macro tome una acción o decisión con la estructura IF.....THEN....ELSE....END IF (Sí Talcosa....Entonces, Sinó, FinSi) Por ejemplo, este código revisa cada registro de la columna E. Si esta columna contiene la palabra "Servicios", entonces el valor de la columna F se copia a la columna G y de una vez, la fila la podemos colorear de verde. En caso contrario, haremos que la pinte de color rojo.

Recuerde que se pueden comentar los pasos de la macro anteponiendo un apóstrofo

```
Sub SeparandoServicios()

`Esta macro mueve los ingresos por servicios de la columna F a la G
`Pinta las filas de los ingresos por servicios de Verde
`Pinta las filas de los ingresos por productos de Rojo

    For ThisRow = 2 To 16171 Step 1

        If Cells(ThisRow, 5) = "Servicios" Then

            `Movemos el valor de la columna 6 a la columna 7
            Cells(ThisRow, 7).Value = Cells(ThisRow, 6).Value

            `Ponemos un Cero en la columna 6
            Cells(ThisRow, 6).Value = 0

            `Coloreamos la fila de Verde
            Cells(ThisRow, 1).Resize(1, 7).Interior.ColorIndex = 4

        Else

            `Ponemos un cero en la columna 7
            Cells(ThisRow, 7).Value = 0

            `Pintamos la fila de rojo
            Cells(ThisRow, 1).Resize(1, 7).Interior.ColorIndex = 3

        End If

    Next ThisRow

End Sub
```

En SEGUNDOS, esta macro analiza más de 16 mil registros en Excel, reorganiza los valores, y les aplica un color como se muestra en la siguiente imagen

	A	B	C	D	E	F	G
1	Vendor	Vendor Name	Date	Qty	Type	Amount	
2	558968	Pepito Perez	31/08/2020		Servicios		691.669,00
3	793561	Melva Perez	31/08/2020		Servicios		823.440,00
4	395210	Jesusita Perez	01/09/2020	22	Producto	256.553,00	
5	551743	Ruperta Perez	01/09/2020	33	Producto	110.272,00	
6	395210	Jesusita Perez	02/09/2020	62	Producto	455.825,00	
7	551743	Ruperta Perez	02/09/2020	12	Producto	31.870,00	
8	793561	Melva Perez	03/09/2020		Servicios		96.453,00
9	551743	Ruperta Perez	03/09/2020		Servicios		803.013,00
10	395210	Jesusita Perez	04/09/2020		Servicios		559.334,00
11	278486	Perez de Osa	04/09/2020	52	Producto	17.979,00	
12	793561	Melva Perez	04/09/2020	52	Producto	149.253,00	
13	186955	Marucha Perez	05/09/2020	50	Producto	212.451,00	
14	83719	Benilda Perez	05/09/2020	54	Producto	336.569,00	
15	186955	Marucha Perez	05/09/2020	50	Producto	77.765,00	
16	558968	Pepito Perez	06/09/2020	5	Producto	676.144,00	
17	278486	Perez de Osa	06/09/2020		Servicios		802.403,00
18	793561	Melva Perez	07/09/2020		Servicios		820.136,00
19	83719	Benilda Perez	07/09/2020		Servicios		258.917,00
20	580968	Gumercinda Perez	08/09/2020		Servicios		660.588,00
21	580968	Gumercinda Perez	08/09/2020		Servicios		359.663,00
22	278486	Perez de Osa	09/09/2020		Servicios		840.971,00
23	558968	Pepito Perez	09/09/2020	96	Producto	532.958,00	
24	558968	Pepito Perez	09/09/2020	65	Producto	536.935,00	
25	580968	Gumercinda Perez	10/09/2020	59	Producto	392.858,00	
26	551743	Ruperta Perez	10/09/2020	79	Producto	759.821,00	
27	83719	Benilda Perez	11/09/2020		Servicios		410.314,00
28	186955	Marucha Perez	11/09/2020		Servicios		249.314,00
29	186955	Marucha Perez	12/09/2020		Servicios		677.874,00
30	395210	Jesusita Perez	12/09/2020		Producto	625.203,00	
31	580968	Gumercinda Perez	13/09/2020		Producto	27.324,00	
32	83719	Benilda Perez	13/09/2020		Producto	95.087,00	
33	278486	Perez de Osa	14/09/2020		Producto	788.549,00	

Ojo, bajo esta técnica CELLS tenga cuidado cuando se eliminen filas

Bueno, y que si queremos en lugar de copiar una celda, eliminar la fila completa? Pues aquí hay una ocurrencia extraña que pasa cuando en un recorrido (Loop) se elimina la fila. La fila que esta abajo y se elimina no se evalúa....

Supongamos que tenemos una macro que recorre desde la fila 1 a la 10 Digamos que de acuerdo al contenido de la celda, la cuarta fila se borra. Según lo que hemos visto, se haría con este código de macro:

```
Cells(ThisRow, 1).EntireRow.Delete
```

Recuerde que Entirerow.Delete significa eliminar o borrar la fila entera
 Resulta que al eliminar la fila 4 en Excel, toda la información de la fila 5 se mueve y pasa a ser la fila 4. Sin embargo, en el próximo registro revisará la fila 5, que era la 6. Como el tema se enreda, lo importante es que se soluciona haciendo que la macro corra de atrás para adelante. O mejor dicho, desde abajo hacia arriba, empezando por el último registro. Esto se haría reversando el orden de las

filas en la declaración FOR, y especificando que el Paso (Step) es -1. Recuerde que con Step 1, va bajando de a un registro. Pues con Step -1, va subiendo de a un registro.

El Código podría ser:

```
Sub BorrarFilasQueNoMegustan()
`Borro las filas que contienen Q4

    For ThisRow = 16171 To 2 Step -1
        If Cells(ThisRow, 4) = "Q4" Then
            Cells(ThisRow, 1).EntireRow.Delete
        End If
    Next ThisRow

End Sub
```

Trabajando con un número desconocido de Filas

Resulta que no todas las veces el archivo sobre el cual corre la macro tiene la misma cantidad de filas.... Sería "muy de buenas"..... en la realidad, esto no ocurre. Por lo general los reportes varían en cantidad de registros, y tener que estar cambiando el código

```
For ThisRow = 2 To 16171 Step 1
```

Hasta la fila "To 25645", por decir algo, sería una cosa muy aburridora, además de que no tiene presentación ni sentido.

Hay un truquito que se puede usar para trabajar en diferentes cantidades de registros según los archivos. Imagine que usted va hasta la última fila de la columna A. Es decir, si está en Excel 2003, se ubicaría en la celda A65536. Si está en 2007 se ubicaría en la celda A1048576. Estando allí, Presione la tecla END, y luego la flecha hacia arriba. Excel sale disparado y se "estrella" contra el último registro. Esta acción se puede grabar en una macro de varias formas, y la última celda hacer que Excel "La tenga presente", es decir, "Que no la olvide", guardando la dirección de esa ultima celda en una variable, así:

En Excel 2003 podría usar:

```
UltimaFila = Cells(65536, 1).End(xlUp).Row
```

En Excel 2007 podría usar:

```
UltimaFila = Cells(1048576, 1).End(xlUp).Row
```

Si no sabe en que versión de Excel correrá la macro, puede colocar este código:

```
UltimaFila = Cells(Application.Rows.Count, 1).End(xlUp).Row
```

Trabajando con Otras Hojas de Cálculo

Por lo general las macros corren sobre una hoja de cálculo activa; sin embargo, algunas veces se necesita mover información desde la hoja de cálculo activa hacia otra hoja.

No	Descripción	Numeral	Trimestre	Valor
1	uno	a	Trim1	\$ 100
2	dos	b	Trim2	\$ 200
3	tres	c	Trim3	\$ 300
4	cuatro	d	Trim4	\$ 400
5	cinco	e	Trim1	\$ 500
6	seis	f	Trim2	\$ 600
7	siete	g	Trim3	\$ 700
8	ocho	h	Trim4	\$ 800
9	nueve	i	Trim3	\$ 900
10	diez	j	Trim4	\$ 1.000
11	once	k	Trim1	\$ 1.100
12	doce	l	Trim2	\$ 1.200
13	trece	m	Trim1	\$ 1.300
14	catorce	n	Trim2	\$ 1.400
15	quince	o	Trim3	\$ 1.500
16	diesiseis	p	Trim4	\$ 1.600
17	diesisiete	q	Trim3	\$ 1.700
18	diesiocho	r	Trim4	\$ 1.800

Digamos que usted tiene cuatro hojas de cálculo adicionales en las cuales quiere guardar la información por trimestres, en hojas separadas, que se llaman Trim1, Trim2, Trim3, Trim4. Si se quiere revisar cada registro de la hoja activa y copiarlo hacia la hoja del trimestre correspondiente.

Para referirse a la celda J2 en una hoja llamada hoja3, puede utilizar:

```
Worksheets("hoja3").Cells(2, 10)
```

Si el nombre de su hoja, está guardada en una variable llamada ThisTrim, puede utilizar algo así como esto:

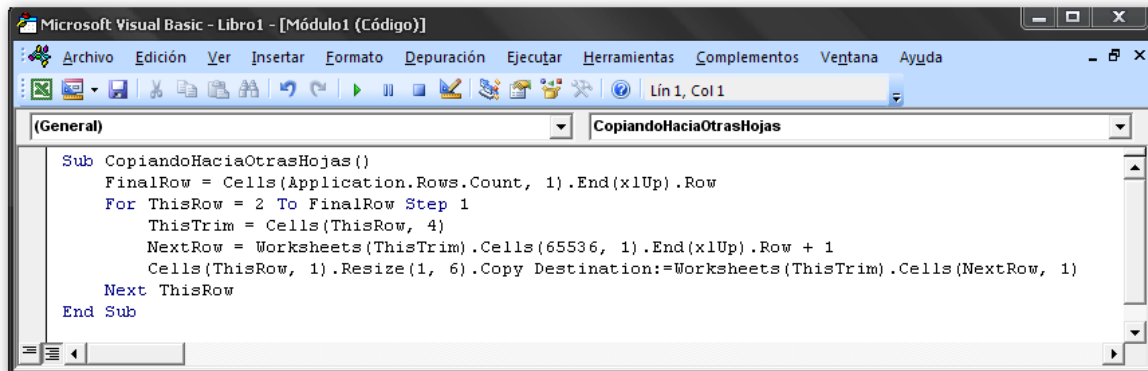
```
Worksheets(ThisTrim).Cells(2, 10)
```

Ok. La siguiente macro revisará cada registro de la cuarta columna y chequeará a que trimestre corresponde. Le asigna el periodo a una variable llamada ThisTrim. La macro buscará en la siguiente fila en la hoja del trimestre apropiada y copiará el registro en esa hoja.

```

Sub CopiandoHaciaOtrasHojas()
    FinalRow = Cells(Application.Rows.Count, 1).End(xlUp).Row
    For ThisRow = 2 To FinalRow Step 1
        ThisTrim = Cells(ThisRow, 4)
        NextRow = Worksheets(ThisTrim).Cells(65536, 1).End(xlUp).Row + 1
        Cells(ThisRow, 1).Resize(1, 6).Copy
        Destination:=Worksheets(ThisTrim).Cells(NextRow, 1)
    Next ThisRow
End Sub

```



Recorriendo todas las hojas de cálculo

Hay una clase especial de recorrido o Loop conocida como "For Each" Un libro por lo general tiene una colección de hojas. Una hoja contiene una colección de filas. Las filas pueden contener una colección de cuentas.

Para recorrer todos los objetos de la colección, puede usar algo como esto:

```

For Each VariableName in Collection
    `Hacer algo
Next VariableName

```

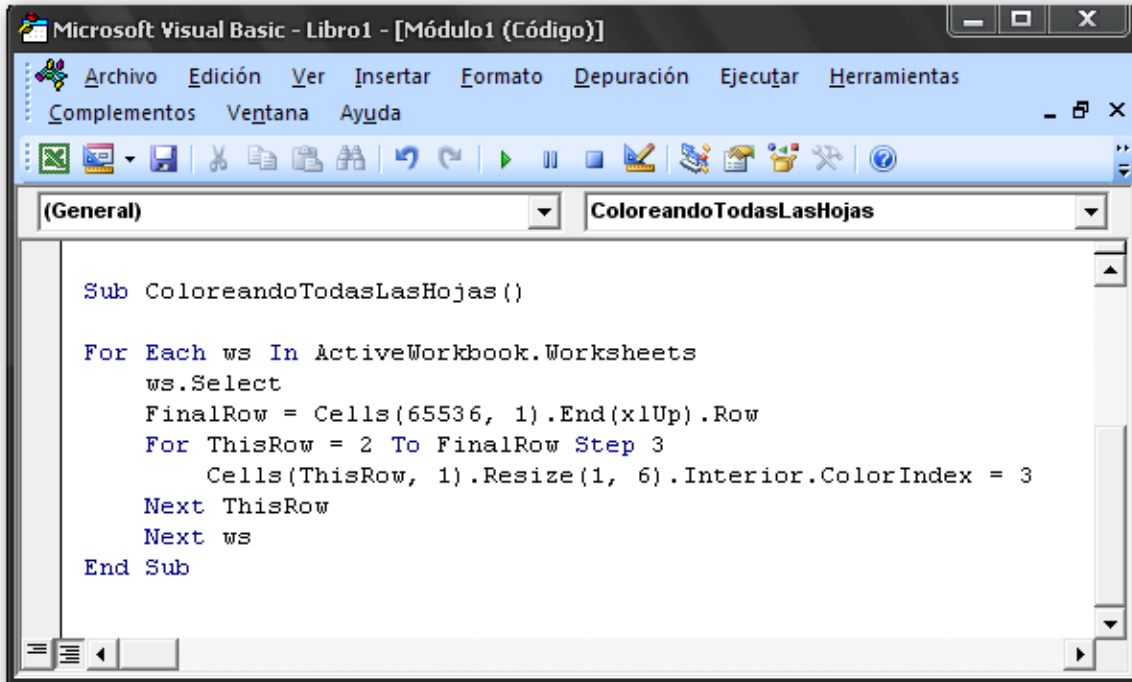
Por ejemplo, si le queremos aplicar un formato a las filas de todas las hojas de cálculo, rojo, blanco, blanco, del libro podría usarse un código como este:

```

Sub ColoreandoTodasLasHojas()

For Each ws In ActiveWorkbook.Worksheets
    ws.Select
    FinalRow = Cells(65536, 1).End(xlUp).Row
    For ThisRow = 2 To FinalRow Step 3
        Cells(ThisRow, 1).Resize(1, 6).Interior.ColorIndex = 3
    Next ThisRow
Next ws
End Sub

```

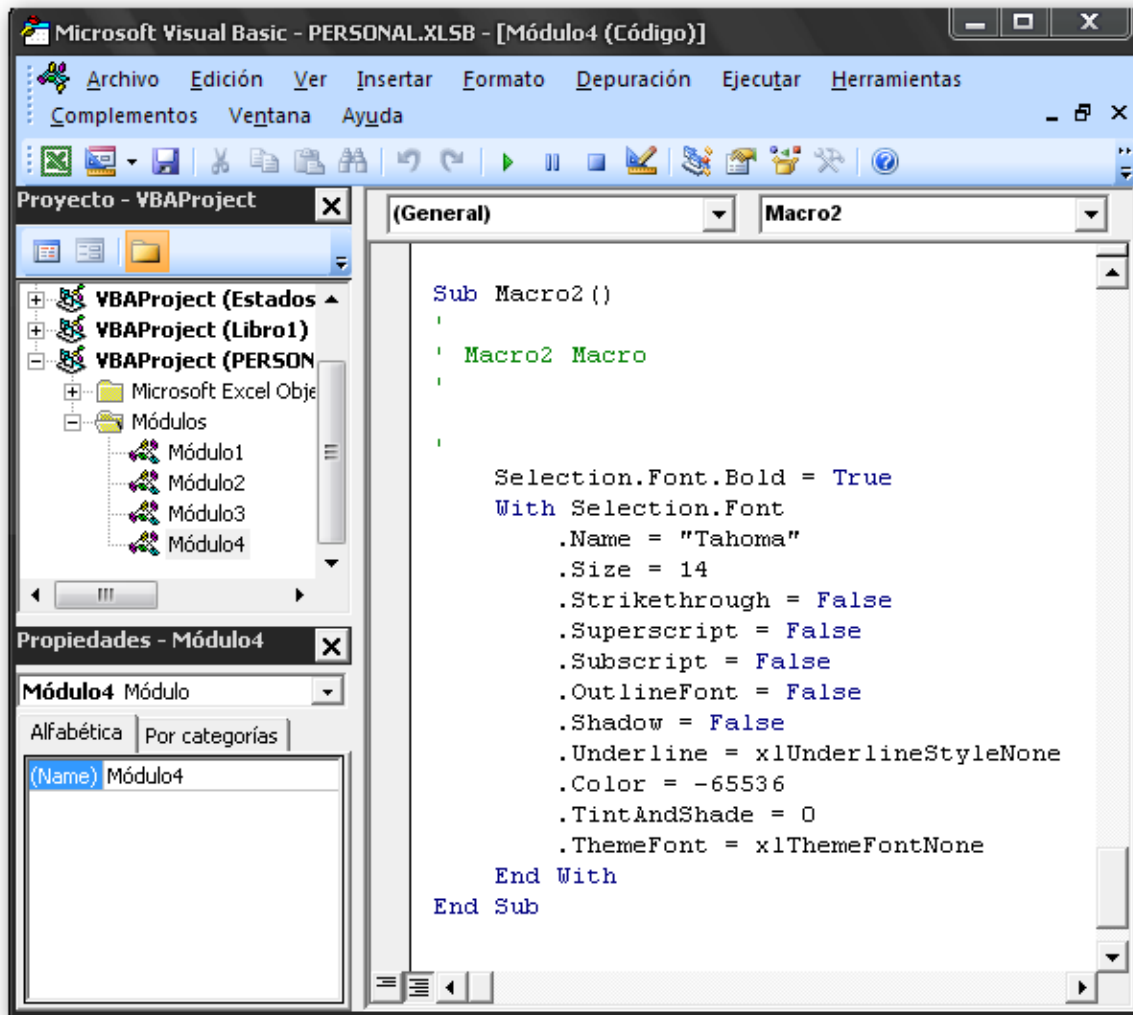
Usando el grabador de macros para aprender otro código

Hemos visto como colorear las filas de verde o de rojo...Pero y si se quisiera pintar las celdas de azul, ponerle negrilla, tipo de letra tahoma, de tamaño 14? Yo no conozco el código para hacer eso. Usted tampoco. ¡La mayoría de los programadores tampoco lo conoce! Pero como se hace?

Prenda el grabador de macros, en Excel, dele el formato deseado a una celda. Detenga la grabación de la macro, y examine el código...

Haga lo siguiente:

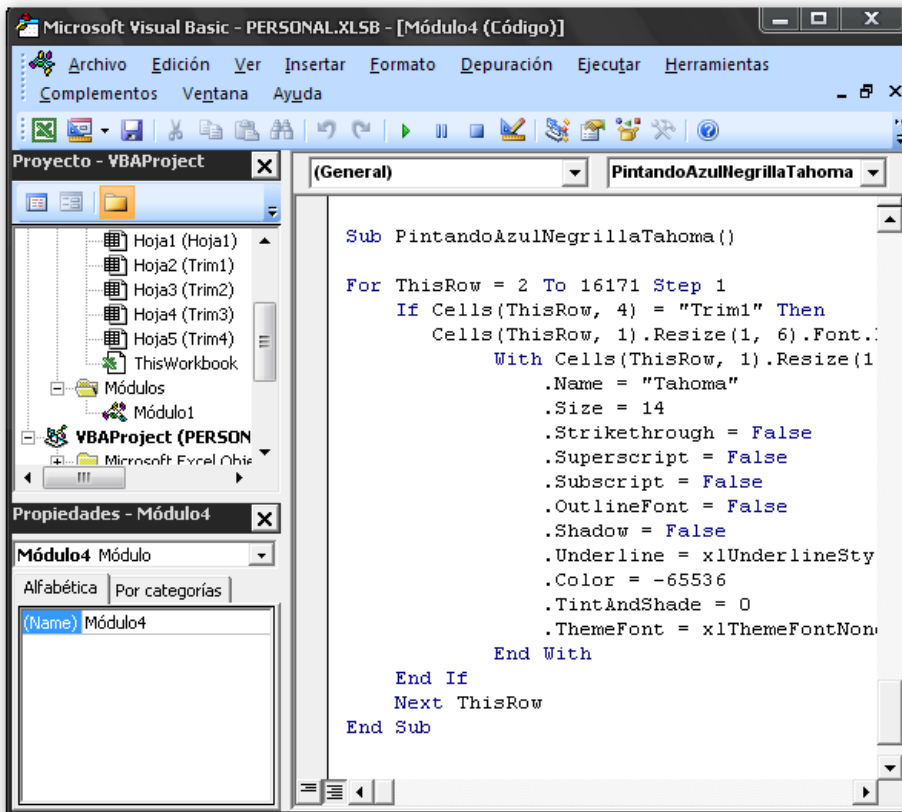
1. Ya sabe que si se encuentra en Excel 2003 va por el menú Herramientas – Macro – grabar nueva macro. Si es 2007, Ficha Programador – Código – Grabar Macro.
2. La ventana de diálogo de la macro le pedirá un nombre, una combinación de letras y descripción. Para lo que buscamos, no hace falta, porque no estamos armando una macro como tal, solo queremos que el grabador de macros escriba el código que no conocemos y que necesitamos ver.
3. Use los comandos de formato para cambiar la celda corriente en negrilla, azul, tahoma, 14.
4. Detenga el grabador de macros.
En Excel 97-2003: Presione el botón Stop en la mini barra flotante.
En Excel 2007: Haga clic en la barra de estatus o deténgala con el botón Stop Recording en la ficha Programador.
5. Llame el editor de Visual Basic con Alt + F11
6. Excel siempre coloca el código en un modulo nuevo. En el panel de proyectos, verá un nuevo modulo, de doble clic allí para abrir el panel de código. Allí verá que un código de la macro similar a esto:



Ahora que conoce el código de la macro grabada, puede cortar y pegar esta sección e incorporarla en la macro que esté armando.

Algo que debe tener en cuenta es que el grabador de macros siempre trabaja haciendo una selección fija de cada celda. En la vida real las macros corren mucho más rápido si no las obliga a seleccionar cada fila. Cambie la Selección de la sintaxis del comando CELLS, como se muestra a continuación:

```
Sub PintandoAzulNegrillaTahoma()  
  
For ThisRow = 2 To 16171 Step 1  
  If Cells(ThisRow, 4) = "Trim1" Then  
    Cells(ThisRow, 1).Resize(1, 6).Font.Bold = True  
    With Cells(ThisRow, 1).Resize(1, 6).Font  
      .Name = "Tahoma"  
      .Size = 14  
      .Strikethrough = False  
      .Superscript = False  
      .Subscript = False  
      .OutlineFont = False  
      .Shadow = False  
      .Underline = xlUnderlineStyleNone  
      .Color = -65536  
      .TintAndShade = 0  
      .ThemeFont = xlThemeFontNone  
    End With  
  End If  
Next ThisRow  
End Sub
```



Lo que hay que tener en cuenta del grabador de macros

El grabador de macros es muy útil para ver como es que se escribe el código que necesitamos. Sin embargo, la cosa tampoco es cuestión de prender el grabador de macros, hacer todo lo que se suele

hacer en la hoja de cálculo, apagar el grabador, abrir un archivo en donde queremos repetir todo otra vez, ejecutar la macro grabada y esperar que todo salga bien.

Como se le indicaba hace un momento, (en el apartado anterior), cuando usted selecciona, por ejemplo, la ultima celda en su hoja de calculo, el Visual Basic graba que seleccionó esa celda de manera fija, (por ejemplo la ultima celda de la columna A en un archivo puede ser la A22356, y en otro archivo puede estar mas abajo, en la celda A34750) y usted probablemente necesite es que se vaya hasta la ultima celda, PERO de la hoja actual, de la hoja en la cual está corriendo la macro, no del archivo sobre el cual grabó la macro.

Ir hasta la última celda con información:

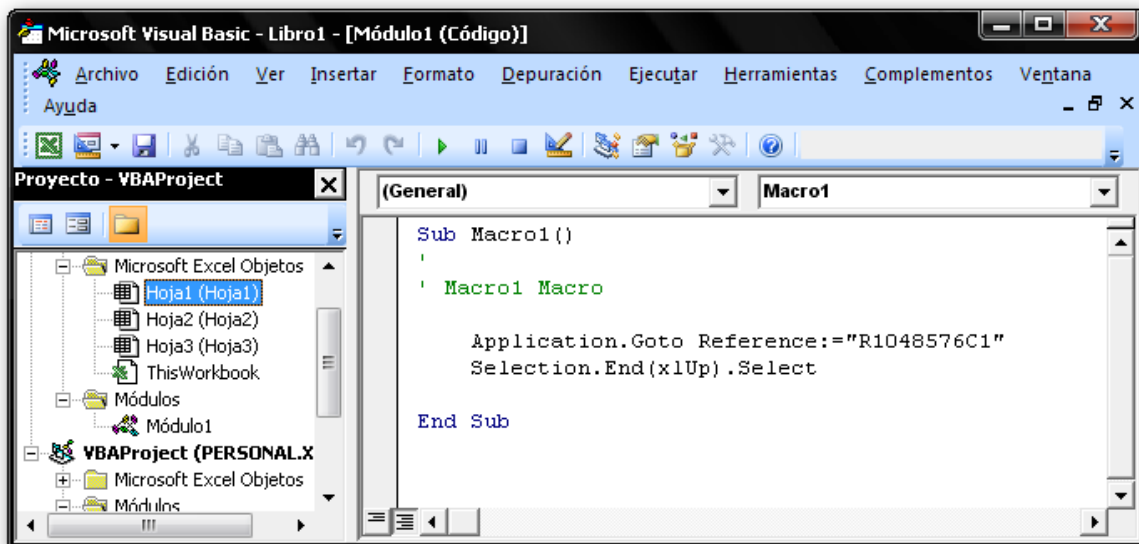
Lo que tiene que tener presente es, que siempre que necesite ubicarse en la última celda de su hoja, utilice el código que se le enseñó en el apartado anterior For Next:

```
UltimaFila = Cells(Application.Rows.Count, 1).End(xlUp).Row
```

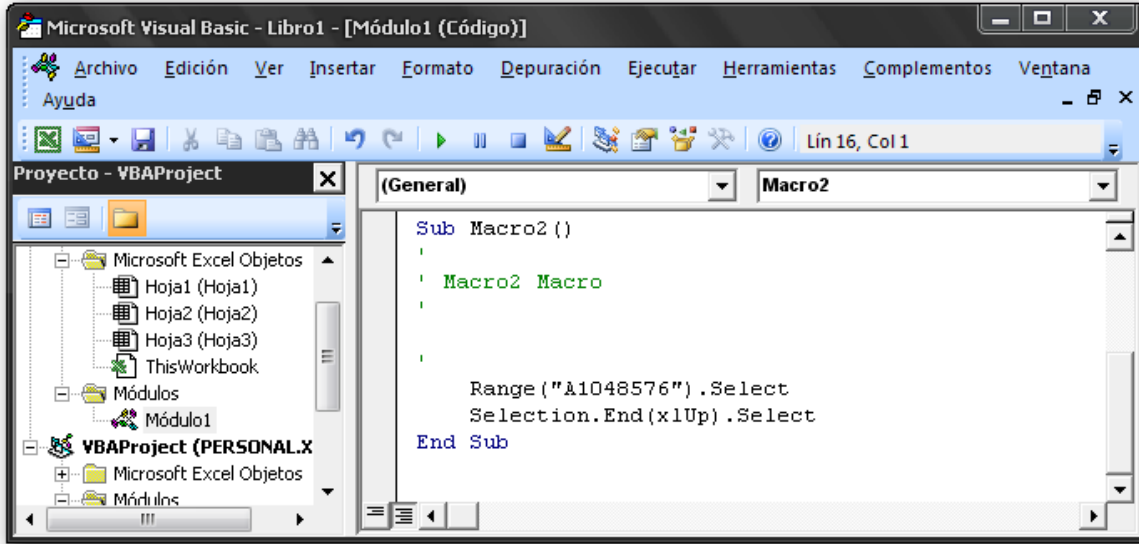
En este caso, de una vez, se le dice a Excel que tenga en cuenta, que la ultima fila es la que va a encontrar si se va hasta la ultima fila de la primera columna, esto es, la 65536 si está en Excel 2003 o la 1048576 si está en 2007 y suba, hasta que se estrelle con la ultima celda que contenga datos.

Así, cada vez que necesite que la macro vaya a la última fila, pues refiérase a la variable UltimaFila.

Para entender esto mas fácilmente suponga que si se grabara, con el grabador de macros, la operación de ir hasta la ultima celda de la columna A, de Excel, (esta operación la haríamos con la ficha Inicio el botón de buscar y seleccionar, Ir A, se le escribe "A1048576" y luego con la tecla Fin + flecha arriba), subir hasta la ultima celda con datos, quedaría grabado esto:

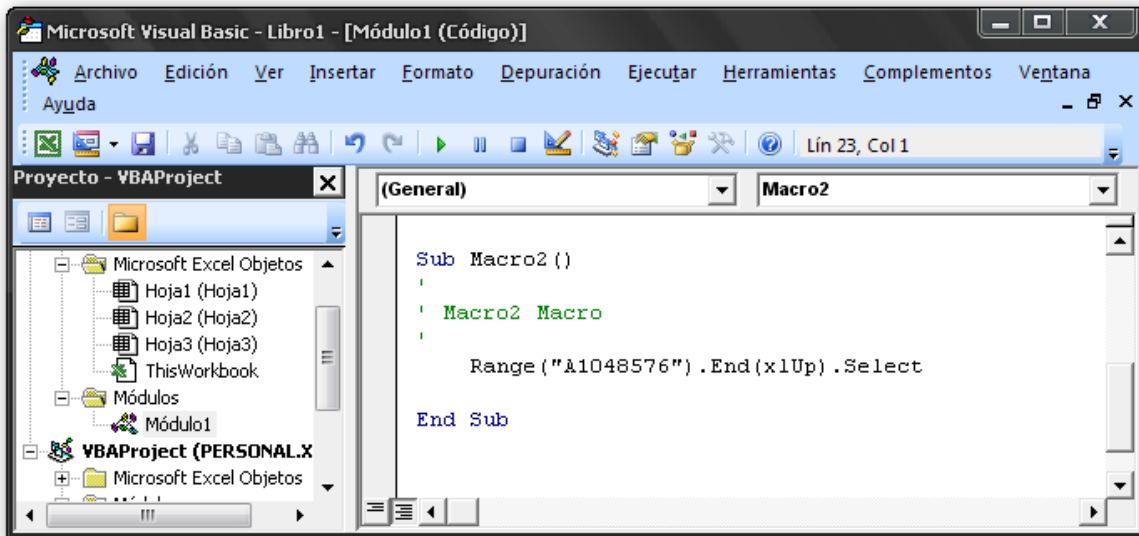


O podríamos seleccionar la última celda directamente y luego subir, quedaría grabado esto:



Cualquiera de las dos formas, da lo mismo.

Ahora, el grabador de macros tiende a ser muy redundante, y si le quitamos tanta redundancia, podemos hacer que la macro, al ser mas corta, pues se tarde menos. (Aunque a veces no se nota) Una regla que se deduce es que no hay necesidad de obligar a Excel de que vaya hasta las celdas seleccionándolas, por ejemplo, para ir hasta la ultima celda de nuestra hoja de calculo, NO LA ULTIMA CELDA DE EXCEL, pues no hay necesidad de seleccionar la ultima celda de Excel y hacerlo subir. Lo que grabó el grabador de macros, se puede editar, es decir, corregirle la escritura (el código) y dejarlo así:



Si se fija, podrá darse cuenta que estas dos sentencias o palabritas, Select y Selection, las seleccioné y luego las eliminé:

```

Sub Macro2 ()
'
' Macro2 Macro
'
'
Range("A1048576").Select
Selection.End(xlUp).Select
End Sub

```

Y si se fija, verá que allí hay una redundancia y no hay necesidad de que Excel vaya hasta la última celda del Excel, para ir hasta la última celda de información.

Ok. Debe saber que mejorando el código ni siquiera hace falta que Excel seleccione la última fila de su reporte, como se indicaba anteriormente, esta instrucción no obliga a Excel a ir hasta la última celda, simplemente solo se necesita referenciarla:

```
UltimaFila = Cells(Application.Rows.Count, 1).End(xlUp).Row
```

Puede ver que este ejemplo por ninguna parte aparece el "Select". Esto se traduce en mucha velocidad de ejecución. La instrucción "Application.Rows.Count" sirve independientemente de que tipo de Excel, 2003 o 2007 esté usando.

Desplazarnos a partir de una celda sin necesidad de seleccionarla:

Como le mencionaba hace un momento, cuando grabamos con el grabador de macros, pues está el problema que cada vez que seleccionamos una celda en Excel, pues en el código queda seleccionada esa celda de manera fija, y eso es un problema, pues en ocasiones necesitamos ubicarnos en cierta posición y a partir de allí movernos hacia arriba, hacia abajo, hacia alguno de los lados de esa celda, pero al quedar seleccionado, pues no nos queda bien, por la razón de que la ubicación en otro archivo similar, quedará en otra posición y, si la macro graba que la posición es una posición fija, pues correrá mal.

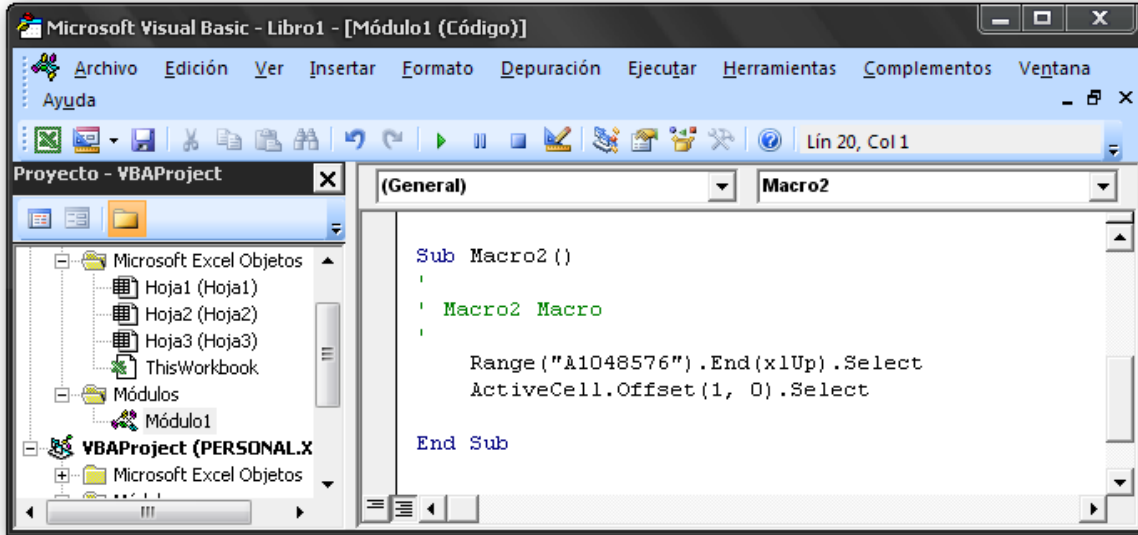
Por ejemplo, ya sabemos como ir hasta la ultima celda con datos de nuestro archivo, haciendo que Excel tome por referencia la última celda de la hoja de cálculo, pero, ¿Y si queremos que una vez posicionados en esa ultima celda, desplazarnos hacia la derecha, izquierda, arriba, o abajo?

Por ejemplo, si queremos ir hasta la ultima celda SIN información de nuestro archivo, necesitamos bajar a la fila que está abajo, pues la ultima celda tiene información, y no queremos hacerle nada, solo hacer otra cosa pero en la celda de abajo que sabemos que está limpia, sin datos, lo haríamos con esta instrucción:

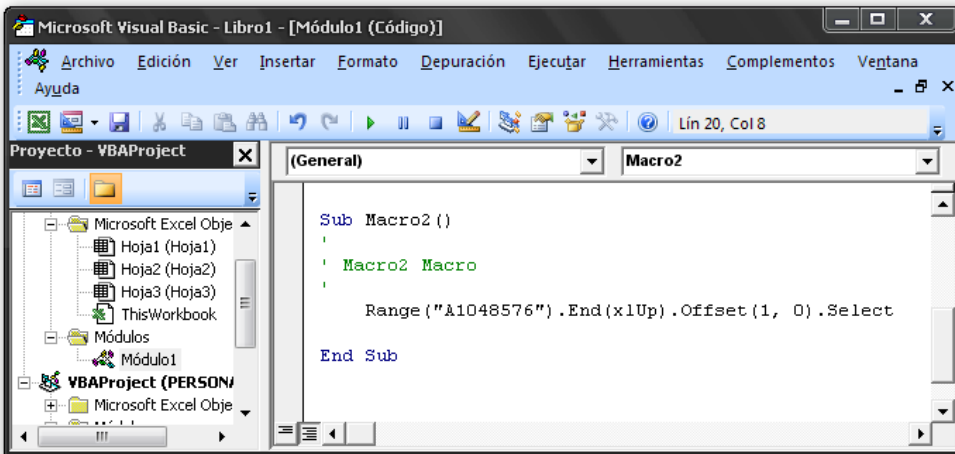
```
ActiveCell.offset(1,0).Select
```

(Que significaría literalmente algo así como CeldaActiva.Desvíese(1,0).Selecione)

Y pegando esta instrucción al código de seleccionar la última celda con información pero bajar a la siguiente fila que sabemos que está en blanco, quedaría así:



Ahora, nuevamente se puede "pegarle" una "depuradita" al código, pues nuevamente, tampoco hay necesidad de hacer que Excel seleccione la ultima fila, y luego que seleccione -otra vez- la que está una fila abajo.... Podemos direccionarlo de una sola vez, hasta la celda destino que queremos, y el código depurado quedaría así:



Como se puede depurar o mejorar el código escrito?

Con la práctica. Probando, ensayando, observando, compenetrándose más con el código Visual Basic. Leyendo más libros avanzados sobre el tema. Pero que esto no constituya un desaliento, por el contrario una motivación más: Debe saber que igual, Excel es muy rápido, y lo importante es que su macro funcione. Poco a poco, con el tiempo, irá mejorando sus macros. Por ejemplo, recuerde el problema que se le puede presentar si en una macro monta un buscarV, y la base de datos es demasiado grande, use Índice con Coincidir, como explico en el anexo correspondiente de **"La falencia del BuscarV"**. Una situación que le representará a Usted una mejora en la escritura de sus macros, será cuando advierta que no hay necesidad de hacer que Excel seleccione las celdas. Por ejemplo, imagínese que yo le digo a usted que vaya al sur de la ciudad y para eso lo hago ir al polo sur del planeta, y luego que suba hasta que se encuentre con su ciudad. Si le digo que es subiendo por el sur hasta su ciudad, usted ya no tendrá necesidad de ir, solo "VER" desde la distancia, para llegar al sur. Así, se ahorra TODA ESA VUELTA.

Eso mismo es lo que se debe propender por hacer al escribir las macros, para que le queden mas legibles, dinámicas, y rápidas. La selección se debe procurar hacer solo cuando realmente se necesite y a veces ni eso es necesario, o solo en el punto de llegada, o cuando necesite almacenar una variable.... Una razón de peso para evitar escribir macros así, es que cuando otras personas ven el código de las macros, al ejecutarlas paso a paso entienden que hacen, pues la celda en la hoja de calculo se va moviendo paso a paso, va "delatando" que hace su macro. Caso contrario ocurre si logra escribir su código sin seleccionar las celdas, así, cuando "chismoseen" el código de sus macros.... Tendrán algunos problemas para entender como corre y que hace la macro que Usted escribió. Por ejemplo Volvamos a la primera macro que vimos cuando explicamos el truco de "Rellenar Celdas en blanco": El código era así:

```
Sub Copiar()
  Do Until ActiveCell = "Fin"
    If ActiveCell = "" Then
      Selection.FillDown
    End If
    ActiveCell.Offset(1, 0).Select
  Loop
End Sub
```

Por ejemplo, si en lugar de hacer el Do Until quisiéramos escribirla al estilo de For Next, que vimos al inicio de las macros, la principal mejora es que Excel no necesita seleccionar paso a paso la celda e irse desplazando. El código, cuando se tiene clara cual es la columna que se debe copiar, quedaría así: Al ejecutarla paso a paso, solo verá como se mueven las celdas, pero por ninguna parte verá que se seleccionan.

```
Sub Copiar2()

  Finish = Cells(Application.Rows.Count, 1).End(xlUp).Row
  For ThisRow = 2 To Finish Step 1
    If Cells(ThisRow, 1) = "" Then
      Cells(ThisRow, 1).FillDown
    End If
  Next ThisRow

End Sub
```


Algo más sobre la instrucción ActiveCell.Offset

Tenga en cuenta que cuando se hace un desvío se hace desde la celda activa, la celda activa no es más que la celda por donde se está moviendo o se va ubicando la celda a medida que se desarrollan los pasos.

Como puede ver en el ejemplo, moverse una celda abajo desde la celda activa se hace con esta instrucción:

```
ActiveCell.offset(1,0).Select
```

Dos celdas abajo, en lugar del UNO, le pones un DOS: (y así sucesivamente, tres, cuatro, veinte, las que necesites)

```
ActiveCell.offset(2,0).Select
```

Una columna a la derecha pero sin bajar de fila:

```
ActiveCell.offset(0,1).Select
```

Dos celdas a la derecha, en lugar del UNO, le pones un DOS: (y así sucesivamente, tres, cuatro, veinte, cincuenta, las que necesites)

```
ActiveCell.offset(0,2).Select
```

Conociendo ya el "truquito", ya puede moverse por donde quiera, una celda abajo y una a la derecha:

```
ActiveCell.offset(1,1).Select
```

Para ir a, una celda arriba, y una a la izquierda, desde la celda activa, le metes un MENOS UNO (-1):

```
ActiveCell.offset(-1,-1).Select
```

Ojo, si estas en la columna A, te saldrá un error, pues Excel no podrá moverse otra columna a la izquierda, PUES NO HAY a donde mas moverse. Lo mismo aplica para la última fila, la primera fila o ultima columna.

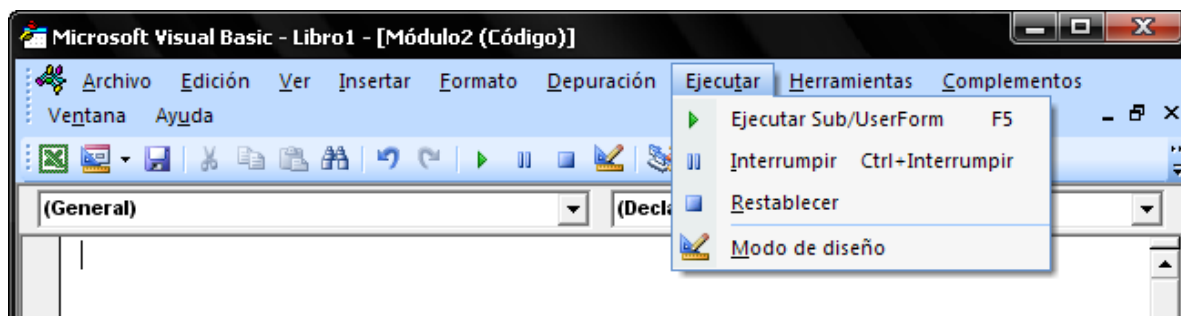
Una anécdota personal:

Cuando conocí estas dos instrucciones, posicionarme en la última celda, y moverme sin necesidad de seleccionar las celdas, ME LIBERÉ DE LAS LIMITACIONES DEL GRABADOR DE MACROS y pude ir un poco más allá haciendo que mis macros trabajaran hasta el final deseado, pues todas las demás transformaciones de mis archivos, cuadros o reportes, lo podía hacer con el grabador. Formulas, copiar, pegar, borrar, eliminar columnas, filas, poner formatos, ya todo lo demás lo hacía el grabador de macros. Solo tenía que entrar y ejecutar la macro paso a paso, y revisar, "Con lupa" si en un nuevo archivo mi macro trabajaba bien.

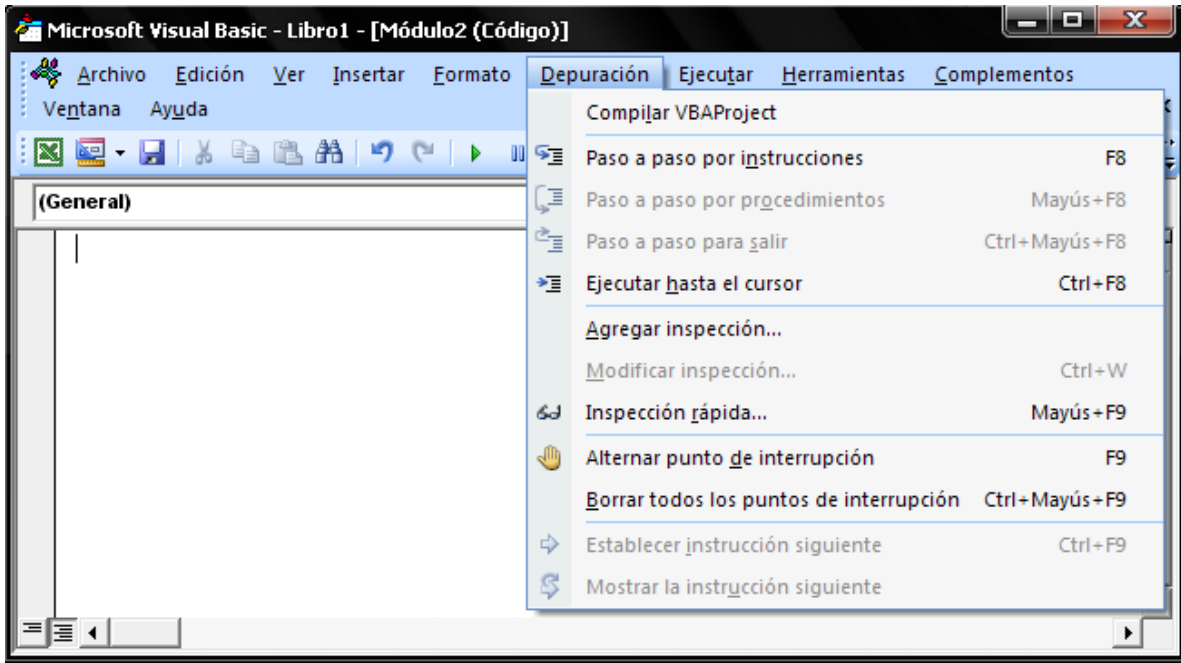
Ejecutar una macro paso a paso

Resulta que lo que yo hacía, porque no sabía, (pues la mayoría de los que nos metemos en este cuento no tenemos formación de Ingenieros de Sistemas, por lo cual la cosa se dificulta un poco), era correr la macro y cuando surgía un error, la macro se detenía y Excel me sacaba un mensaje de error y una de las opciones del mensaje era "Depurar". Entonces, yo le daba clic allí y Excel abría el Visual Basic en donde había grabado la macro y me llevaba al punto en el cual la macro "se troteaba". (Es decir, el punto en el cual la macro no quería seguir ejecutándose). Entraba y revisaba que era lo que pasaba, de manera muy intuitiva, hacia mi mejor esfuerzo, trataba de analizar porque no quería seguir corriendo, y por lo general eran instrucciones que Excel era incapaz de realizar, por ejemplo, le decía que se desviara dos columnas a la izquierda, y resulta que era imposible pues solo se podía desplazar solo una columna, y cosas así por el estilo... errores "pueriles" de este tipo, que no me da pena contar, pues en este mundo nadie nace aprendido, pero la voluntad de querer es la voluntad del poder, que genera la observación, la aprehensión y con ello la construcción del nuevo conocimiento.

Luego observando el menú del visual basic, vi que al lado de estas opciones que eran las únicas que yo manejaba, estas:

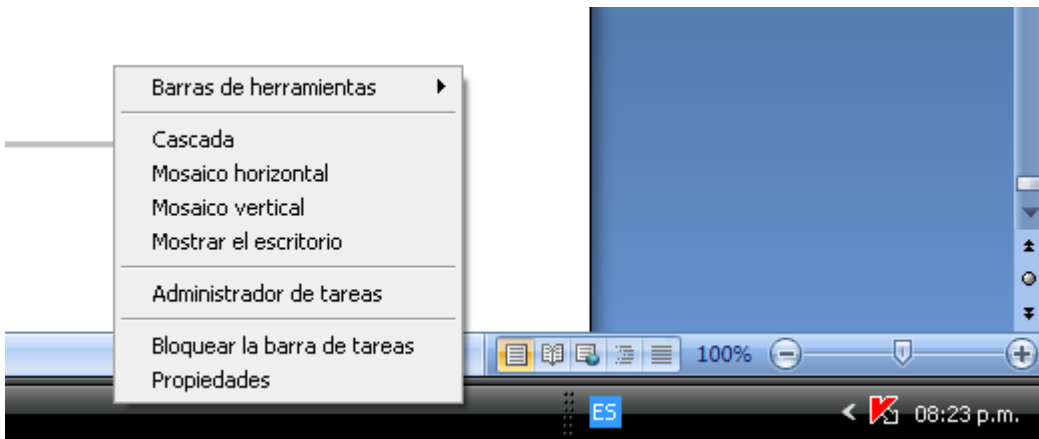


Había otra opción en el menú que decía "Depuración":



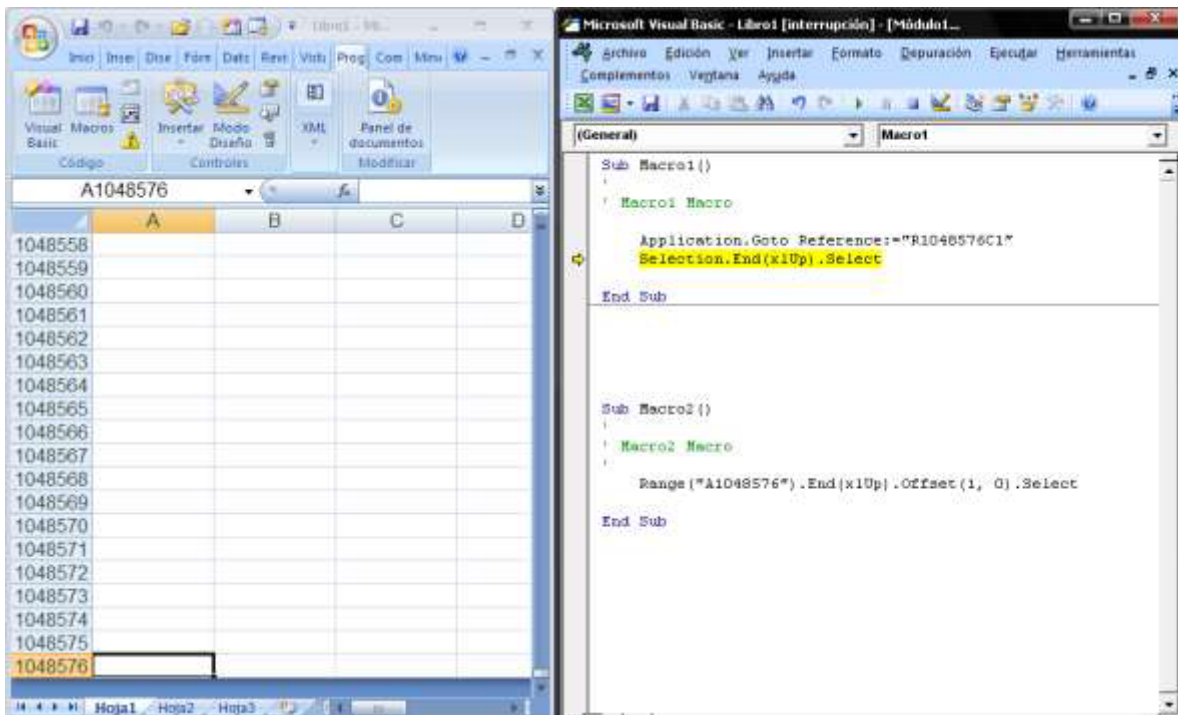
Y, **OH SORPRESA...!** había algo que decía "Paso a paso por instrucciones --- F8" y "Ejecutar hasta el cursor Ctrl+F8"

Resulta que, si Usted **DIVIDE** la pantalla de su computador en DOS, en uno de los lados coloca Excel, abre el Visual Basic, y lo acomoda en el otro lado, (o se va por esta opción:



Y escoge Mosaico horizontal o vertical, la pantalla se le divide en las ventanas que tenga activas)

Podrá visualizar Excel y el Visual basic al mismo tiempo, y si una de sus macros en lugar de ejecutarla, va oprimiendo la tecla F8 de su computador, verá como se van ejecutando las instrucciones, tal como se indica allí, paso a paso, instrucción por instrucción y por ende, las podrá analizar, estudiar y mejorar.



Ejemplo, aquí puede ver como al dar F8 en la macro1, puedo ver como en Excel el cursor bajó hasta la última celda y se resalta en amarillo la siguiente instrucción que se ejecutará. De esta manera, se puede ir analizando si lo que programamos en Excel funciona correctamente, o podemos depurar o mejorar el programa o instrucciones de nuestra macro.

Las variables.

Ok. Con las anteriores cositas, tales como hacer que Excel ubique la última celda y desplazarme sin necesidad de seleccionar las celdas pude hacer macros por lo menos funcionales, pero hubo un momento en que tuve necesidad de grabar una fórmula, y esta tomaba un rango que dependiendo del nuevo archivo en el que trabajara, podía variar de tamaño, ser de 10, 15 50, 200 filas.... Entonces me encontré con ese problemita... no sabía como hacerlo. Entonces, investigando y asesorándome, preguntando a mis amigos más *sabiondos*, supe que la solución era usar una variable, que para que Usted me entienda, es más o menos como decirle a Excel que "ojo, tenga en cuenta la celda TAL, que mas adelante que arme una suma, tendrá que ir desde tal celda, hasta la celda TAL".

Hace un momento, vimos el ejemplo de como definir mediante variable la última celda de Excel, esta instrucción:

```
UltimaFila = Cells(Application.Rows.Count, 1).End(xlUp).Row
```

Técnicamente hablando, las variables se usan en todos los lenguajes de programación para almacenar datos en un espacio de la memoria temporal del sistema de la computadora. El dato que se almacena se usa mientras se ejecuta la macro.

Ejemplo:

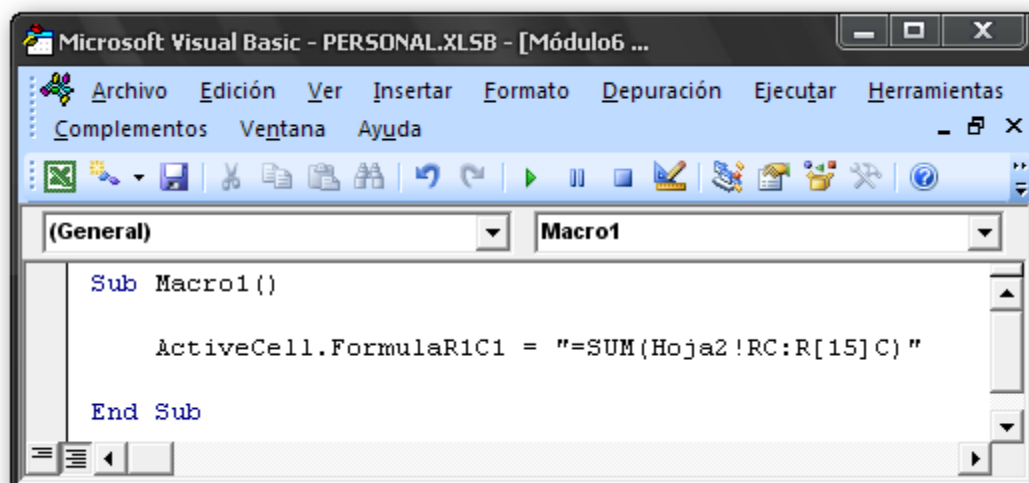
Supongamos que estamos en una hoja1 y en cierto lugar, queremos sumar el rango de la columna A, de la hoja2:

	A	B	C	D
1	856790			
2	872349			
3	648063			
4	970987			
5	377629			
6	392558			
7	628839			
8	512308			
9	624831			
10	440423			
11	706998			
12	286442			
13	918122			
14	280771			
15	637153			
16	596700			
17				
18				

En la hoja1, quisiéramos hacer la suma de ese rango:

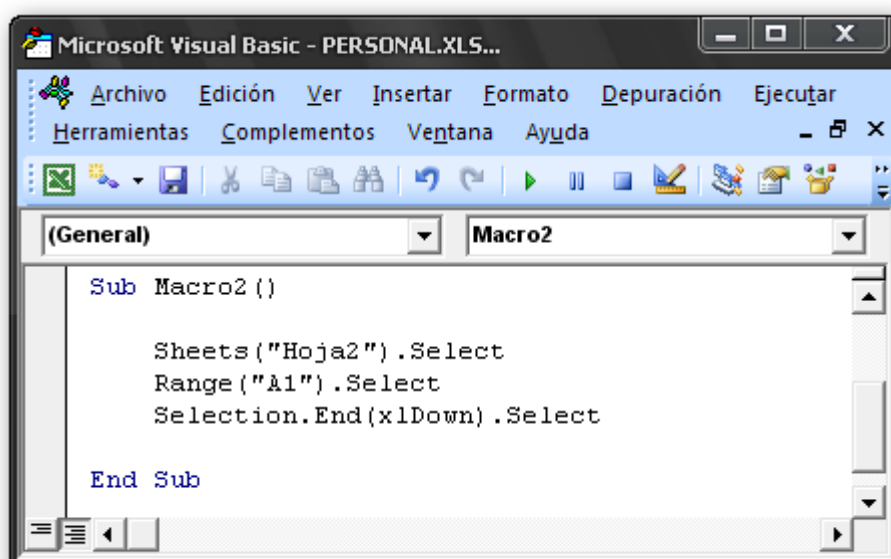
	A	B	C	D	E
1	9750963				
2					
3					
4					

Pero resulta que la hoja2, en otro archivo que nos envíen, puede contener más o menos valores, ser más o menos larga... si grabáramos con el grabador de macros, el código nos quedaría así:

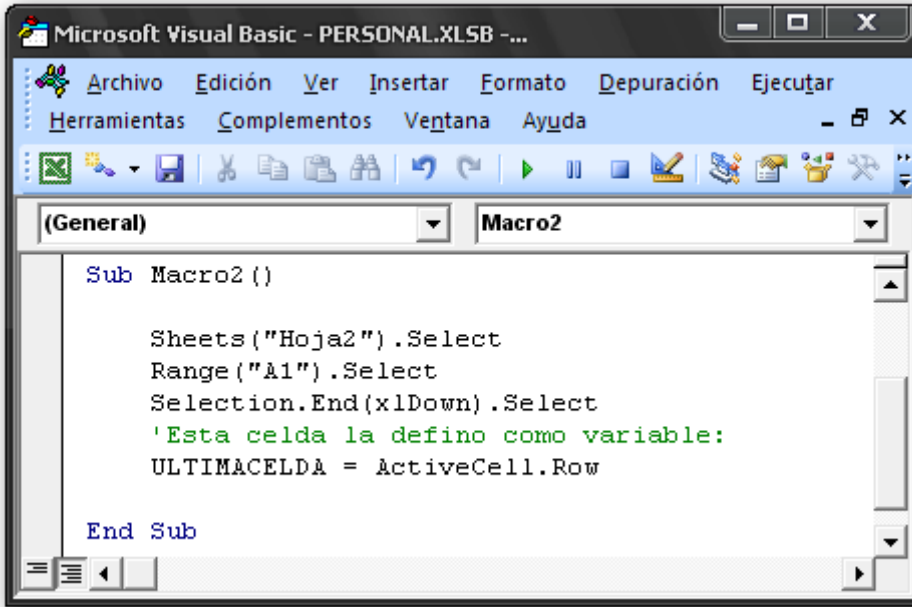


Pero si corriéramos la macro en un nuevo archivo en donde el rango no es de 15 filas sino de cincuenta.... GRAVE, pues la macro quedó con el rango Fijo de 15 filas R[15] (Row-fila 15, es decir, 15 filas).

Lo que podemos hacer es definir en una variable la última celda de la hoja2. Para esto, me sirvo del grabador de macros para que haga por mí, una parte del código que necesito, y luego lo "AFINO": Entonces: Prendo el grabador de macros, selecciono la hoja2, selecciono la celda A1, oprimo la celda FIN + flecha abajo. Excel selecciona la última celda. Detengo la macro. Esto es lo que queda grabado:



En este punto "LE METO LA MANO" (es decir, edito el código por mi mismo) al código, y le digo que ESA ULTIMA CELDA EN LA QUE ME DETUVE ME LA TENGA EN CUENTA COMO LA ULTIMA CELDA:

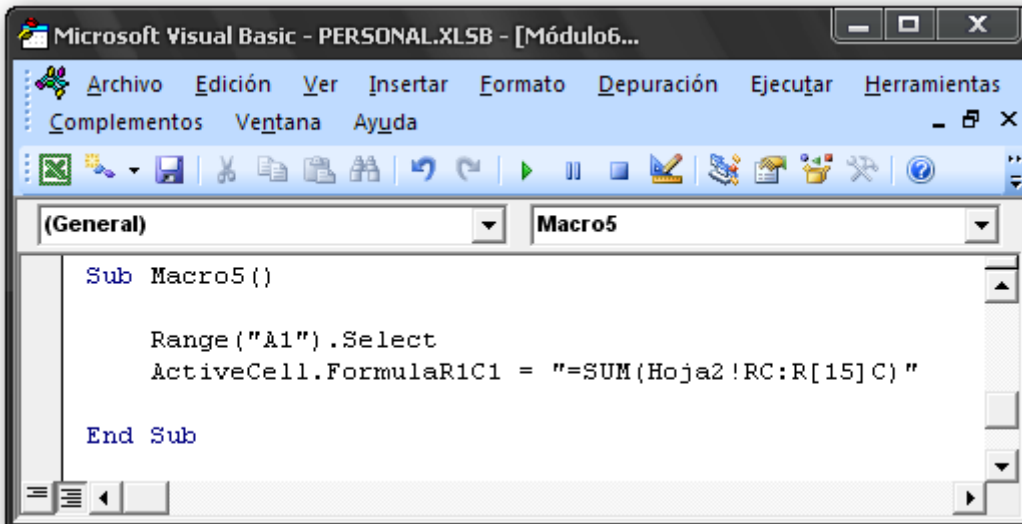


Recuerde que cuando le mete un apóstrofo, puede hacer comentarios para que recuerde mejor que es lo que hace la siguiente instrucción. Cuando lo hace, Excel automáticamente le da color verde.

Ojo, Apréndase esa palabrita ActiveCell.Row (CeldaActiva.Fila)

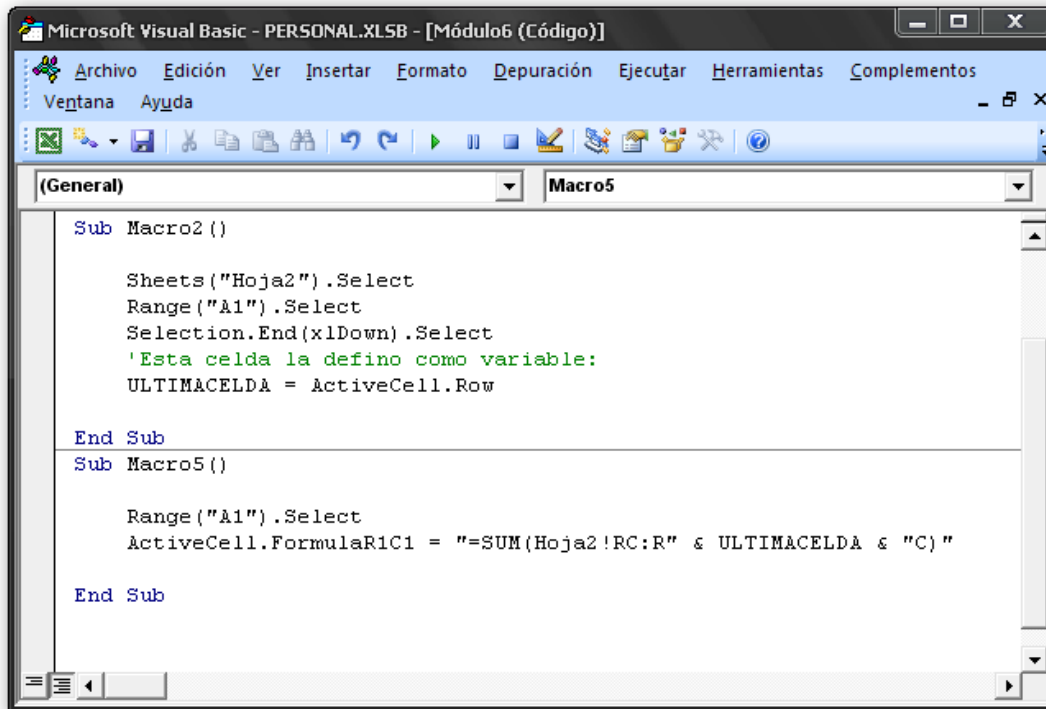
Listo, ahora la fórmula que escribí en la otra macro, la Sub Macro1(), la puede escribir diciéndole, NO que se vaya hasta la fila 15, SINO que se vaya hasta la última fila activa, que hemos llamado ULTIMACELDA:

Prendo nuevamente el grabador de macros, me posiciono en donde quiero que quede mi fórmula, y escribo la fórmula con el grabador de macros, (luego la arreglo), así:



Entonces, en lugar de R[15] le ponemos esto: " & ULTIMACELDA & " (Recuerde que ULTIMACELDA es la variable que está almacenada como la última celda activa de la fila)

Listo, esta macro5, la uno con la macro2 que acabé de hacer, selecciono lo de la mitad que dice End Sub JUNTO con lo que dice Sub Macro5() y borro ese pedazo:

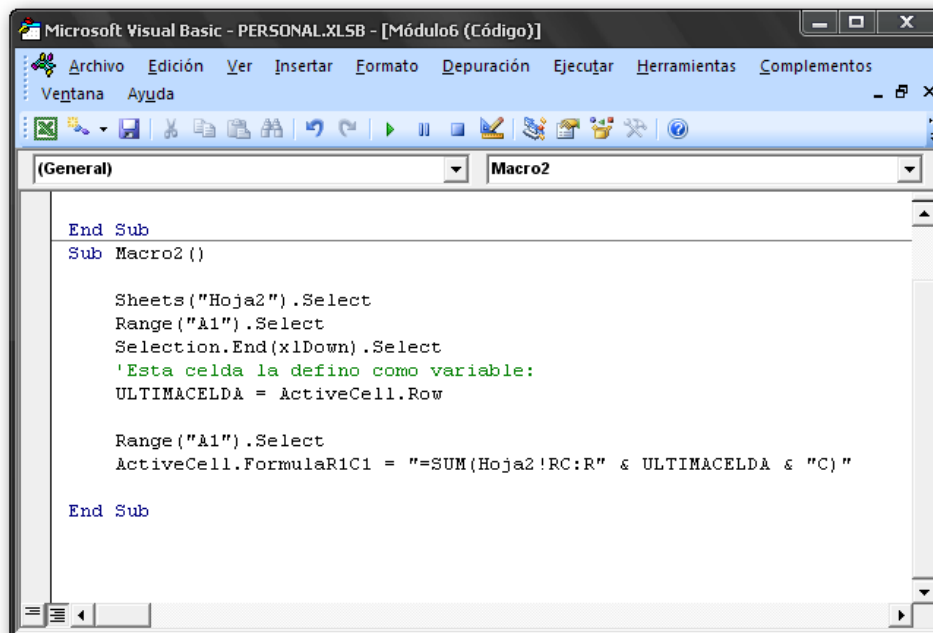


```

Microsoft Visual Basic - PERSONAL.XLSB - [Módulo6 (Código)]
Archivo Edición Ver Insertar Formato Depuración Ejecutar Herramientas Complementos
Ventana Ayuda
(General) Macro5
Sub Macro2 ()
    Sheets("Hoja2").Select
    Range("A1").Select
    Selection.End(xlDown).Select
    'Esta celda la defino como variable:
    ULTIMACELDA = ActiveCell.Row
End Sub
Sub Macro5()
    Range("A1").Select
    ActiveCell.FormulaR1C1 = "=SUM(Hoja2!RC:R" & ULTIMACELDA & "C)"
End Sub

```

Me queda así:



```

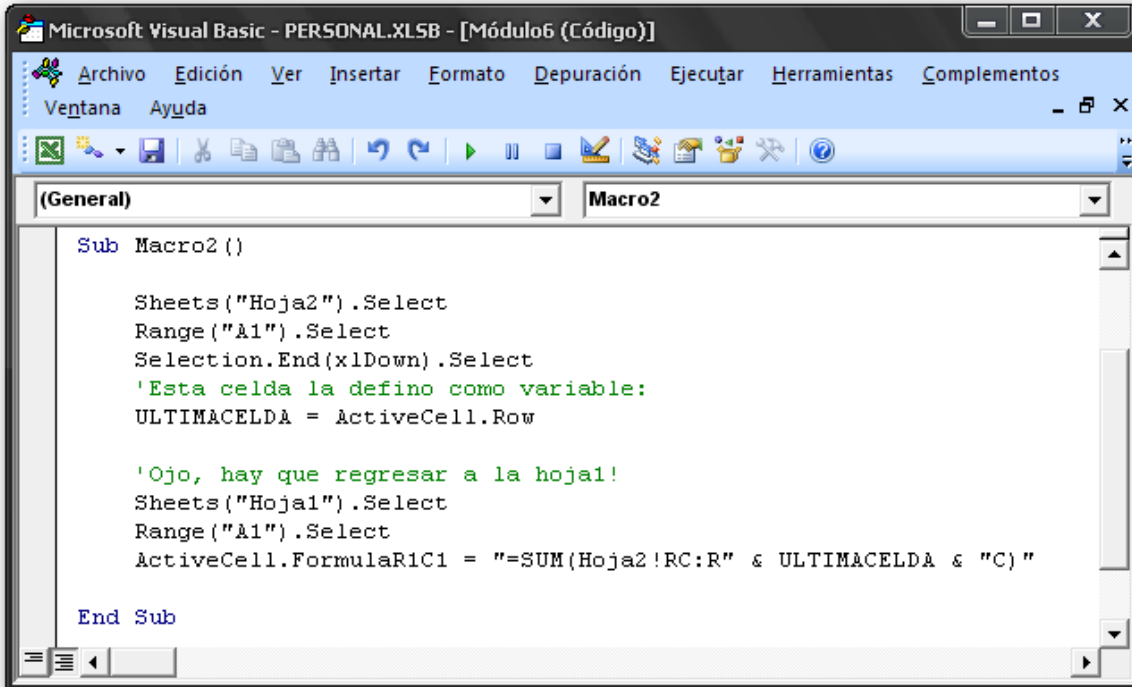
Microsoft Visual Basic - PERSONAL.XLSB - [Módulo6 (Código)]
Archivo Edición Ver Insertar Formato Depuración Ejecutar Herramientas Complementos
Ventana Ayuda
(General) Macro2
End Sub
Sub Macro2 ()
    Sheets("Hoja2").Select
    Range("A1").Select
    Selection.End(xlDown).Select
    'Esta celda la defino como variable:
    ULTIMACELDA = ActiveCell.Row
    Range("A1").Select
    ActiveCell.FormulaR1C1 = "=SUM(Hoja2!RC:R" & ULTIMACELDA & "C)"
End Sub

```

Entonces la comienzo a probar. Con la depuración paso a paso, partiendo la pantalla de tal forma que pueda ver mi macro paso a paso y al lado lo que pasa en mi hoja de Excel. En la hoja1 en Excel, borro la formula y comienzo a correrla paso a paso oprimiendo F8, para ver si todo va bien.

Al llegar al paso `ULTIMACELDA = ActiveCell.Row`,

Me doy cuenta que Excel se dispone a MONTAR LA FORMULA en la misma HOJA2, Y ESO NO PUEDE SER!! TIENE QUE SER EN LA HOJA1, ASI QUE COPIO COMO SE SELECCIONA LA HOJA2, (lo que dice Sheets("Hoja2").Select) Y PEGO EL RENGLON, PERO DICIENDOLE QUE SELECCIONE LA HOJA1, ASI:



Echo la macro a ANDAR, y veo que corre perfecto. LOGICAMENTE EL CODIGO SE PUEDE DEPURAR, aunque notará que sin depurarlo igualmente Excel es SUPERRAPIDO. Una depurada puede dejarlo así:

```
Sub Macro2 ()  
  
    Sheets("Hoja2").Select  
    Range("A1").End(xlDown).Select  
    'Esta celda la defino como variable:  
    ULTIMACELDA = ActiveCell.Row  
  
    'Ojo, hay que regresar a la hoja1!  
    Sheets("Hoja1").Select  
    Range("A1") = "=SUM(Hoja2!RC:R" & ULTIMACELDA & "C)"  
  
End Sub
```

Todavía se puede depurar un poco mas, recuerde como definimos la ultima celda en apartados anteriores, con lo cual esos tres primeros renglones nos quedarían convertidos en solo uno.

Si quiere que la variable ULTIMACELDA Excel la tenga presente en TODA la macro, pues puede que la necesite usar después en otros pasos, deberá declararla al principio de la macro, pero si quiere que funcione solo en ESE PEDACITO o rango, no hay necesidad de declararla.

```

Sub Macro2 ()

Dim ULTIMACELDA As String

    Sheets("Hoja2").Select
    Range("A1").End(xlDown).Select
    'Esta celda la defino como variable:
    ULTIMACELDA = ActiveCell.Row

    'Ojo, hay que regresar a la hojal!
    Sheets("Hoja1").Select
    Range("A1") = "=SUM(Hoja2!RC:R" & ULTIMACELDA & "C)"

End Sub

```

Declarar con DIM es decirle a Excel que reserve un espacio de memoria para almacenar el dato. En este caso, estamos armando una cadena de caracteres que se referirá a la ultima celda de la hoja2, que vendrá a ser lo que anteriormente era la instrucción fija R[15],

Que reemplazamos por " & ULTIMACELDA & "

Si se fija verá que con el operador AMPERSAND (&) entre comillas, que por lo general está encima de la tecla numero 6 del teclado numérico en línea, concatenamos una cadena de caracteres, que nos armará el rango con la ultima fila de un rango mas grande, de mas o menos filas.

Aquí le estoy resaltando y aumentando el tamaño al pedazo de la fórmula en donde monto la cadena con la variable:

```

Sub Macro2()
Dim ULTIMACELDA As String

    Sheets("Hoja2").Select
    Range("A1").End(xlDown).Select
    'Esta celda la defino como variable:
    ULTIMACELDA = ActiveCell.Row

    'Ojo, hay que regresar a la hojal!
    Sheets("Hoja1").Select
    Range("A1") = "=SUM(Hoja2!RC:R" & ULTIMACELDA & "C)"

End Sub

```

Entonces, ya sabe, que CUALQUIER FORMULA que grabe con el grabador de macros, la edita y la organiza con este mismo procedimiento de variable, para que la pueda ejecutar sin problemas en otros archivos que sean diferentes en extensión, pero parecidos en estructura. En este caso se definió el final del rango, pero también se puede definir el inicio del rango con variable, para el caso en que el rango no empiece en la primera fila. También se puede meter dentro de un Loop o un recorrido, pero primero quiero mostrarle otro procedimiento muy útil de recorrido:

Veámos al principio del tema de las macros que por lo general, una macro sigue la siguiente estructura:

Comienza en el primer renglón de información de la hoja de cálculo.
 Examina un valor en una de las columnas.

Basado en este valor, decide hacer algo.

Se mueve al siguiente renglón en el rango de datos y repite.

Y vimos el procedimiento For Next. Ahora vamos a ver o retomar otro que vimos mas arriba y que también es muy bueno y muy parecido, el procedimiento DO UNTIL, que significa, "Ejecute hasta que"

El procedimiento DO UNTIL

Ok. Recuerde que Do, en el inglés del colegio es el verbo hacer. Y Until, significa Hasta-que.

¿Recuerda que cuando le expliqué como solucionar el tema de Rellenar las celdas que están en blanco, le mostré como hacerlo con una macro? Esa macro utiliza el procedimiento Do Until, y en esa sección, expliqué lo necesario para que entendiera esa estructura de recorrido. Ahora que ya conoce la estructura FOR NEXT, comprenderá que esta opción también es buena, también puede ver que Do Until puede incluir la estructura IF-THEN-ELSE (Si tal cosa, entonces, sino...) igual a como lo hicimos en la macro copiar:

```
Sub Copiar()  
  
    Do Until ActiveCell = "Fin"  
        If ActiveCell = "" Then  
            Selection.FillDown  
        End If  
        ActiveCell.Offset(1, 0).Select  
    Loop  
  
End Sub
```

Aquí, debajo del DO Until está la instrucción que dice que SI la celda está vacía, ENTONCES rellene. El SINO, (ELSE) no hay necesidad, pero siendo Ortodoxos, debería haberse puesto así:

```
Sub Copiar()  
    Do Until ActiveCell = "fin"  
        If ActiveCell = "" Then  
            Selection.FillDown  
        Else  
            ActiveCell.Offset(1, 0).Select  
        End If  
    Loop  
End Sub
```

Le cuento que, cuando conocí esta macro, me dí cuenta que en lugar de la acción

```
Selection.FillDown
```

Podía hacer OTRAS COSAS QUE NECESITARA, pues no todo es rellenar, podía insertar filas, o cortar una celda y moverla para otro lado, o copiarla y pegarla en otro lado, meterle formato y regresar a la columna en donde estaban los datos que quería trabajar, ya sea vuelvo y repito, rellenar, copiar, cortar, y finalmente podía regresar al punto en donde debía continuar evaluando la celda activa con la instrucción

```
ActiveCell.Offset
```

Entonces, ya depende de Usted que comience a crear sus macros, el código que le haga falta y que no conozca, ya sabe que puede usar el grabador de macros, usar el procedimiento Cells en lugar del

Select, y luego, intente escribirlas sin necesidad de seleccionar las celdas, así como esta macro copiar la escribimos en estilo For Next. También puede comenzar a profundizar en este maravilloso mundo de las macros y conseguir libros mas avanzados, o buscar en Internet.

Haciendo Un DO UNTIL combinado con variables temporales que nos montan una formula tal como =SUMA()

Ok. A Este punto quería llegar, pues alguna vez me ocurrió, que tuve necesidad de crear un reporte el cual hice con una tabla dinámica, pero la presentación de la tabla dinámica, es una presentación que aunque sea muy bonita, no nos gusta a quienes estamos en el rollo contable, pues la fuerza de la costumbre ha creado ciertas normas de presentación de informes, tales como el subrayado, la negrilla, la fuente, la sangría, etc.

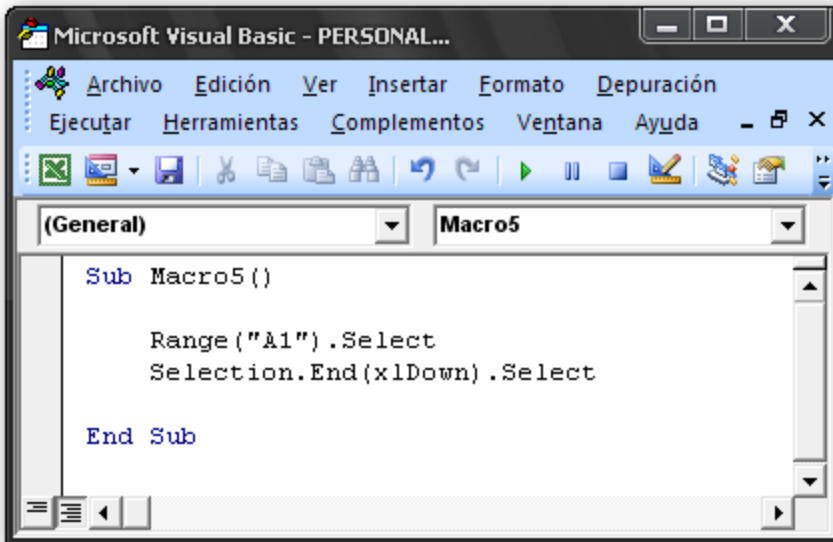
Además nos gusta siempre poder constatar viendo la sumatoria, la función suma de los diferentes rangos, y la tabla dinámica no los pone. Pero si nos podemos valer de ella para armar el reporte.

Supongamos que tenemos una base de datos que ya tenemos organizada en Excel. Recuerde que este ejemplo puede ser otra cosa, como una base de datos de un balance de comprobación por terceros, o un Kardex, etc. La idea es mostrarle UNA de tantas maneras para crear el reporte.

Observe la siguiente información:

	A	B	C	D	E
1	Cuenta	Nombre de la cuenta	Vendedor	Valor	
2	1A	Uno A	Fulanito	4360	
3	1A	Uno A	Menganito	7638	
4	1A	Uno A	Sutanito	6402	
5	1A	Uno A	Perencejo	3100	
6	1A	Uno A	Melgarejo	2951	
7	2B	Dos B	Muerganito	7762	
8	2B	Dos B	Songo	5774	
9	2B	Dos B	Borondongo	9029	
10	2B	Dos B	Bernabé	5509	
11	2B	Dos B	Muchilanga	7135	
12	2B	Dos B	Fulanito	7790	
13	2B	Dos B	Menganito	7993	
14	3C	Tres Cé	Sutanito	3863	
15	3C	Tres Cé	Perencejo	7257	
16	3C	Tres Cé	Melgarejo	714	
17	3C	Tres Cé	Muerganito	1396	
18	3C	Tres Cé	Songo	6048	
19	3C	Tres Cé	Borondongo	9254	
20	3C	Tres Cé	Bernabé	5170	
21					

Prendo el grabador de macros y lo primero que hago es definir que la ultima celda es la variable ULTIMAFILA. Este código es el mismo para todos. Podemos ir desde la ultima fila de Excel hacia arriba, o desde la primera celda hacia abajo. Detengo la grabación de la macro, abro Visual Basic y reviso el código:



Esto ya lo sabe hacer, pero para efectos de repaso lo volvemos a enseñar, para que le quede claro el asunto.

Edito la macro y le defino que la ultima celda es la variable UltimaCelda:

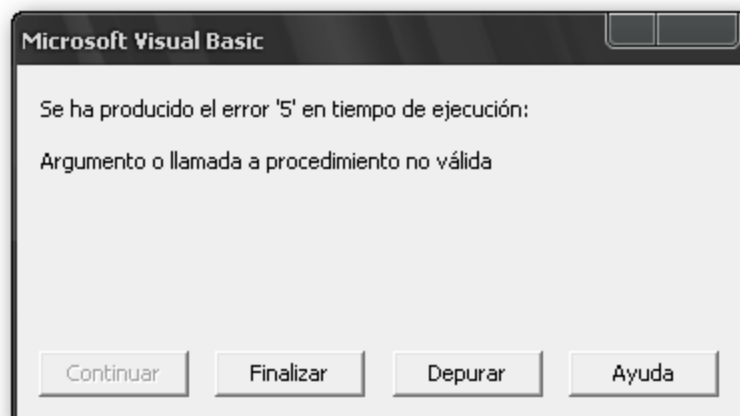
```
Sub Macro5()  
  
    Range("A1").Select  
    Selection.End(xlDown).Select  
    'Defino que esa ultima fila, me _  
    la tenga en cuenta:  
    ULTIMACELDA = ActiveCell.Row  
  
End Sub
```

Listo, vuelvo y prendo el grabador de macros y hago otra macro, esta vez, voy a crear una tabla dinámica sobre la hoja ejemplo que le he mostrado:

	A	B	C	D
1				
2				
3	Cuenta	Nombre de	Vendedor	Suma de Valor
4	1A	Uno A	Fulanito	4360
5			Melgarejo	2951
6			Menganito	7638
7			Perencejo	3100
8			Sutanito	6402
9		Total Uno A		24451
10	Total 1A			24451
11	2B	Dos B	Bernabé	5509
12			Borondongo	9029
13			Fulanito	7790
14			Menganito	7993
15			Muchilanga	7135
16			Muerganito	7762
17			Songo	5774
18		Total Dos B		50992
19	Total 2B			50992
20	3C	Tres Cé	Bernabé	5170
21			Borondongo	9254
22			Melgarejo	714
23			Muerganito	1396
24			Perencejo	7257
25			Songo	6048
26			Sutanito	3863
27		Total Tres Cé		33702
28	Total 3C			33702
29	Total general			109145

Elimino la hoja y pruebo la macro....

En teoría solo es correrla y ya, pero si me está siguiendo, "En vivo", podrá comprobar que Excel 2007 saca un error.



Al hacer clic en Depurar, nos lleva a este sector:

```
Sub Macro7()  
  
Range("A1").Select  
Sheets.Add  
ActiveWorkbook.PivotCaches.Create(SourceType:=xlDatabase, SourceData:=  
"Hoja1!R1C1:R20C4", Version:=xlPivotTableVersion12).CreatePivotTable _  
TableDestination:="Hoja6!R3C1", TableName:="Tabla dinámica2", _  
DefaultVersion:=xlPivotTableVersion12  
Sheets("Hoja6").Select  
Cells(3, 1).Select  
With ActiveSheet.PivotTables("Tabla dinámica2").PivotFields("Cuenta")  
    .Orientation = xlRowField  
    .Position = 1  
End With  
With ActiveSheet.PivotTables("Tabla dinámica2").PivotFields( _  
"Nombre de la cuenta")  
    .Orientation = xlRowField  
    .Position = 2  
End With  
With ActiveSheet.PivotTables("Tabla dinámica2").PivotFields("Vendedor")  
    .Orientation = xlRowField  
    .Position = 3  
End With  
ActiveSheet.PivotTables("Tabla dinámica2").AddDataField ActiveSheet.PivotTables _  
("Tabla dinámica2").PivotFields("Valor"), "Suma de Valor", xlSum  
ActiveSheet.PivotTables("Tabla dinámica2").RowAxisLayout xlTabularRow  
End Sub
```

Me parece Fabuloso que halla ocurrido este error, para de una vez explicarle que las tablas dinámicas en Excel2007 que se graban con el grabador de macros, deberían correr sin problema como lo hacían en el Excel 2003, pero no, aquí fallan, y este problema me pasó una vez y lo tuve que solucionar estudiando el asunto de las tablas dinámicas en Macro.

Este punto es de alta prioridad, pues la tabla dinámica es un poderoso reporteador que prácticamente nos arma el informe como lo queramos, así que es MORTAL que no nos corra la macro de una tabla dinámica.

Analizando el problema porque la Macro de la tabla dinámica no corre.

```

Sub Macro7()
  Range("A1").Select
  Sheets.Add
  ActiveWorkbook.PivotCaches.Create(SourceType:=xlDatabase, SourceData:= _
    "Hoja1!R1C1:R20C4", Version:=xlPivotTableVersion12).CreatePivotTable _
    TableDestination:="Hoja6!R3C1", TableName:="Tabla dinámica2", _
    DefaultVersion:=xlPivotTableVersion12
  Sheets("Hoja6").Select
  Cells(3, 1).Select
  With ActiveSheet.PivotTables("Tabla dinámica2").PivotFields("Cuenta")
    .Orientation = xlRowField
    .Position = 1
  End With
  With ActiveSheet.PivotTables("Tabla dinámica2").PivotFields( _
    "Nombre de la cuenta")
    .Orientation = xlRowField
    .Position = 2
  End With
  With ActiveSheet.PivotTables("Tabla dinámica2").PivotFields("Vendedor")
    .Orientation = xlRowField
    .Position = 3
  End With
  ActiveSheet.PivotTables("Tabla dinámica2").AddDataField ActiveSheet.PivotTables _
    ("Tabla dinámica2").PivotFields("Valor"), "Suma de Valor", xlSum
  ActiveSheet.PivotTables("Tabla dinámica2").RowAxisLayout xlTabularRow
End Sub

```

EL grabador de macros usa el procedimiento Add de la colección PivotCaches para crear una hoja nueva que pasa a ser la hoja activa. Fíjese que toma el rango del cuadro de nuestro ejemplo, "Hoja1!R1C1:R20C4", quiere decir, Hoja1,Fila1Columna1 hasta la Fila20Columna4

Siguiendo, podemos ver que lo que sigue es que se usa el método .CreatePivotTable para crear la tabla dinámica, y la tabla destino es la "Hoja6!R3C1" Pareciera que el problema está aquí, pues el grabador de macros definió que la hoja destino siempre será la hoja6, pero si se borra la hoja de la tabla dinámica, o volvemos a correr la macro, no será la hoja6, ni la hoja1 o 2, Excel creará la siguiente hoja, si yo volviera a correr la macro, Excel añadiría la Hoja7, y por eso es que se "totea" la macro.

[Nota: una solución es, en lugar de crear la tabla en una hoja diferente *pues hacerla en una columna a la derecha de la base de datos dentro de la misma hoja, ahí sí, correría sin problema*]

La tabla dinámica arranca en esta hoja6 en la celda de la tercera fila y la primera columna, dejando renglones para ubicar los campos de columna de la tabla dinámica. Los campos "Cuenta", "Nombre de la cuenta", "Vendedor", se definen como campos de fila.

Finalmente el campo "Valor" queda definido como campo de información (AddDataField), lo cual significa que aparecerá en el cuerpo de la tabla donde será sumado.

Como se puede ver, el código es muy inflexible, adiciona una nueva hoja y asume que se llamará hoja1, u hoja6. Le asigna el nombre TablaDinamica2 o TablaDinamica1 y depende del nombre del código siguiente. El código se puede organizar así:

Excel aplicado a la Auditoría, la Contaduría y la Administración

```
Sub CreaTablaDinamicaConTabla()  
  
Dim wks As Worksheet  
Dim pvc As PivotCache  
Dim pvt As PivotTable  
  
Range("A1").End(xlDown).Select  
ultima = ActiveCell.Row  
  
ActiveSheet.ListObjects.Add(xlSrcRange, Range("A1:D" & ultima & ""), xlYes).Name = _  
    "Tabla1"  
  
'Adiciona una Nueva hoja  
Set wks = Worksheets.Add  
  
'Se crea el PivotCache  
Set pvc = ActiveWorkbook.PivotCaches.Create( _  
    SourceType:=xlDatabase, _  
    SourceData:="Tabla1")  
  
'Se Crea la Tabla Dinámica  
Set pvt = pvc.CreatePivotTable(TableDestination:=wks.Range("A3"), _  
    DefaultVersion:=xlPivotTableVersion12)  
  
'Se Definen los campos en la Tabla dinámica  
With pvt  
  
    With .PivotFields("Cuenta")  
        .Orientation = xlRowField  
        .Position = 1  
    End With  
    With .PivotFields("Nombre de la Cuenta")  
        .Orientation = xlRowField  
        .Position = 2  
        .Subtotals = Array(False, False, False, False, False, False, False, False, False,  
False, _  
False, False)  
    End With  
  
    With .PivotFields("Vendedor")  
        .Orientation = xlRowField  
        .Position = 3  
    End With  
  
    .RowAxisLayout xlTabularRow  
  
    .AddDataField .PivotFields("Valor"), "Suma de Valor", xlSum  
  
End With  
End Sub
```

Ok. Primero que todo, declaro las variables Worksheet, PivotCaché y PivotTable

```
Dim wks As Worksheet  
Dim pvc As PivotCache  
Dim pvt As PivotTable
```

Ubico la ultima Celda de la tabla y la defino como la variable Ultima

```
Range("A1").End(xlDown).Select  
ultima = ActiveCell.Row
```

El paso siguiente lo realicé con el grabador de macros, mi tabla, la convierto en tabla de Excel, usando la opción de la ficha Insertar, Tabla. Luego el rango lo edito y arreglo que se vaya a mi variable Ultima, como le enseñé a hacerlo en el apartado anterior:

```
ActiveSheet.ListObjects.Add(xlSrcRange, Range("A1:D" & ultima & ""), xlYes).Name = _  
    "Tabla1"
```

Ahora mediante la instrucción SET, (esta palabra en inglés tiene muchos significados pero el más idóneo puede ser como Fijar o establecer que)

```
'Adiciona una Nueva hoja
Set wks = Worksheets.Add

'Se crea el PivotCache
Set pvc = ActiveWorkbook.PivotCaches.Create( _
    SourceType:=xlDatabase, _
    SourceData:="Tabla1")

'Se Crea la Tabla Dinámica
Set pvt = pvc.CreatePivotTable(TableDestination:=wks.Range("A3"), _
    DefaultVersion:=xlPivotTableVersion12)
```

Si se fija todos estos SET van estableciendo una igualdad, como indicando que wks, pvc y pvt, corresponden a las acciones y atributos definidos. Todo esto es similar en todas las tablas dinámicas, así que cuando monte la suya, puede guiarse por toda esta parte.

Ahora vienen los campos de la tabla dinámica:

```
'Se Definen los campos en la Tabla dinámica
With pvt

    With .PivotFields("Cuenta")
        .Orientation = xlRowField
        .Position = 1
    End With
    With .PivotFields("Nombre de la Cuenta")
        .Orientation = xlRowField
        .Position = 2
        .Subtotals = Array(False, False, False, False, False, False, False, False, False, False,
False, _
False, False)
    End With

    With .PivotFields("Vendedor")
        .Orientation = xlRowField
        .Position = 3
    End With

    .RowAxisLayout xlTabularRow

    .AddDataField .PivotFields("Valor"), "Suma de Valor", xlSum

End With

End Sub
```

Si se fija, por cada uno de los campos que he llevado a la tabla dinámica se monta una instrucción With-End_With. En el campo tabla dinámica le he adicionado que NO ME HAGA SUBTOTAL POR ALLÍ. (.Subtotals = Array(False, False, False, False, False, False, False, False, False, False, False, False, False))

Después del tercer campo, "Vendedor", defino que el estilo de la tabla dinámica sea Tabular, que es el estilo de tabla dinámica de Excel 2003 al que estamos acostumbrados, que nos totalice por abajo, ya que las nuevas tablas dinámicas de Excel 2007 por defecto dejan el estilo ESQUEMA. Por Último, el campo que tiene la suma de los totales, el campo valor, es esta línea:

```
.AddDataField .PivotFields("Valor"), "Suma de Valor", xlSum
```

Ok. Esta es una manera de solucionar el asunto de la tabla dinámica, mucho más flexible y al convertir la tabla de datos en Tabla de Excel, le facilita a Excel su aprehensión. Por supuesto que no es la única, es posible que después de leer esta manera se la halla ocurrido otra mejor, lo cual me parece fantástico, pues significará que ya se metió de lleno en el tema.

Hay otras maneras de montar la tabla dinámica, pero esta sirve para el propósito. Cuando monte SU TABLA DINAMICA, solo fíjese como van los campos en fila y columna, y haga los ajustes necesarios.

Si prueba la macro, elimina la tabla dinámica y vuelve y la corre, verá como funciona perfectamente. Acuérdesse de convertir la tabla fuente nuevamente en rango, y de transcribir CORRECTAMENTE el código que le suministro.

	A	B	C	D
1				
2				
3	Cuenta	Nombre de	Vendedor	Suma de Valor
4	1A	Uno A	Fulanito	4360
5			Melgarejo	2951
6			Menganito	7638
7			Perencejo	3100
8			Sutanito	6402
9	Total 1A			24451
10	2B	Dos B	Bernabé	5509
11			Borondongo	9029
12			Fulanito	7790
13			Menganito	7993
14			Muchilanga	7135
15			Muerganito	7762
16			Songo	5774
17	Total 2B			50992
18	3C	Tres Cé	Borondongo	9254
19			Melgarejo	714
20			Muerganito	1396
21			Perencejo	7257
22			Songo	6048
23			Sutanito	3863
24	Total 3C			28532
25	Total general			103975

Ok, habiendo construido la tabla dinámica mediante macro, podemos retomar el tema que veníamos tratando:

Convertir este informe en uno con más presentación.

La idea es:

- Que el nombre de la cuenta, esas que me he inventado que se llaman Uno A, Dos B y Tres Cé, trasladarla a la columna C, en un nuevo renglón que hemos de Insertar y,
- Hacer que los totales nos queden con la formula Suma.
- Ponerles un formato a los números

Haciendo que la macro que nos haga el recorrido con las variables de los rangos de inicio y final de cada bloque de valores de cada una de las cuentas.

Ok. Lo otro es prender el grabador de macros y "Desbaratar" la tabla dinámica, haciendo un copiar y allí mismo pegar solo valores:

```
Sub Macro6()
    Columns("A:D").Select
    Selection.Copy
    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
        :=False, Transpose:=False
    Application.CutCopyMode = False
    Range("A1").Select
End Sub
```

Nota: Recuerde que puede mejorar el código seleccionando solo el rango a copiar. Listo, después de esto nos va quedando así:

	A	B	C	D
1				
2				
3	Cuenta	Nombre de la	Vendedor	Suma de Valor
4	1A	Uno A	Fulanito	4360
5			Melgarejo	2951
6			Menganito	7638
7			Perencejo	3100
8			Sutanito	6402
9	Total 1A			24451
10	2B	Dos B	Bernabé	5509
11			Borondongo	9029
12			Fulanito	7790
13			Menganito	7993
14			Muchilanga	7135
15			Muerganito	7762
16			Songo	5774
17	Total 2B			50992
18	3C	Tres Cé	Borondongo	9254
19			Melgarejo	714
20			Muerganito	1396
21			Perencejo	7257
22			Songo	6048
23			Sutanito	3863
24	Total 3C			28532
25	Total general			103975

Ok. Lo bueno de esta tabla dinámica es que siempre termina con la palabra "Total general", así que la podemos usar para hacer un DO hasta que la celda sea igual a "Total general".

Bueno, ya hicimos una buena parte de la macro, que es crear la tabla dinámica, desbaratarla y luego prendo el grabador de macros otra vez y elimino la fila 3. Luego me ubico en la Celda C1.

La cosa me queda así:

```
Sub Macro8()  
  
    Rows("3:3").Select  
    Selection.Delete Shift:=xlUp  
    Range("C1").Select  
  
End Sub
```

Después puede depurar un poco el código, no se preocupe por eso ahora.

Ahora si, voy a meter una macro escrita, (guiándome por la macro copiar) que, Haga o Ejecute Hasta que la celda de la columna A sea igual a "Total general"; que SI la celda está en blanco, que baje una celda, SINO, indico que ese campo es la variable inicio, con End + flechita hacia abajo voy al final del rango y declaro que ese campo es la variable final, luego me desvío una celda a la derecha y una celda abajo, y monto la función suma, con las dos variables, inicio y final. Y como decimos en Colombia, "LISTO EL POLLO"....

"Tonces", o "Tonz", como escriben los muchachos ahora en los chats, la macro nos queda así:

```
Sub prueba()  
Do Until ActiveCell.Offset(0, -2) = "Total general"  
    If ActiveCell = "" Then  
        ActiveCell.Offset(1, 0).Select  
    Else  
        inicio = ActiveCell.Row  
        Selection.End(xlDown).Select  
        final = ActiveCell.Row  
        ActiveCell.Offset(1, 1).Select  
        ActiveCell.Formula = "=SUM(R" & inicio & "C:R" & final & "C)"  
        ActiveCell.Offset(0, -1).Select  
    End If  
Loop  
End Sub
```

Quiero decirle que esto "cuando se le va cogiendo el tiro", a fuerza de tanto repetirlo uno hasta se lo va aprendiendo, pero a veces lo que uno no sabe, pues se ayuda con el grabador de macros. Por ejemplo, en este caso no me acordaba como escribir el código de la función suma, así que PRENDI el grabador de macros y grabe una macro solo con la suma, así:

```
Sub Macro9()  
'  
' Macro9 Macro  
'  
    Selection.FormulaR1C1 = "=SUM(R[-5]C:R[-1]C)"  
End Sub
```

Luego, lo que está entre los corchetes [] le meto cada una de las variables inicio y final:

```
ActiveCell.Formula = "=SUM(R" & inicio & "C:R" & final & "C)"
```

Prueba la macro paso a paso a ver si funciona bien, y listo.

Otra cosa que se debe propender por hacer, y esto es muy importante, es hacer un solo recorrido y en él ir haciendo todo de una vez. Por ejemplo, nosotros quisiéramos FORMATEAR los números de una vez.

```
Sub Macro11()
'
' Macro11 Macro

Range("D1048576").Select
Selection.End(xlUp).Select
Range(Selection, Selection.End(xlUp)).Select
Selection.NumberFormat = "#,##0_);(#,##0)"
End Sub
```

Que hice: prendí el grabador de macros, escogí la última celda de la columnaB. Oprimí FIN + flecha arriba y me estrellé con la ultima celda de mi hoja de cálculo. Ahora Oprimo SHIFT + Fin + Flecha Arriba para seleccionar el rango. Una vez que me quedó seleccionado hago clic en el botón izquierdo del ratón y en el menú emergente le doy la opción seleccionar formato de celdas, y le cuadro el formato que quiero. Detengo la macro y lo que me quedó fue esto que le enseñé aquí.

Este código se puede depurar, las primeras dos líneas las podemos dejar en una sola:

```
Sub Macro11()
'
' Macro11 Macro

Range("D1048576").End(xlUp).Select
Range(Selection, Selection.End(xlUp)).Select
Selection.NumberFormat = "#,##0_);(#,##0)"
End Sub
```

(Incluso el primer Range("D1048576).end(xlUp) se podría meter en lugar del Selection del segundo "Range", pero puede ser mejor hacer que Excel se detenga en el primer punto del rango.)

Listo, después de esto, ya tenemos las macros que nos hacen las sumas correctas y los formatos.

Pero, en el recorrido en el cual hacemos la sumatoria, podemos meterle un formato a la celda de la suma con una raya arriba y una raya gruesa abajo.

Entonces, Prenda nuevamente el grabador de macros, coja una celda en las cuales hay un total, métale el formato, detenga la macro y copiemos ese código y peguémoslo en la macro que hace las sumas. Este es el código que nos hace la raya y la doble raya:

```
Sub Macro13 ()

Selection.Borders(xlDiagonalDown).LineStyle = xlNone
Selection.Borders(xlDiagonalUp).LineStyle = xlNone
Selection.Borders(xlEdgeLeft).LineStyle = xlNone
With Selection.Borders(xlEdgeTop)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
With Selection.Borders(xlEdgeBottom)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlMedium
End With
Selection.Borders(xlEdgeRight).LineStyle = xlNone
Selection.Borders(xlInsideVertical).LineStyle = xlNone
Selection.Borders(xlInsideHorizontal).LineStyle = xlNone
End Sub
```

Como se puede dar cuenta, un par de infelices rayas generan todo este código y casi todo dice "NONE"... Ninguno. Todos esos "NONES", se los podemos quitar, y (por lo menos a mí) me sigue funcionando: (pruebe si sigue funcionando al quitar los valores en Cero, aquí la línea importante es la que define el peso o intensidad de la línea)

```
Sub Macro13 ()

With Selection.Borders(xlEdgeTop)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
With Selection.Borders(xlEdgeBottom)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlMedium
End With

End Sub
```

Ok, este código lo voy a cortar y pegar en el código que hace la suma, debajo de la operación de la suma:

```

Sub prueba()

Do Until ActiveCell.Offset(0, -2) = "Total general"

    If ActiveCell = "" Then
        ActiveCell.Offset(1, 0).Select
    Else
        inicio = ActiveCell.Row
        Selection.End(xlDown).Select
        final = ActiveCell.Row
        ActiveCell.Offset(1, 1).Select
        ActiveCell.Formula = "=SUM(R" & inicio & "C:R" & final & "C)"
        With Selection.Borders(xlEdgeTop)
            .LineStyle = xlContinuous
            .ColorIndex = 0
            .TintAndShade = 0
            .Weight = xlThin
        End With
        With Selection.Borders(xlEdgeBottom)
            .LineStyle = xlContinuous
            .ColorIndex = 0
            .TintAndShade = 0
            .Weight = xlMedium
        End With
        ActiveCell.Offset(0, -1).Select
    End If
Loop

End Sub

```

Listo. Se coloca antes de que regrese a la columna en la cual realiza el recorrido.

Ok, también necesitamos cortar el nombre de la cuenta (las que he llamado UnoA, DosB, TresCé), insertar una fila y trasladar la celda cortada como título, y ponerle negrita y subrayado. (O en lugar de cortarla, copiarla...) Como está en el renglón o fila en el cual inicia la variable de la suma, podemos ahí mismo hacer este pedazo.

Entonces, La misma cosa, me ubico en la celda en donde comienza la variable de la suma, que es la misma en la cual está el nombre de la cuenta, PRENDO el grabador de macros, y hago lo que necesito: Inserto una fila, corto lo el nombre de la cuenta, lo traslado a donde necesito, le coloco el formato negrita, subrayado y le subo al tamaño. Apago el grabador y me queda esto:


```
Sub Macro14()  
  
Selection.EntireRow.Insert , CopyOrigin:=xlFormatFromLeftOrAbove  
Range("B4").Select  
Selection.Cut  
Range("C3").Select  
ActiveSheet.Paste  
Selection.Font.Bold = True  
Selection.Font.Underline = xlUnderlineStyleSingle  
Selection.Font.Size = 11  
  
End Sub
```

Fíjese que nos toca editar todas esas celdas que quedaron seleccionadas, las (Range("B4").Select) y (Range("C3").Select)

Esto es lo que le comentaba que nos hace el grabador de macros y que hay que arreglar con la técnica del ActiveCell.Offset. Creo que ahora me comprende bien la importancia, dado que la macro hace una selección fija y no se seguiría moviendo. Ese par de celdas, se edita así:

```
Sub Macro14()  
  
Selection.EntireRow.Insert , CopyOrigin:=xlFormatFromLeftOrAbove  
'Range("B4").Select  
'Me muevo una celda abajo y una a la izquierda  
ActiveCell.Offset(1, -1).Select  
Selection.Cut  
'Range("C3").Select  
'El corte lo subo una celda arriba y una celda a la derecha:  
ActiveCell.Offset(-1, 1).Select  
ActiveSheet.Paste  
Selection.Font.Bold = True  
Selection.Font.Underline = xlUnderlineStyleSingle  
Selection.Font.Size = 11  
  
End Sub
```

Lo que acabo de editar, lo pruebo corriéndolo paso a paso, pues esos OFFSET tienden a enredarlo a veces a uno, pero uno corrige al ejecutar paso a paso. Ok, si está bien, ahora a buscarle el punto apropiado en el cual insertar este código dentro del que nos hace las sumas:

```

Sub prueba ()
Do Until ActiveCell.Offset(0, -2) = "Total general"

    If ActiveCell = "" Then
        ActiveCell.Offset(1, 0).Select
    Else

        Selection.EntireRow.Insert , CopyOrigin:=xlFormatFromLeftOrAbove
        'Range("B4").Select
        'Me muevo una celda abajo y una a la izquierda
        ActiveCell.Offset(1, -1).Select
        Selection.Cut
        'Range("C3").Select
        'El corte lo subo una celda arriba y una celda a la derecha:
        ActiveCell.Offset(-1, 1).Select
        ActiveSheet.Paste
        Selection.Font.Bold = True
        Selection.Font.Underline = xlUnderlineStyleSingle
        Selection.Font.Size = 11
        ActiveCell.Offset(1, 0).Select

        inicio = ActiveCell.Row
        Selection.End(xlDown).Select
        final = ActiveCell.Row
        ActiveCell.Offset(1, 1).Select
        ActiveCell.Formula = "=SUM(R" & inicio & "C:R" & final & "C)"
        With Selection.Borders(xlEdgeTop)
            .LineStyle = xlContinuous
            .ColorIndex = 0
            .TintAndShade = 0
            .Weight = xlThin
        End With
        With Selection.Borders(xlEdgeBottom)
            .LineStyle = xlContinuous
            .ColorIndex = 0
            .TintAndShade = 0
            .Weight = xlMedium
        End With
        ActiveCell.Offset(0, -1).Select
    End If
Loop
End Sub

```

OK. Fíjese que la macro que nos hizo las sumas parciales, se ha enriquecido para que en el primer repaso se haga lo necesario, para no tener que volver a hacer el recorrido.

Esta macro, la podemos unir debajo de las otras que ya hicimos:

- La que nos armó la tabla dinámica.
- La que nos arregló los valores

Todo Junto, nuevamente se prueba paso a paso para ver que nos queda faltando, o en donde falla. Solo falta eliminar las dos primeras columnas, y en la última columna colocar un gran total. Eliminar las columnas, ya sabe como hacerlo con el grabador de macros:

```
Sub Macro18()

    Columns("A:B").Select
    Selection.Delete Shift:=xlToLeft

End Sub
```

Que depurado, ya sabe que le puede quedar así:

```
Sub Macro18()

    Columns("A:B").Delete Shift:=xlToLeft

End Sub
```

En este punto, la transformación ya va así:

	A	B	C	D
1				
2				
3	Uno A			
4	Fulanito	4.360		
5	Melgarejo	2.951		
6	Menganito	7.638		
7	Perencejo	3.100		
8	Sutanito	6.402		
9		24.451		
10	Dos B			
11	Bernabé	5.509		
12	Borondongo	9.029		
13	Fulanito	7.790		
14	Menganito	7.993		
15	Muchilanga	7.135		
16	Muerganito	7.762		
17	Songo	5.774		
18		50.992		
19	Tres Cé			
20	Borondongo	9.254		
21	Melgarejo	714		
22	Muerganito	1.396		
23	Perencejo	7.257		
24	Songo	6.048		
25	Sutanito	3.863		
26		28.532		
27				
28				

Listo, en la celda B27, podemos colocar un total que nos sume las sumas parciales. Es repetir, de otra forma, lo que le expliqué acerca de cómo montar una fórmula con variables:

En este caso necesitaríamos tomar el punto donde comienza el rango, la celda B4, y la última, la B26. Este rango lo podemos usar en una función =Sumar.Si(). Si se fija, al lado de los totales, en la columna A, hay una celda en blanco, entonces, podemos hacer que un Sumar.Si de todo lo que se halle en blanco en ese rango en la columna A.

Entonces, necesito encontrar los puntos que serán las variables de mi función. Prendo el grabador, al primer punto del rango luego escogiendo la columna B1 y oprimiendo Fin+Flecha abajo. El otro punto, lo encuentro subiendo desde la parte de abajo de la hoja de cálculo y oprimiendo Fin+Flecha Arriba. Luego edito un poco lo que me grabó la macro.

Esto es lo que hizo el grabador de macros:

```
Sub Macro19()

    Range("B1").Select
    Selection.End(xlDown).Select
    Application.Goto Reference:="R1048576C2"
    Selection.End(xlUp).Select

End Sub
```

Editando los puntos indicando las variables primerpunto y segundopunto, y borrando el valor total que nos dejó la tabla dinámica, queda así:

```
Sub Macro19()

    Range("B1").Select
    Selection.End(xlDown).Select
    primerpunto = ActiveCell.Row

    Application.Goto Reference:="R1048576C2"
    Selection.End(xlUp).Select
    Selection.ClearContents
    ActiveCell.Offset(-1, 0).Select
    segundopunto = ActiveCell.Row

End Sub
```

Ok, ahora nos falta colocar la función =Sumar.Si, justo al final del rango. Prendo el grabador de macros y realizo la operación de subir por la parte inferior y escribir la función Sumar.Si, (De una vez se le coloca el formato deseado) (luego la arreglo metiéndole las variables) esto es lo que nos queda:

```
Sub Macro21()

    Range("B1048576").Select
    Selection.End(xlUp).Select
    Range("B27").Select
    ActiveCell.FormulaR1C1 = "=SUMIF(R[-23]C[-1]:R[-1]C[-1],""""",R[-23]C:R[-1]C)"
    Selection.Font.Bold = True
    Selection.NumberFormat = "#,##0.00_);[Red](#,##0.00)"

End Sub
```

Ok, corrigiendo el código queda así:

```
Sub Macro21()
    Range("B1048576").Select
    Selection.End(xlUp).Select
    ActiveCell.Offset(1, 0).Select
    'Range("B27").Select
    'ActiveCell.FormulaR1C1 = "=SUMIF(R[-23]C[-1]:R[-1]C[-1],"",R[-23]C:R[-1]C)"
    ActiveCell.FormulaR1C1 = _
    "=SUMIF(R" & primerpunto & "C[-1]:R" & segundopunto & "C[-1],""",R" & primerpunto & "C:R" & segundopunto & "C)"
    Selection.Font.Bold = True
    Selection.NumberFormat = "#,##0.00_);[Red] (#,##0.00)"
End Sub
```

La línea a tener en cuenta es esta:

```
ActiveCell.FormulaR1C1 = _
    "=SUMIF(R" & primerpunto & "C[-1]:R" & segundopunto & "C[-1],""",R" & primerpunto & "C:R" & segundopunto & "C)"
```

Ojo, "C[-1]" VA PEGADO DEL &, lo que pasa es que el Word donde he hecho el manuscrito del libro me lo desplaza un renglón abajo, pero todo va JUNTO, como se ve en la imagen.

Fíjese que de una vez le metí el formato, y le puse negrita. Ahora hay que unir esta macro con la que define las variables, así:

```
Sub Macro19()
    Range("B1").Select
    Selection.End(xlDown).Select
    primerpunto = ActiveCell.Row

    Application.Goto Reference:="R1048576C2"
    Selection.End(xlUp).Select
    Selection.ClearContents
    ActiveCell.Offset(-1, 0).Select
    segundopunto = ActiveCell.Row
End Sub
'Sub Macro21()
    Range("B1048576").Select
    Selection.End(xlUp).Select
    ActiveCell.Offset(1, 0).Select
    'Range("B27").Select
    'ActiveCell.FormulaR1C1 = "=SUMIF(R[-23]C[-1]:R[-1]C[-1],"",R[-23]C:R[-1]C)"
    ActiveCell.FormulaR1C1 = _
    "=SUMIF(R" & primerpunto & "C[-1]:R" & segundopunto & "C[-1],""",R" & primerpunto & "C:R" & segundopunto & "C)"
    Selection.Font.Bold = True
    Selection.NumberFormat = "#,##0.00_);[Red] (#,##0.00)"
End Sub
```

Como siempre, les puse el apóstrofo para que queden como comentario y así saber en donde comenzaba las macros, recuerde que cuando se le coloca un apóstrofo, la línea de código queda sin efecto, y nos ayuda para que mientras entendemos el tema, cuando volvamos a ver esta macro sepamos que hace cada bloque de código.

Cualquier función, anidada, matricial, la puede hacer con el grabador de macros y luego le acomoda los puntos de los rangos con las variables, así, si el archivo es mas o menos largo, la función NO LE FALLARÁ.

Si lo desea, puede pasarse a la celda de al lado en la columna A, y colocar algo así como "Gran Total" También puede insertar unas filas en la parte superior y colocar el título de su reporte.

Esto ya lo puede hacer por su cuenta, recuerde desviarse con ActiveCell.Offset

	A	B	C	D
1				
2				
3	Uno A			
4	Fulanito	4.360		
5	Melgarejo	2.951		
6	Menganito	7.638		
7	Perencejo	3.100		
8	Sutanito	6.402		
9		24.451		
10	Dos B			
11	Bernabé	5.509		
12	Borondongo	9.029		
13	Fulanito	7.790		
14	Menganito	7.993		
15	Muchilanga	7.135		
16	Muerganito	7.762		
17	Songo	5.774		
18		50.992		
19	Tres Cé			
20	Borondongo	9.254		
21	Melgarejo	714		
22	Muerganito	1.396		
23	Perencejo	7.257		
24	Songo	6.048		
25	Sutanito	3.863		
26		28.532		
27		=SUMAR.SI(A\$4:A\$26;"";B\$4:B\$26)		

En esta imagen puede ver como la función que nos montó la macro, está operando bien.

Finalmente todo nos quedaría así: (Para efectos didácticos, voy a ponerle un APOSTROFO en donde iban los Sub y End sub, para anularlos pero poder ver en donde comenzaban esas macros parciales): Recuerde que solo puede quedar un SUB al inicio y otro END SUB al final. Al cortar y pegar los bloques, revise que no le queden sobrando, o la macro no le correrá completamente, o le sacará un mensaje de error:

(Recuerde que si va a probar toda la macro, tiene que volver a convertir la TABLA fuente en RANGO)

```
Sub CreaTablaDinamicaConTabla()

    Dim wks As Worksheet
    Dim pvc As PivotCache
    Dim pvt As PivotTable

    Range("A1").End(xlDown).Select
    ultima = ActiveCell.Row

    ActiveSheet.ListObjects.Add(xlSrcRange, Range("A1:D" & ultima & ""), xlYes).Name = _
        "Tabla1"

    'Adiciona una Nueva hoja
    Set wks = Worksheets.Add
```

Excel aplicado a la Auditoría, la Contaduría y la Administración

```
'Se crea el PivotCache
Set pvc = ActiveWorkbook.PivotCaches.Create( _
    SourceType:=xlDatabase, _
    SourceData:="Tabla1")

'Se Crea la Tabla Dinámica
Set pvt = pvc.CreatePivotTable(TableDestination:=wks.Range("A3"), _
    DefaultVersion:=xlPivotTableVersion12)

'Se Definen los campos en la Tabla dinámica
With pvt

    With .PivotFields("Cuenta")
        .Orientation = xlRowField
        .Position = 1
    End With
    With .PivotFields("Nombre de la Cuenta")
        .Orientation = xlRowField
        .Position = 2
        .Subtotals = Array(False, False, False, False, False, False, False, False, False,
False, _
False, False)
    End With

    With .PivotFields("Vendedor")
        .Orientation = xlRowField
        .Position = 3
    End With

    .RowAxisLayout xlTabularRow

    .AddDataField .PivotFields("Valor"), "Suma de Valor", xlSum

End With

'End Sub
'Sub Macro6()

Columns("A:D").Select
Selection.Copy
Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
:=False, Transpose:=False
Application.CutCopyMode = False
' Range("A1").Select (esto ya es innecesario)

'End Sub
'Sub Macro8()

Rows("3:3").Select
Selection.Delete Shift:=xlUp
' Range("C1").Select (ya no hace falta)

'End Sub
'Sub Macro12()

Range("D1048576").Select
Selection.End(xlUp).Select
Range(Selection, Selection.End(xlUp)).Select
Selection.NumberFormat = "#,##0_);(,##0)"

'End Sub
'Sub prueba()
'Esta instrucción se tuvo que poner, pues el Do iba a comenzar en _
donde finalizó la anterior, y es un error que comience el Loop allí.

Range("C1").Select

Do Until ActiveCell.Offset(0, -2) = "Total general"

    If ActiveCell = "" Then
        ActiveCell.Offset(1, 0).Select
```

```

Else

Selection.EntireRow.Insert , CopyOrigin:=xlFormatFromLeftOrAbove
'Range("B4").Select
'Me muevo una celda abajo y una a la izquierda
ActiveCell.Offset(1, -1).Select
Selection.Cut
'Range("C3").Select
'El corte lo subo una celda arriba y una celda a la derecha:
ActiveCell.Offset(-1, 1).Select
ActiveSheet.Paste
Selection.Font.Bold = True
Selection.Font.Underline = xlUnderlineStyleSingle
Selection.Font.Size = 11
ActiveCell.Offset(1, 0).Select

    inicio = ActiveCell.Row
    Selection.End(xlDown).Select
    final = ActiveCell.Row
    ActiveCell.Offset(1, 1).Select
    ActiveCell.Formula = "=SUM(R" & inicio & "C:R" & final & "C)"
        With Selection.Borders(xlEdgeTop)
            .LineStyle = xlContinuous
            .ColorIndex = 0
            .TintAndShade = 0
            .Weight = xlThin
        End With
        With Selection.Borders(xlEdgeBottom)
            .LineStyle = xlContinuous
            .ColorIndex = 0
            .TintAndShade = 0
            .Weight = xlMedium
        End With
    ActiveCell.Offset(0, -1).Select
End If
Loop

'Sub Macro18()

    Columns("A:B").Delete Shift:=xlToLeft

'End Sub

'Sub Macro19()

    Range("B1").Select
    Selection.End(xlDown).Select
    primerpunto = ActiveCell.Row

    Application.Goto Reference:="R1048576C2"
    Selection.End(xlUp).Select
    Selection.ClearContents
    ActiveCell.Offset(-1, 0).Select
    segundopunto = ActiveCell.Row

'End Sub
'Sub Macro21()

    Range("B1048576").Select
    Selection.End(xlUp).Select
    ActiveCell.Offset(1, 0).Select
    'Range("B27").Select
    'ActiveCell.FormulaR1C1 = "=SUMIF(R[-23]C[-1]:R[-1]C[-1],"",R[-23]C:R[-1]C)"
    ActiveCell.FormulaR1C1 = _
    "=SUMIF(R" & primerpunto & "C[-1]:R" & segundopunto & "C[-1],"",R" & primerpunto & "C:R"
    & segundopunto & "C)"
    Selection.Font.Bold = True
    Selection.NumberFormat = "#,##0.00_);[Red](#,##0.00)"

End Sub

```


Ok. Creo que quedó suficientemente claro, como conociendo estos trucos:

- Saber ir hasta el final de la información de Excel
- Almacenar los puntos como variables
- Hacer los recorridos Con For Next o Do Until
- Usar el grabador de macros

Ya es mucho lo que puede hacer por su cuenta. Si se fija, cualquier reporte que le haga falta, puede programarlo POR USTED MISMO, un reporte de producción, un reporte gerencial, notas a los Estados financieros, Los mismos Estados Financieros, Cédulas Sumarias, hacer cruces de información, etc.

Las bases ya las tiene aquí. Solo me resta mostrarle algunas técnicas para convertir los archivos planos en Bases de Datos, que es una recopilación de las técnicas vistas en este libro. Punto importante pues un Auditor, un Contador o un Administrador, normalmente encuentra clientes o trabajo con diversos software, y no todos tienen soluciones que exporten la información limpia a Excel, y si las tuvieran, aun así necesitarían saber algo de macros para maximizar el rendimiento de esas aplicaciones.

En este punto, si encuentra que le gusta el tema, puede continuar investigando por su cuenta, buscar en otros libros, o internet y poco a poco, volverse todo un GURÚ o una GURÚ del Excel.

Algunas técnicas para depurar los archivos planos

Ok. Algunos programas contables como el Cg1, uno de los mas populares en Colombia, están hechos en Cobol, y para suplir la necesidad de la exportación a Excel, se han creado por allí algunas herramientas que se incorporan como complemento de Excel. Un ejemplo es el aplicativo llamado BIABLE. El Biable no Exporta del sistema Cg1 a Excel, sino todo lo contrario, hace que Excel penetre en una base de datos que contiene la información contable y Excel toma los datos de allí. Osea, que se debe de correr previamente un proceso desde del sistema CG1 que carga los datos del programa Cobol, del CG1 al Biable, que está hecho en Oracle. Si no se hace esta actualización o cargue de información, El complemento leerá datos desactualizados, o en el peor de los casos, si está mal parametrizado el cargue de información, no los leerá.

El Biable es un buen soft, aunque tengo las siguientes observaciones:

- El complemento se instala por máquina, con lo cual, por lo general se instala en la computadora del Contador o Gerente o Financiero.
- Al estar centralizado en la computadora del Contador, es una excusa que usan las demás áreas para "delegarle" algunas tareas extra al Contador, con lo cual, tras de que mantiene escaso de tiempo, pues debe sacar tiempo para atender los requerimientos de los demás.
- El costo no facilita que todos los usuarios (mas que todo las pequeñas empresas) tengan el complemento BIABLE. Si tienes un papá con la posibilidad, pídele que le coloque Biable no solo a ti, sino al Gerente administrativo y a quien le haga falta.
- La operación de cargue de datos, directamente de la base a Excel, hace que se consuman recursos de memoria y procesador por lo cual, si no se tiene una buena computadora, el Excel se puede quedar un tiempo no pequeño, calculando y cargando la información. Si tienes otros programas abiertos, el cargue de información será mas lento aún. Sabemos que un Contador no se puede dar el lujo de tener que soportar procesos lentos. Es para volverse loco en pleno cierre contable.

De todas formas, si el Contador no maneja buen Excel, no explotará las bondades del BIABLE. Por otra parte el Auditor no tiene acceso (tampoco debe de contar con que se la van a dar) a tener la información directamente del BIABLE. El Biable trabaja con Funciones propias, que para que operen, debe de estar habilitada la conexión a la base de datos, y si te llevas una hoja de cálculo en una memoria USB, en cuanto abras tu hoja de cálculo, verás una gran cantidad de errores de Excel de tipo #¿NOMBRE?, que es el tipo de error que muestra Excel cuando no reconoce el texto de la fórmula:

	<u>Cuenta</u>	<u>CONCEPTO</u>
10		
11		
12	415505	#¿NOMBRE?
13	41559503	#¿NOMBRE?
14	41559501	#¿NOMBRE?
15	41352405	#¿NOMBRE?
16		

Y es lógico, pues son funciones propias del programa que se ejecutan en donde está el programa instalado. La solución es pedir que le hagan una copia de la hoja de cálculo pero con solo valores.

De todas formas, el auditor debe de estar SEGURO de que las cifras que le están mostrando son ciertas o nó. Es por esto que lo mejor, en mi opinión es pedir los listados impresos, por lo menos la primera y la última hoja si son muy voluminosos, y pedir ese mismo listado como archivo plano, para que lo pueda trabajar posteriormente en Excel. Así que es posible, mediante los archivos planos, transformar la información del archivo en una base de datos, mediante algunas técnicas de Excel, muchas de las cuales ya hemos visto en este libro. Ahora las veremos en un ejemplo como todas las técnicas trabajan

para este fin: depurar la información de los archivos planos y ya con ellos organizados, comenzar a verificar cositas de interés. El Biable no puede ser una objeción para no aprender buen Excel. Finalmente tendrá que hacerlo algún día.

Depurando un balance de comprobación por terceros.

Bueno, este libro le sugiere una manera de trabajar la información. No es algo de lo cual se deba pegar al pie de la letra de manera ortodoxa, en Excel cada uno puede "matar sus pulgas" de la mejor manera que estime conveniente. Una vez que se muestra una manera, es posible que se ilumine nuestro maravilloso pensamiento y se encuentre una manera alternativa, la mayoría de las veces mejor, de hacer las cosas. Incluso a mi mismo me sucede cada cierto periodo de tiempo, en ocasiones me doy cuenta que he podido hacer tal o cual tarea de una manera mas eficiente. (A veces me doy cuenta que la puedo hacer de otra manera y me cuesta un poco romper el "hábito" de trabajarla a la vieja usanza. Esto es porque el cerebro es el órgano que más energía consume, así que siempre opta por la opción más barata: la rutina que ya ha mecanizado). Pero es debido al conocimiento de nuevas técnicas, y las mismas son incorporadas a los viejos procedimientos, haciéndolos mucho mas rápidos. Es la misma evolución del KnowHow en acción.

En la apertura de los archivos planos, lo que he recomendado es que, en cuanto se abra el archivo plano con Excel, veremos que Excel abre el asistente de importación, y lo que indico es que se incorpore todo en una sola columna. Esto, *en mi opinión*, facilitará limpiarle toda la basura que no es información transaccional de manera muy rápida.

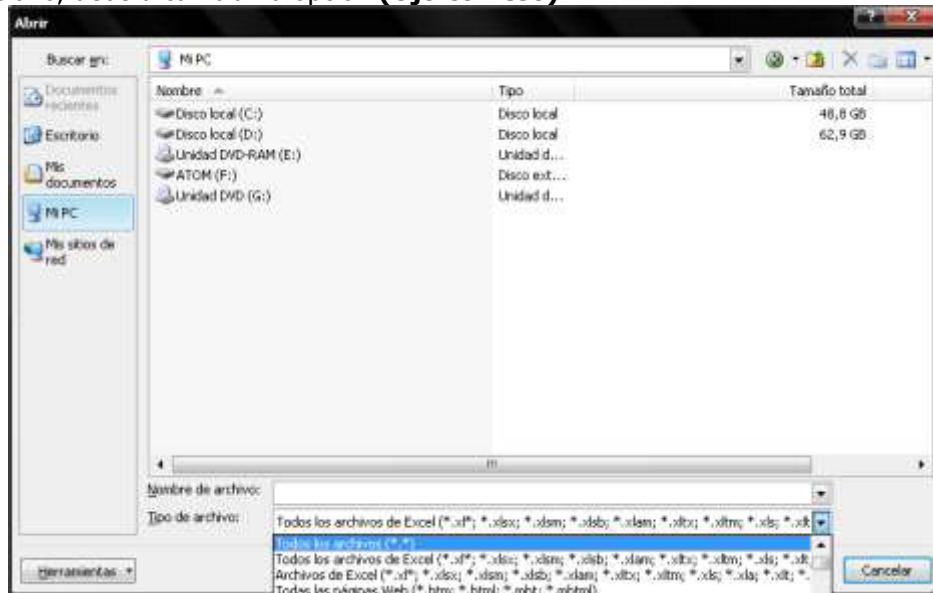
Este tipo de reporte, por lo general tiene por nombre en el sistema CG1 version 5, CG03013.P01. Pues bien, solicite un listado Balance de comprobación por terceros **con totales por abajo**.

El listado con totales por arriba le presenta el inconveniente de que le duplica alguna información, y esto es así, porque como lo que usted tiene es un LISTADO que estaba destinado a IMPRIMIRSE, pues en las impresoras de punto, las clases, grupos y subgrupos se muestran en negrilla, y lo que hace el sistema es hacer que la impresora REPITA la impresión, colocando los registros DOS VECES. Por esto es que es mejor pedir este listado con totales por abajo, así no tendremos que eliminar los datos repetidos en el procedimiento, ahorrándonos un pasito.

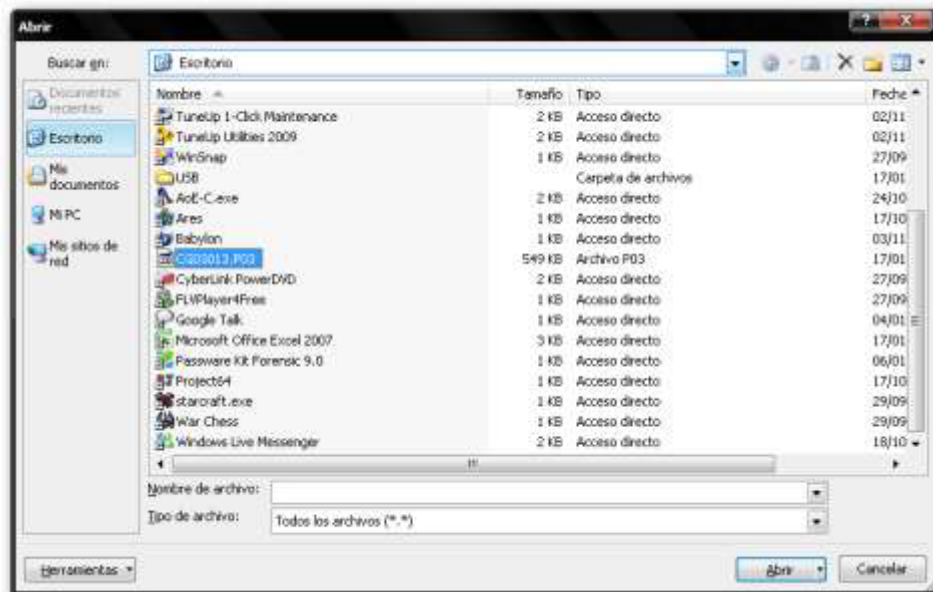
Cuando el listado CG03013 lo genere en el sistema CG1, y lo esté viendo por pantalla, no lo imprima ni lo mande como hoja Excel. Envíelo por la opción DISCO. Y tome nota del nombre del archivo. Debe llamarse CG03013.P01 o P02, la extensión la puede definir el usuario, y por lo general se cambia la extensión P01 por P02, si otro usuario está mirando por pantalla el mismo reporte en otra computadora terminal. Estos listados por lo general quedan almacenados en una carpeta del programa, por lo general es la PRT (PRT es la abreviatura de Printer - Impresora). Si no lo ubica fácilmente, haga lo siguiente: Use el Buscar archivos o carpetas de Windows y ponga a que el sistema le ubique en donde está. Tenga en cuenta que si el programa está alojado en un servidor, es decir en otro computador y si solo busca en los discos duros de **su** computadora, no los encontrará. Si es así, busque en el Computador Host que comúnmente llaman Servidor, en el cual se encuentra instalado el programa contable CG1. Una vez que Windows lo encuentre, verifique que sea, mirando por la opción detalles de archivo, que el archivo sea el que se generó en la hora y fecha en que Usted lo pidió, es importante pues a veces hay más de un archivo, y se puede confundir de archivo. (Ojo con eso), los detalles quedan asociados a todos los archivos, y al seleccionar la opción Detalles, puede ver ese tipo de información, también llamada "Meta Datos".

Bien. Una vez haya encontrado el archivo, si es auditor, pida que se lo graben a una Memoria USB para que lo pueda trabajar posteriormente, si es el Contador, (**y no tiene BIABLE**), haga una copia y colóquelo en sus documentos, o en donde considere apropiado. Seguidamente, vaya a Excel, Abra

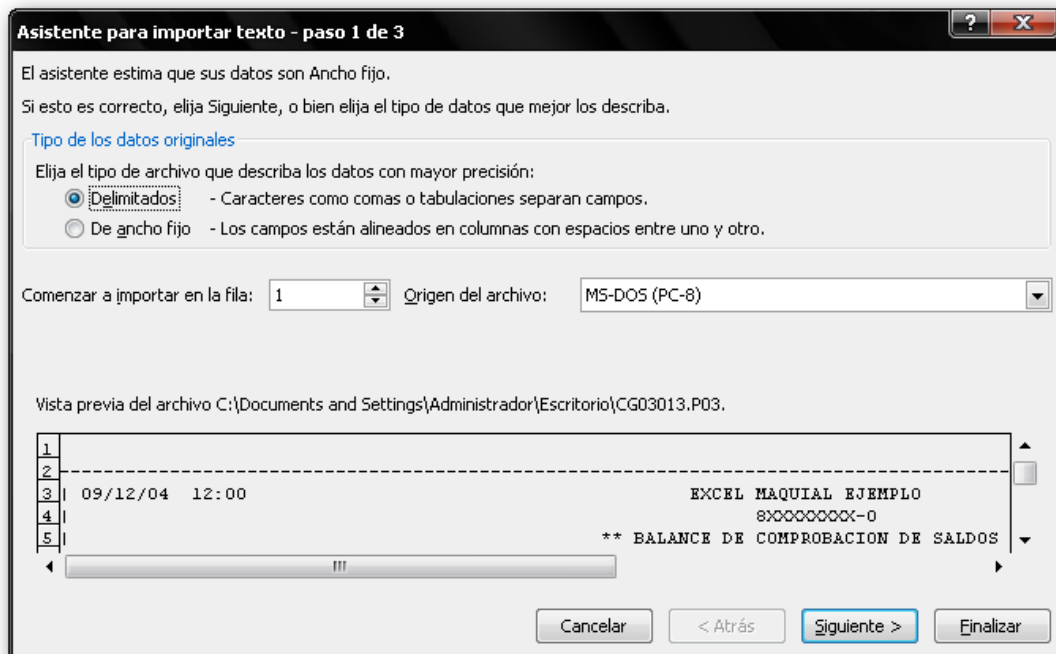
Excel, y con el botón ABRIR, busque el archivo plano. Asegúrese de que puede ver TODOS LOS ARCHIVOS. Por lo general Excel le muestra TODOS LOS ARCHIVOS DE EXCEL. Pero como está abriendo un archivo plano, deberá cambiar la opción **(Ojo con eso)**:



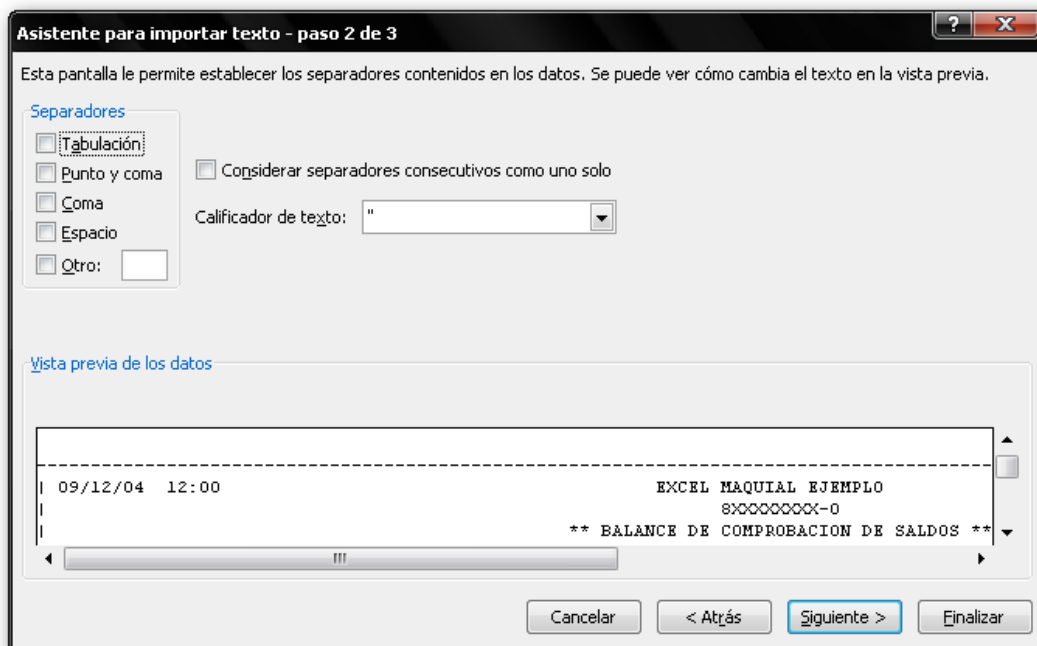
Como en mi caso, la copia la puse en el escritorio, pues lo busco allí:



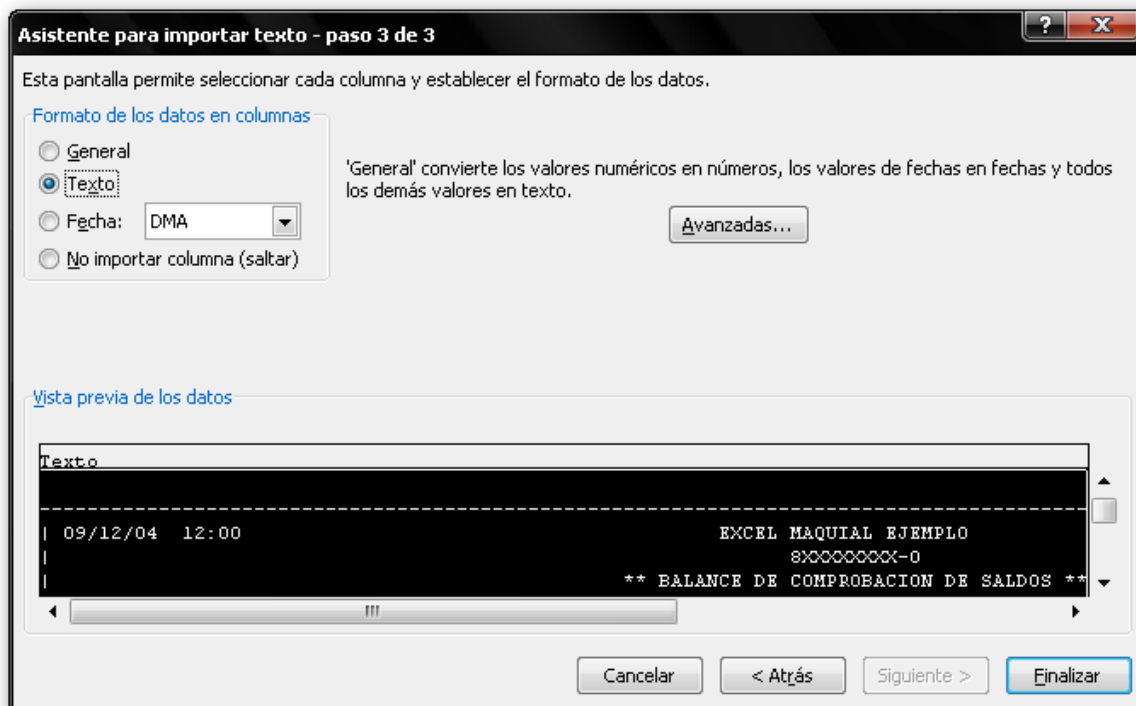
En cuanto haga clic en abrir, saldrá el asistente para importar texto:



Hago clic en Delimitados, y clic en Siguiente.
Desactivo la casilla de verificación del tipo de Separador:



Clic en siguiente.



Indico que todo es tipo Texto, y clic en Finalizar:

	A	B	C	D	E	F	G	H
1								
2	-----							
3	09/12/04 12:00			EXCEL MAQUIAL EJEMPLO				
4				8XXXXXXXX-0				
5				** BALANCE DE COMPROBACION DE SALDOS **			PAG : 1	
6								
7			2018/01	ENERO 2.018			NIVEL: 09	
8			2018/12	DICIEMBRE 2.018				
9	-----							
10	CUENTA	DESCRIPCION		SALDO ANTERIOR	DEBITOS	CREDITOS		NUEVO SALDO
11	-----							
12	1	ACTIVO						
13	11	DISPONIBLE						
14	1105	CAJA						
15	110505	CAJA GENERAL						
16			0.00	11,833,098,872.66	11,833,098,872.66		0.00	
17	110510	CAJAS MENORES						
18	11051001	CAJA MENOR BOGOTA						
19			1,500,000.00	0.00	0.00	1,500,000.00		
20	-----							
21	-----							
22	****	CAJAS MENORES		1,500,000.00	0.00	0.00		1,500,000.00
23	****	CAJAS MENORES		1,500,000.00	0.00	0.00		1,500,000.00
24	110515	MONEDA EXTRANJERA						
25			0.00	3,477,000.00	111,615.00	3,365,385.00		
26	-----							

Justifico la columna A:

CUENTA	DESCRIPCION	SALDO ANTERIOR	DEBITOS	CREDITOS	NUEVO SALDO
1	ACTIVO				
11	DISPONIBLE				
1105	CAJA				
110505	CAJA GENERAL	0.00	11,833,098,872.66	11,833,098,872.66	0.00

Puede ver que el archivo plano quedó "Empaquetado" en una sola columna. Esto me permitirá:

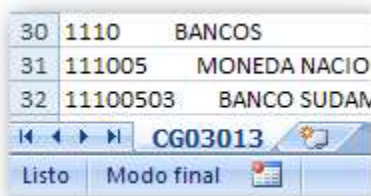
- Usar la opción de seleccionar las celdas en blanco para eliminarlas de una sola vez.
- Usar los filtros para depurar los encabezados del listado. Si el listado es demasiado grande, y Excel se niega a eliminar la selección del filtro, usar:
- La opción de ordenar la información y allí si, eliminarla rápidamente.

Después de abrir el archivo, coloco un título cualquiera a la celda A1, pues **voy** a colocar (ojo todavía no) un filtro.

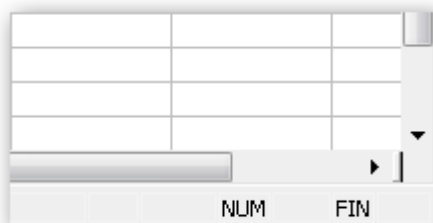
OJO: DADO que es posible que existan filas en BLANCO, si coloca el Filtro en este paso, el filtro NO TOMARÁ todo el compendio de información, no tomará todo el archivo plano, sino, solo un rango, por eso, debe hacer el siguiente paso:

Siguiente Paso: tomar el rango COMPLETO del archivo plano. Como? Me paso a la celda B1. Oprimo la tecla del computador FIN una sola vez. Cuando se activa la celda FIN, en Excel 2007 verá que abajo a la izquierda dice "Modo Final" si está en Excel 2003, verá un "FIN" que se activa en la extrema derecha inferior:

Excel 2007:



Excel 2003

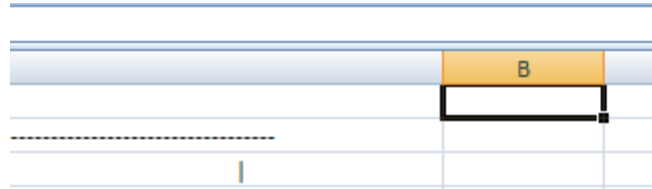


OK. Recuerde que debe de oprimir FIN UNA SOLA VEZ. Si la oprime una segunda vez, YA NO VERÁ "Modo Final", ni "FIN". Se desactivarán. Y necesitamos activar la opción para IR en un solo instante al FINAL DE LA HOJA DE CALCULO. Una vez allí, nos pasamos de nuevo a la columna A, oprimimos FIN

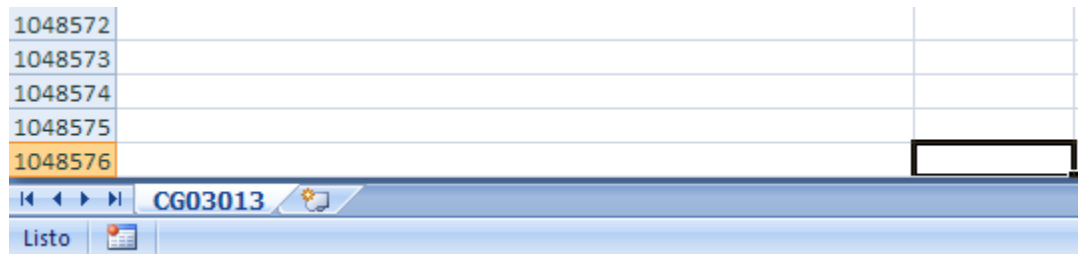
otra vez, y oprimimos la tecla Flecha Arriba. Con esto, estaremos en la parte final del archivo plano, **y ya podremos tener la seguridad de que lo vamos a Seleccionar Completamente.**

Entonces: le ilustro todo lo anterior paso a paso:

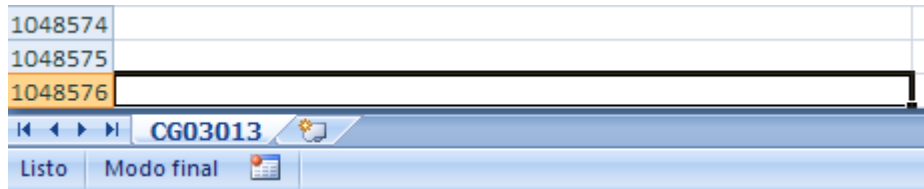
1. Que pasa cuando me paso a la celda B1:



2. Que pasa cuando hago FIN y luego oprimio la flecha hacia ABAJO:



3. Me paso a la columna A nuevamente y oprimio FIN:



4. Con FIN activado, Oprimio Flecha hacia arriba:

	A				
4569	** COSTOS INDIRECTOS	0.00	4,869,377,996.88	4,869,377,996.88	0.00
4570					
4571					
4572	* COSTOS DE OPERACION Y PRO.	0.00	6,722,012,631.11	6,722,012,631.11	0.00
4573	* COSTOS DE OPERACION Y PRO.	0.00	6,722,012,631.11	6,722,012,631.11	0.00
4574					
4575					
4576	TOTAL... G E N E R A L	0.00	174,514,022,592.37	174,514,022,592.37	0.00
4577	TOTAL... G E N E R A L	0.00	174,514,022,592.37	174,514,022,592.37	0.00
4578					
4579					

Excel se "Estrella" con la parte final del listado.

5. Ahora, desde aquí, TOMAREMOS TODO EL RANGO QUE CONFORMA EL ARCHIVO PLANO:
Como? Oprimiendo SIMULTANEAMENTE desde este punto las teclas Control + Shift + Inicio
Pasará que se seleccionará TODO el rango del archivo plano:

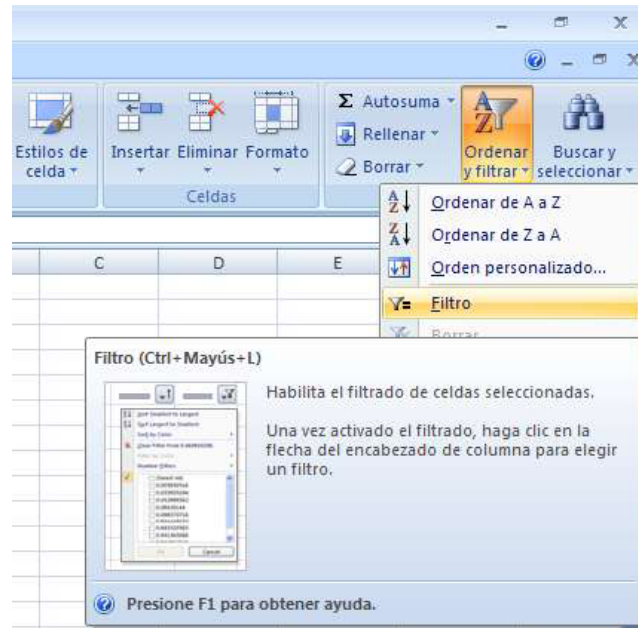
A					
1	dffsdsdffffs				
2	-----				
3	09/12/04 12:00	EXCEL MAQUIAL EJEMPLO			
4		8XXXXXXXX-0			
5		** BALANCE DE COMPROBACION DE SALDOS **		PAG : 1	
6		-----			
7		2018/01	ENERO 2.018	NIVEL: 09	
8		2018/12	DICIEMBRE 2.018		
9		-----			
10	CUENTA	DESCRIPCION	SALDO ANTERIOR	DEBITOS	CREDITOS
11					NUEVO SALDO
12	1	ACTIVO			
13	11	DISPONIBLE			
14	1105	CAJA			

NOTA:

Porque no simplemente selecciono la columna A y coloco el filtro? Todo esto que le he dicho se ahorra con solo seleccionar la columna A y hacer el filtro. PERO... Recuerde que Excel 2007 ya no maneja las viejas 65 mil celdas... Maneja UN MILLON CUARENTA Y OCHO MIL Y "PUNTA" de celdas, entonces, serán más de UN MILLON CINCUENTA MIL CELDAS EN BLANCO que Excel Cargará en la memoria, con lo cual le DISMINUIRÁ TREMENDAMENTE LOS RECURSOS DE MEMORIA A LA COMPUTADORA. No obstante, lo anterior, puede hacerlo y ver como es el rendimiento de su computadora. Si encuentra que no es significativo, pues seleccione la columna entera. SINÓ, ya sabe cómo seleccionar solo el rango y así su computadora lenticia NO SUFRIRA! Que la computadora no sea un pretexto que torpedee tu labor.

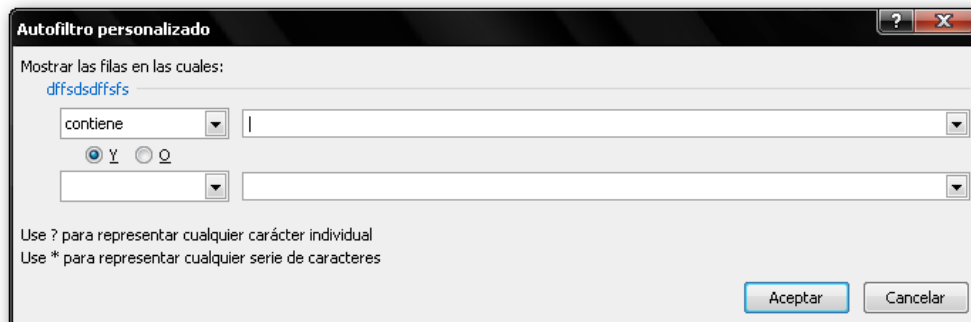
6. **Ahora si,** Con la región SELECCIONADA, voy por el menú INICIO, sección de la derecha "modificar" y ACTIVO el filtro:





Ok, ahora, si, comenzamos a usar los filtros para limpiar la basura de los encabezados.

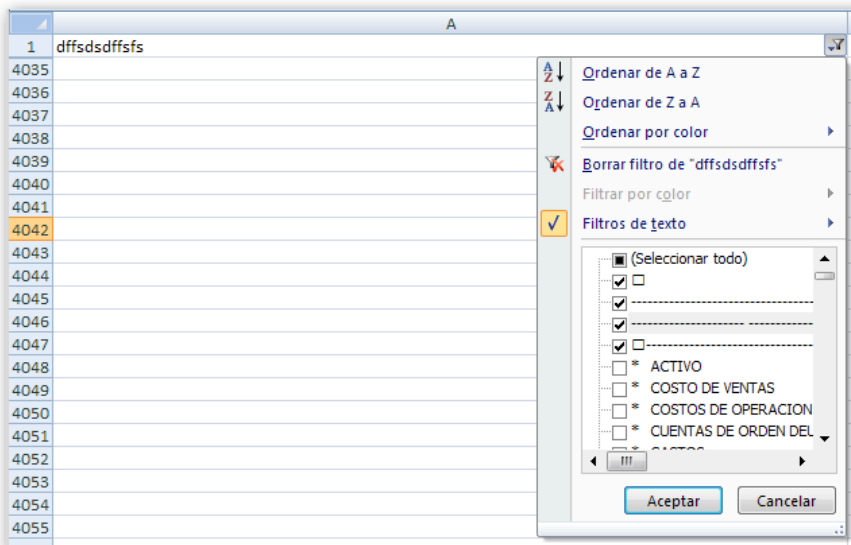
Primero que todo me deshago de esos encabezados que contienen una raya vertical: |



(Recuerde que esta rayita tipo palito vertical ya vimos como eliminarla detenidamente en el tema de los filtros.)

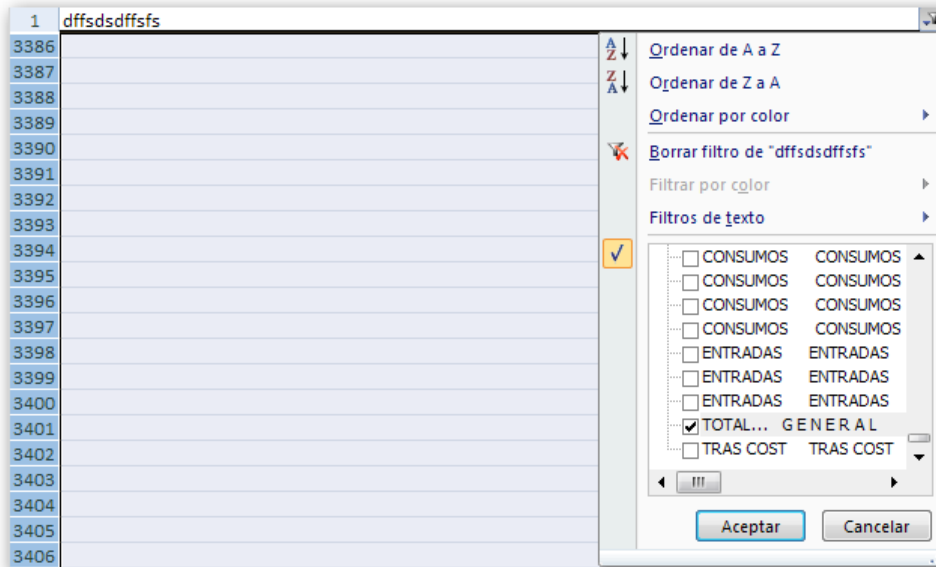
Doy aceptar, selecciono las filas desde la que aparece debajo del título hasta el final y las borro.

Ahora, cuando se eliminen, NO SE PREOCUPE SI NO VE NADA, el resto de la información sigue allí, solo debe de ir nuevamente al filtro y seguir seleccionando la basura de los encabezados que no nos sirve:



En Excel 2003 solo podías seleccionar 2 filtros... aquí si la tabla no es gigante, puedes filtrar bastantes:

Toda esta basura la eliminas. Luego vamos por el total general...aunque son pocas líneas las podemos eliminar...de una vez.

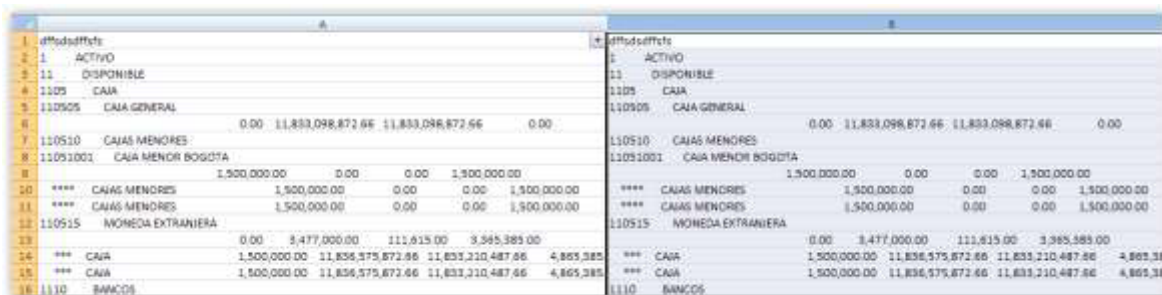


	A				
1	dffsdsdffsf				
3384	TOTAL... G E N E R A L	0.00	174,514,022,592.37	174,514,022,592.37	0.00
3385	TOTAL... G E N E R A L	0.00	174,514,022,592.37	174,514,022,592.37	0.00

Al mostrar nuevamente todas las celdas, el archivo nos va quedando así:

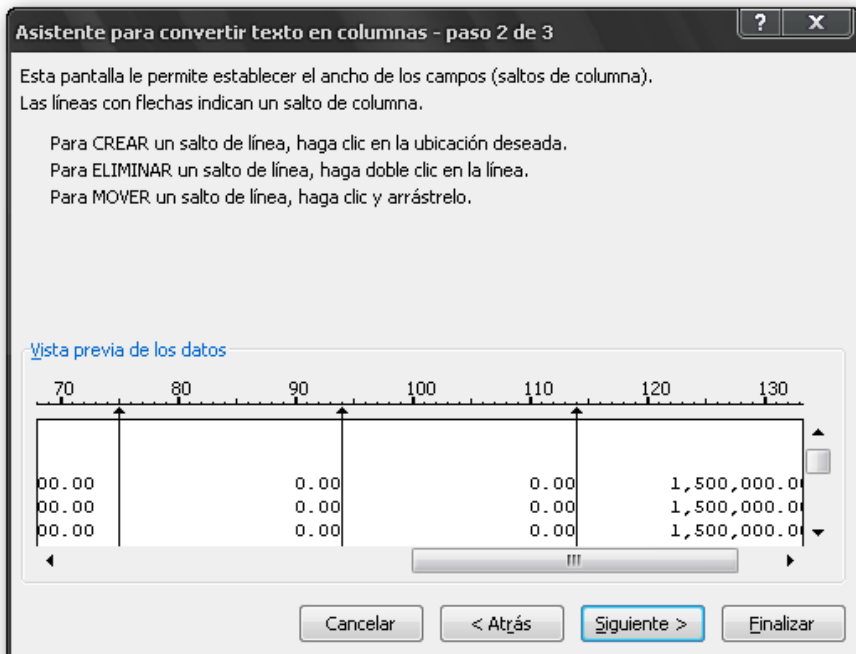
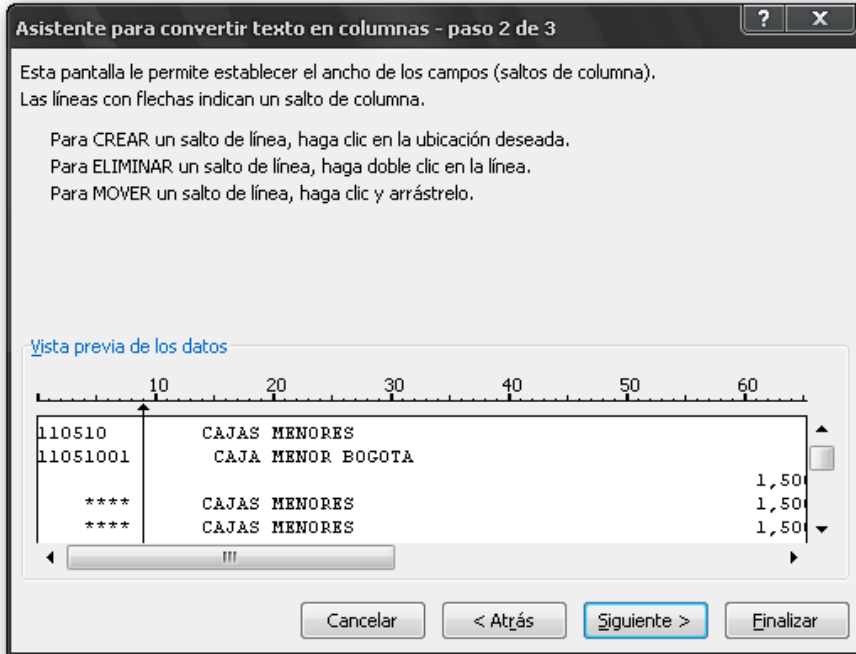
	A					
1	dffsdsdffsf					
2	1	ACTIVO				
3	11	DISPONIBLE				
4	1105	CAJA				
5	110505	CAJA GENERAL				
6		0.00	11,833,098,872.66	11,833,098,872.66	0.00	
7	110510	CAJAS MENORES				
8	11051001	CAJA MENOR BOGOTA				
9		1,500,000.00	0.00	0.00	1,500,000.00	
10	****	CAJAS MENORES	1,500,000.00	0.00	0.00	1,500,000.00
11	****	CAJAS MENORES	1,500,000.00	0.00	0.00	1,500,000.00
12	110515	MONEDA EXTRANJERA				

Viene el trabajito de deshacernos de esos asteriscos. Los asteriscos pueden ser un problema, pues si se fija, para Excel un asterisco puede ser cualquier carácter, así que si escoge eliminar *** tres asteriscos, Excel le eliminará todo lo que haya por allí (todo es TODO). Para deshacernos de ellos, lo que hago es copiar toda la columna A en la columna B, y usar la opción texto en columnas, así:

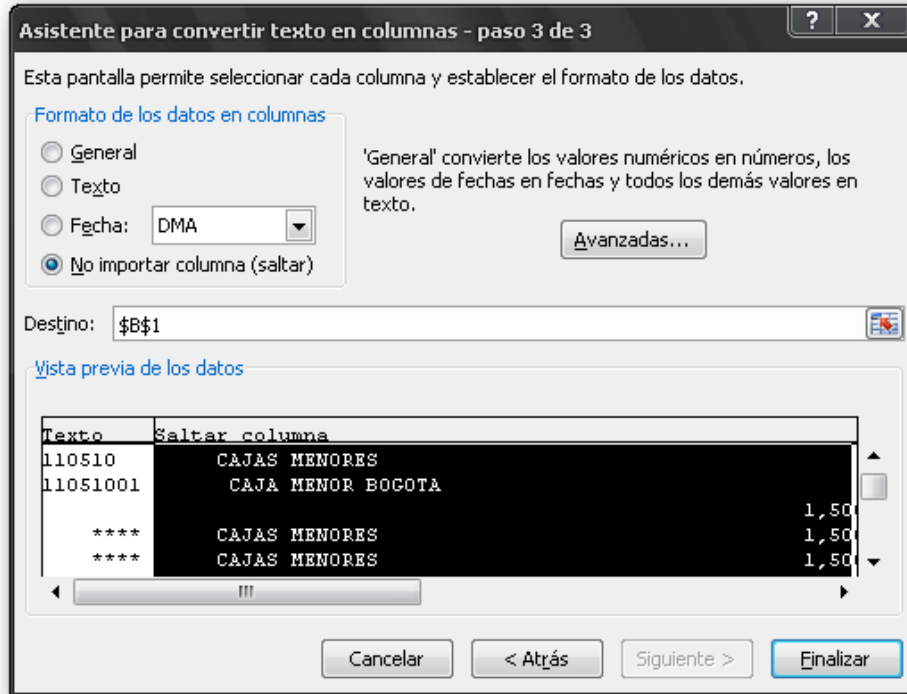


(En el tema siguiente, cuando todo esto lo grabemos en una macro, haremos este paso de otra forma, hay muchas formas de hacer las mismas cosas en Excel, no lo olvide!)

Con la columna B marcada, voy a la herramienta texto en columnas, opción de ancho fijo y solo divido la parte en donde se encuentran los asteriscos, el resto de info de la columna B, elimino las rayas para dejar el resto en un solo bloque y en el paso siguiente, la primera parte le pondré tipo texto y a la segunda parte le pondré NO IMPORTAR:



Recuerde que debe desplazarse con la barra horizontal a la derecha Y ELIMINAR esas rayas, de lo contrario Excel las traerá y las colocará en columnas, y no las queremos, pues no olvide que por ahora queremos deshacernos de las líneas que tienen asteriscos:



Clic en Finalizar, y justifica el ancho de la columna B:

	A	B
1	dfdsdfds	dfdsdf
2	1 ACTIVO	1
3	11 DISPONIBLE	11
4	1105 CAJA	1105
5	110505 CAJA GENERAL	110505
6	0.00 11,833,098,872.66 11,833,098,872.66 0.00	
7	110510 CAJAS MENORES	110510
8	11051001 CAJA MENOR BOGOTA	11051001
9	1,500,000.00 0.00 0.00 1,500,000.00	
10	**** CAJAS MENORES 1,500,000.00 0.00 0.00 1,500,000.00	****
11	**** CAJAS MENORES 1,500,000.00 0.00 0.00 1,500,000.00	****
12	110515 MONEDA EXTRANJERA	110515

Ahora, colocamos los filtros nuevamente, y por la columna b, llamamos todos los asteriscos y los eliminamos, tal y como hemos hecho con el resto de la basura que queremos borrar.

	A	B
1	dffsdsdffsf	dffsdsdff
2	1 ACTIVO	
3	11 DISPONIBLE	
4	1105 CAJA	
5	110505 CAJA GENERAL	
6		
7	110510 CAJAS MENORES	
8	11051001 CAJA MENOR BOGOTA	
9		
10	**** CAJAS MENORES	
11	**** CAJAS MENORES	
12	110515 MONEDA EXTRANJERA	
13		
14	*** CAJA	
15	*** CAJA	

No se confunda con la imagen, aquí escogí el filtro de la columna B, y escogí todos los asteriscos:

	A	B
10	**** CAJAS MENORES	****
11	**** CAJAS MENORES	****
14	*** CAJA	***
15	*** CAJA	***
26	**** MONEDA NACIONAL	****
27	**** MONEDA NACIONAL	****
31	**** MONEDA EXTRANJERA	4****
32	**** MONEDA EXTRANJERA	4****

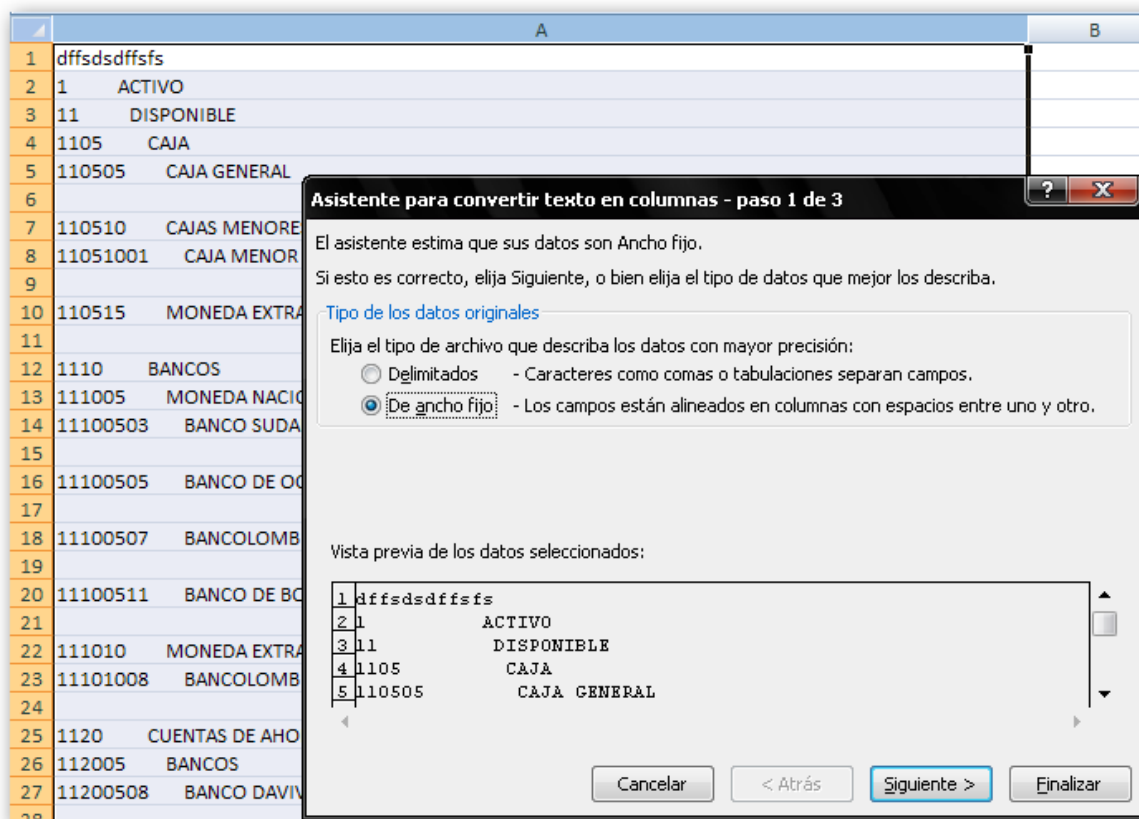
Fíjese que aparece el símbolo del EMBUDITO en la columna B. Esto es porque hice el filtro en la columna B.

Elimina todas esas filas y vuelve a seleccionar todo, para que vea todos los registros, y así, ver que otra basura hacer falta borrar:

	A	B
1	dffsdsdffsf	dffsdsdff
2	1 ACTIVO	1
3	11 DISPONIBLE	11
4	1105 CAJA	1105
5	110505 CAJA GENERAL	110505
6		
7	110510 CAJAS MENORES	110510
8	11051001 CAJA MENOR BOGOTA	11051001
9		
10	110515 MONEDA EXTRANJERA	110515
11		
12	1110 BANCOS	1110
13	111005 MONEDA NACIONAL	111005
14	11100503 BANCO SUDAMERIS COLOMBIA	11100503

Esa columna B, ya no la necesitamos. También quitamos el filtro. Después de eliminar la columna B, procedemos a partir (dividir) con la herramienta texto en columnas, la columna principal, esto es, la

columna A: Marco o selecciono toda la columna A y llamo nuevamente la herramienta texto en columnas:

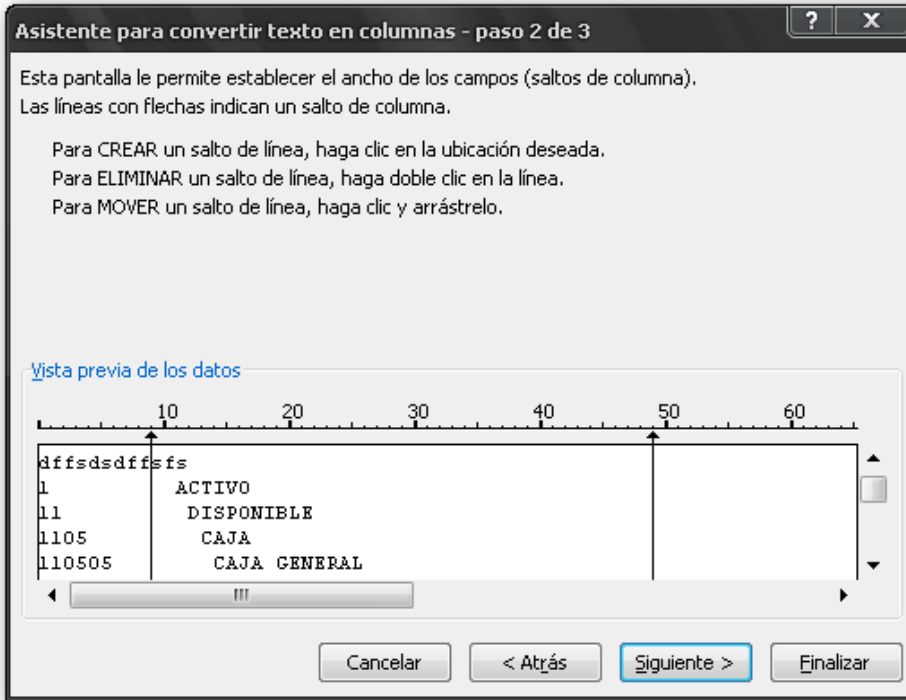


OJO Esta parte es delicada, pues si partes o divides mal las columnas Saldo inicial, débitos, créditos y saldo final, tendrás que volver a empezar, a menos que quieras copiar nuevamente la columna A en B, y partir la columna B. Es recomendable, para no perder todo lo hecho hasta el momento.

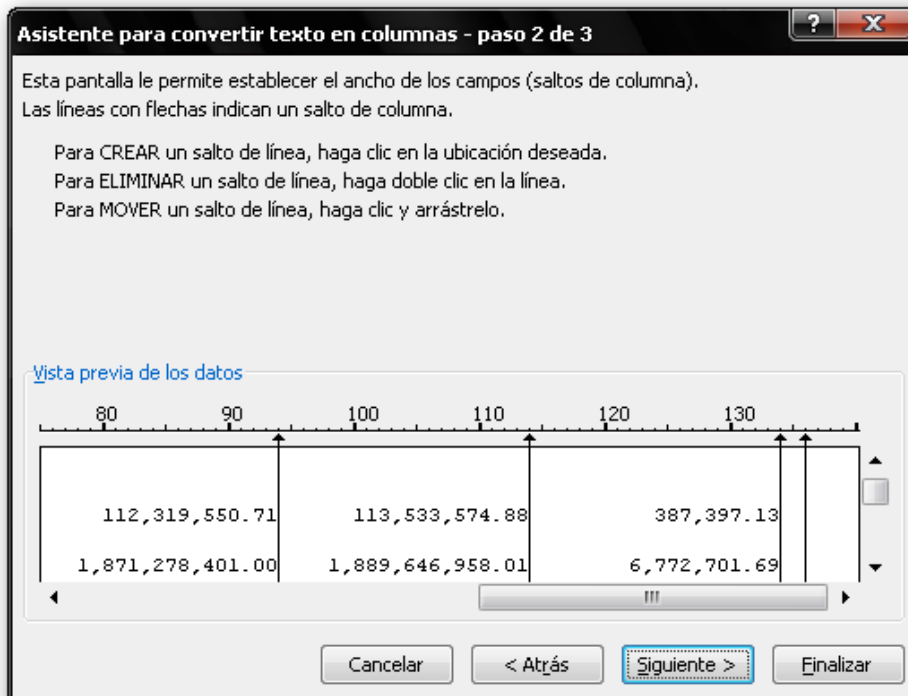
Tenga en cuenta que la opción deshacer en la herramienta TEXTO EN COLUMNAS NO FUNCIONA CORRECTAMENTE.

También se debe tener en cuenta que este archivo, a los valores crédito les pone un texto CR, que también hay que separar.

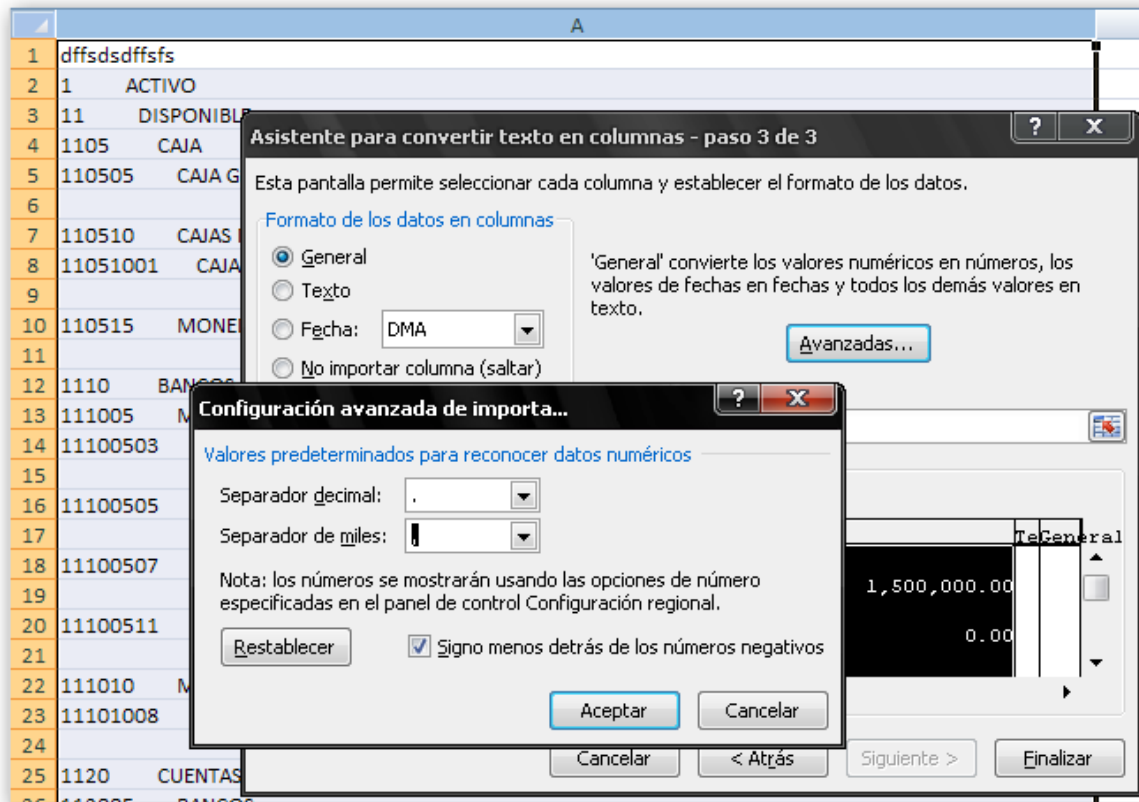
La partición queda así, para la primera parte:



Así queda para la segunda parte: (ojo que aquí solo se ven debito, crédito y saldo final. Asegúrese de separar el CR de Saldo inicial, que aquí no se ve, así como quedaron las dos rayitas de la parte final, en donde está el saldo final):



Fíjese en la regleta que aparece allí. En el saldo inicial coloque un par de rayas en la posición 73 y 75. Ahora, todos los campos los defino como texto. Si se conoce bien el comportamiento de su equipo en el separador de miles y de decimales, puede usar la opción avanzadas:



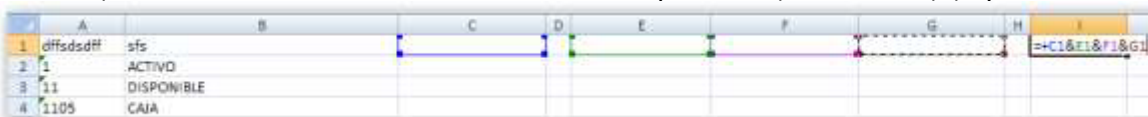
En esta opción se coloca el separador de miles y de decimales que tiene el archivo plano, para que Excel lo convierta en el que maneja el computador, que por lo general en estos archivos es al revés.

Sigamos: Tenga en cuenta que este es un listado balance de comprobación por terceros, **y los terceros están en la misma columna que las cuentas:**

	A	B	C	D	E	F	G	H
43	13	DEUDORES						
44	1305	CLIENTES						
45	130505	NACIONALES						
46	900200514	ANITO DE TAL		0	24.336.000,00	24.336.000,00		0
47	860006333	MENGANITO DE TAL		15.243.750,00		0	15.243.750,00	0
48	900015051	SUTANITO DE TAL		0	28.102.500,00	23.197.500,00		4.905.000,00

La cuenta DEBE estar en la misma Línea de la transacción, que ya tiene el código y el nombre del tercero, tenemos que **REUBICARLA** en esa misma línea. Cómo?

Fíjese que en la figura se ve una situación constante: en donde hay cuentas no hay valores saldos inicial, débitos, créditos y saldo final. Están en blanco. Por allí las podemos detectar: Entonces, en la columna I1, concatenamos o unimos las celdas con un ampersand &, las celdas C,E,F y G:



Para qué? Para detectar lo que REALMENTE ESTA EN BLANCO, y lo que una vez unido, quede en blanco, indudablemente serán las líneas de los número de las cuentas. Esta fórmula similar a

=Concatenar() se copia hasta el final del archivo plano, luego se desbarata la función que nos concatenó todo con un copiar, pegado especial, solo valores.

Entonces, recapitulando: Dado que los NITS o Códigos, o terceros están en la misma columna que las cuentas, hace falta separar las cuentas, colocarlas en columnas aparte, pues necesitamos esos dos datos por aparte, en la misma fila de las transacciones.

Como ya detectamos que en donde están las cuentas no hay saldos sino celdas en blanco, y ya las tenemos ubicadas con el concatenar, entonces las podemos separar fácilmente, ordenando la información por esta nueva columna I que hemos creado. Pero para no perder el orden original, insertamos una columna en A, y colocamos un número serial. Es la técnica para depurar la información con ordenar que le enseñé al principio del libro. Si no hacemos esto, podemos perder la información pues no sabríamos como es que estaba ordenada originalmente:

	A	B	C
1	1	dfdsdfff	sfs
2	2	1	ACTIVO
3	3	11	DISPONIBLE
4	4	1105	CAJA
5	5	110505	CAJA GENERAL
6	6		
7	7	110510	CAJAS MENORES
8	8	11051001	CAJA MENOR BOGOTA
9	9		
10	10	110515	MONEDA EXTRANJERA
11	11		

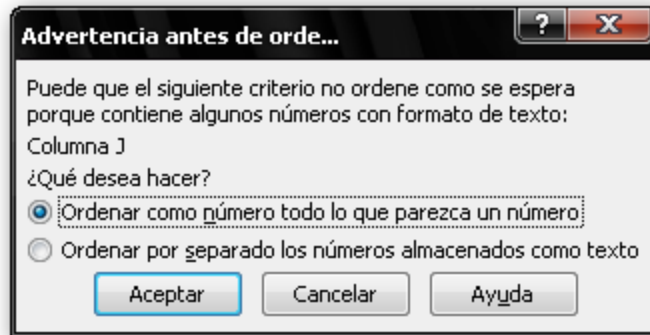
Insertamos otras dos columnas en B y C. les colocamos cualquier cosa como encabezado **para que Excel las integre en la región corriente de la tabla**. En estas dos columnas que hemos ABIERTO, que hemos INSERTADO, TRASLADAREMOS las cuentas y los nombres de las cuentas, así:

	A	B	C	D	E	F
1	1			dfdsdfff	sfs	
2	2			1	ACTIVO	
3	3			11	DISPONIBLE	
4	4			1105	CAJA	
5	5			110505	CAJA GENERAL	
6	6					0
7	7			110510	CAJAS MENORES	
8	8			11051001	CAJA MENOR BOGOTA	
9	9					1.500.000,00
10	10			110515	MONEDA EXTRANJERA	
11	11					0
12	12			1110	BANCOS	
13	13			111005	MONEDA NACIONAL	
14	14			11100503	BANCO SUDAMERIS COLOMBIA	
15	15					1.601.421,30

Ahora, después de haber insertado el par de columnas, colocamos cualquier cosa como encabezado de columna en B1 y C1, **y ordenaremos todo por la columna de las celdas que identificamos que realmente están vacías, recuerda? Fíjese que sea la columna en donde montamos la función concatenar con el & Ampersand, la columna en donde unimos las celdas de los**

saldos, débitos y créditos. Tenga presente que como se acabó de insertar un par de celdas en B y C, esa columna se desplazó de J a L :

A la pregunta:



Dele "Ordenar por separado los números almacenados como texto". Ojo, la segunda opción, no la que se ve aquí en la imagen. Así, las celdas vacías quedarán arriba, y con ellas, si se fija, las cuentas de las columnas D y E!:

	A	B	C	D	E
1	1	dsds	dsdsds	dffsdsdff	sfs
2	2			1	ACTIVO
3	3			11	DISPONIBLE
4	4			1105	CAJA
5	5			110505	CAJA GENERAL
6	7			110510	CAJAS MENORES
7	8			11051001	CAJA MENOR BOGOTA
8	10			110515	MONEDA EXTRANJERA

Ahora, seleccione todas las cuentas que aparezcan en el rango D y E, CORTELAS y PASELAS a la B y la C:

	A	B	C	D	E
1	1	dsds	dsdsds	dffsdsdff	sfs
2	2			1	ACTIVO
3	3			11	DISPONIBLE
4	4			1105	CAJA
5	5			110505	CAJA GENERAL
6	7			110510	CAJAS MENORES
7	8			11051001	CAJA MENOR BOGOTA
8	10			110515	MONEDA EXTRANJERA
9	12			1110	BANCOS

	A	B	C	D	E
1	1	dsds	dsdsds	dffsdsdff	sfs
2	2	1	ACTIVO		
3	3	11	DISPONIBLE		
4	4	1105	CAJA		
5	5	110505	CAJA GENERAL		
6	7	110510	CAJAS MENORES		
7	8	11051001	CAJA MENOR BOGOTA		
8	10	110515	MONEDA EXTRANJERA		
9	12	1110	BANCOS		
10	13	111005	MONEDA NACIONAL		
11	14	11100503	BANCO SUDAMERIS COLOMBIA		
12	16	11100505	BANCO DE OCCIDENTE		

Vuelva y ordene por el numero serial. Ese era el fin de hacer ese serial, para retornar al orden original:

A	B	C	D	E	F	G	H	I	J	K
43	43	13 DEUDORES								
44	44	1305 CLIENTES								
45	45	130505 NACIONALES								
46	46	130505 NACIONALES	900200514 FULANITO DETAL		0		24.336.000,00		24.336.000,00	0
47	47	130505 NACIONALES	860006333 MENGANITO DETAL		15.243.750,00		0		15.243.750,00	0
48	48	130505 NACIONALES	900015051 TUTANITO DETAL		0		28.102.500,00		28.197.500,00	4.905.000,00

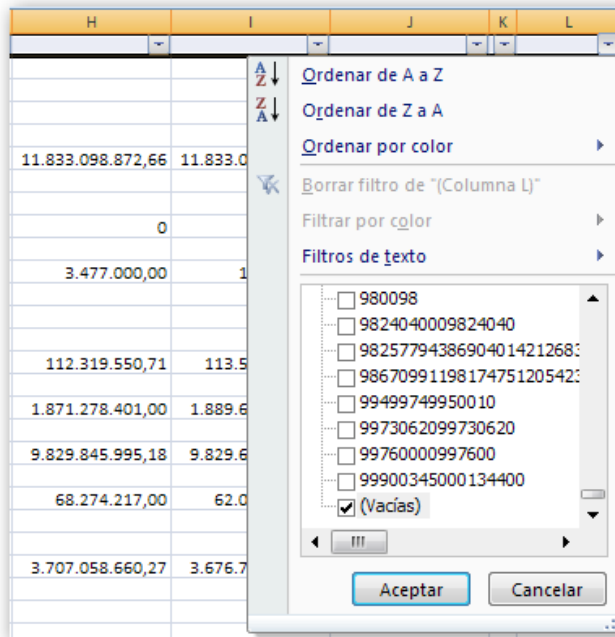
Ahora fíjese como las cuentas ya quedaron en SU columna aparte.

Ahora, a las transacciones que tienen en su misma fila, su NIT, solo les falta tener esta cuenta copiada al lado de estos, Verdad? Y esto lo haremos con la técnica de rellenar celdas en blanco que vimos al principio del libro. Por eso no me voy a detener en imágenes paso a paso para esta parte.

Una vez rellenadas, nos quedan así: Bingo! Ya tenemos la información transaccional con todo lo necesario! En la misma línea la Cuenta, descripción, código de tercero, nombre de tercero, saldo inicial, debito, crédito y saldo final.

A	B	C	D	E	F	G	H	I	J	K	L
43	43	13 DEUDORES									
44	44	1305 CLIENTES									
45	45	130505 NACIONALES									
46	46	130505 NACIONALES	900200514 FULANITO DETAL		0		24.336.000,00		24.336.000,00	0	02433600024
47	47	130505 NACIONALES	860006333 MENGANITO DETAL		15.243.750,00		0		15.243.750,00	0	15243750015
48	48	130505 NACIONALES	900015051 TUTANITO DETAL		0		28.102.500,00		28.197.500,00	4.905.000,00	02810250028

Ok, ahora nos deshacemos de las cuentas que copiamos, las que ahora NO tienen información transaccional (las amarillas B y C) pues como ya las copiamos al lado de los Nits, con la técnica de rellenar, y sabemos en donde están gracias a la columna L, pues simplemente filtramos las vacías y las eliminamos:



A	B	C	D	E	F	G	H	I	J	K	L
1	1	21	ACTIVO								
2	2	311	DISPONIBLE								
3	3	41105	CAJA								
4	4	510505	CAJA GENERAL								
5	5	710510	CAJAS MENORES								

Si ve alguna otra basura en el filtro elimínela también.

Elimine la columna L pues ya no la necesitamos y desaparezca el filtro. El archivo ya le deberá quedar depurado así:

	A	B	C	D	E	F	G	H	I	J	K
16	48	130505	NACIONALES	900300814 PULANITO DETAL		0		24.836.000,00	24.836.000,00		0
17	47	130505	NACIONALES	860006233 MENGANITO DETAL		15.243.750,00		0	15.243.750,00		0
18	48	130505	NACIONALES	900015051 SUTANITO DETAL		0		28.102.500,00	23.197.500,00		4.905.000,00

Fíjese que el archivo ya parece una base de datos. Ahora solo le falta arreglar esos CR que indican que la cuenta es saldo CREDITO. Esta parte también la comenté más arriba, en la primera parte del libro: mediante un Si condicional, colocamos que en donde aparezca un CR, multiplique la celda por -1, sino deje la misma celda. Recuerda? Esa formula la pondría aquí en la columna L2, después de eliminar la que contenía las celdas concatenadas:

L2		fx =SI(G2="CR";-F2;F2)									
	F	G	H	I	J	K	L				
1											
2	0		11.833.098.872,66	11.833.098.872,66		0	0				
3	1.500.000,00		0	0	1.500.000,00						
4	0		3.477.000,00	111.615,00	3.365.385,00						
5	1.601.421,30		112.319.550,71	113.533.574,88	387.397,13						
6	25.141.258,70		1.871.278.401,00	1.889.646.958,01	6.772.701,69						
7	71.026.493,78		9.829.845.995,18	9.829.633.983,29	71.238.505,67						
8	4.264.612,00		68.274.217,00	62.057.955,80	10.480.873,20						
9	16.927.368,80		3.707.058.660,27	3.676.789.533,29	47.196.495,78						
10	3.325.948,11		160.435.806,57	162.486.262,54	1.275.492,14						
11	354.954,57		77.703.741,90	77.426.994,31	631.702,16						
12	81.376.000,00		0	0	81.376.000,00						
13	9.000.000.000,00		0	0	9.000.000.000,00						
14	101.394.302,00		0	0	101.394.302,00						
15	132.398.538,36	CR	5.578.529,00	0	126.820.009,36	CR					

Copia la formula hacia abajo, copia nuevamente todo el rango y le da pegado especial solo valores:

	F	G	H	I	J	K	L
1							
2	0		11.833.098.872,66	11.833.098.872,66		0	0,00
3	1.500.000,00		0	0	1.500.000,00		1.500.000,00
4	0		3.477.000,00	111.615,00	3.365.385,00		0,00
5	1.601.421,30		112.319.550,71	113.533.574,88	387.397,13		1.601.421,30
6	25.141.258,70		1.871.278.401,00	1.889.646.958,01	6.772.701,69		25.141.258,70
7	71.026.493,78		9.829.845.995,18	9.829.633.983,29	71.238.505,67		71.026.493,78
8	4.264.612,00		68.274.217,00	62.057.955,80	10.480.873,20		4.264.612,00
9	16.927.368,80		3.707.058.660,27	3.676.789.533,29	47.196.495,78		16.927.368,80
10	3.325.948,11		160.435.806,57	162.486.262,54	1.275.492,14		3.325.948,11
11	354.954,57		77.703.741,90	77.426.994,31	631.702,16		354.954,57
12	81.376.000,00		0	0	81.376.000,00		81.376.000,00
13	9.000.000.000,00		0	0	9.000.000.000,00		9.000.000.000,00
14	101.394.302,00		0	0	101.394.302,00		101.394.302,00
15	132.398.538,36	CR	5.578.529,00	0	126.820.009,36	CR	(132.398.538,36)

Esta columna ya puede reemplazar la F. Como saldo final se conforma del saldo inicial mas débitos menos créditos, mediante una formulita simple, organiza la columna J:

	F	G	H	I	J	K	L
1							
2	0,00		11.833.098.872,66	11.833.098.872,66		0	0,00
3	1.500.000,00		0	0	1.500.000,00		1.500.000,00
4	0,00		3.477.000,00	111.615,00	3.365.385,00		3.365.385,00
5	1.601.421,30		112.319.550,71	113.533.574,88	387.397,13		387.397,13
6	25.141.258,70		1.871.278.401,00	1.889.646.958,01	6.772.701,69		6.772.701,69
7	71.026.493,78		9.829.845.995,18	9.829.633.983,29	71.238.505,67		71.238.505,67
8	4.264.612,00		68.274.217,00	62.057.955,80	10.480.873,20		10.480.873,20
9	16.927.368,80		3.707.058.660,27	3.676.789.533,29	47.196.495,78		47.196.495,78
10	3.325.948,11		160.435.806,57	162.486.262,54	1.275.492,14		1.275.492,14
11	354.954,57		77.703.741,90	77.426.994,31	631.702,16		631.702,16
12	81.376.000,00		0	0	81.376.000,00		81.376.000,00
13	9.000.000.000,00		0	0	9.000.000.000,00		9.000.000.000,00
14	101.394.302,00		0	0	101.394.302,00		101.394.302,00
15	(132.398.538,36)	CR	5.578.529,00	0	126.820.009,36	CR	(126.820.009,36)
16	0,00		24.336.000,00	24.336.000,00		0	0,00

Elimina las columnas G y K, que contienen esos "CR", y ya la información le queda en forma de base de datos.

Se puede hacer una macro en lugar de hacer todo esto manualmente?

Puede ver que este proceso es muy repetitivo, que aunque le he mostrado como hacerlo paso a paso, y que parece **TEDIOSO** de hacer, con la práctica se hace bastante rápido, en menos de 5 minutos. Pero como es repetitivo, siempre y cuando el archivo sea SIEMPRE el mismo formato (totales por abajo, y mismo nivel, sin poner valores de moneda extranjera) se puede grabar como una macro, y la macro lo organiza en menos de 5 segundos. (Por supuesto que totales por arriba también se puede hacer, solo hay que tener en cuenta las líneas duplicadas y consecutivas y eliminarlas mediante macro).

Ahora, veremos cómo hacer la macro que nos haga todo esto que hicimos sin necesidad de volverlo a hacer manualmente. Solo deberá pedir el listado del CG1, colocarlo en el escritorio y correr la macro. (Incluso puede generar el archivo CG03013 y dejarlo en la carpeta PRT y comenzar a grabar la macro desde el momento que va y lo busca allá con el explorador Abrir archivo de Excel). Casi todo se hace con el grabador de macros, no necesita ser un experto programador para hacer una macro que haga esta tarea. Los conceptos clave ya se los enseñé en el tema de macros y depuración de archivos planos.

La macro también se puede mejorar todo lo que se quiera, y escribir de varias maneras. Yo me voy a servir del grabador de macros de Excel y en lo mínimo evitaré EDITAR el código de la macro, salvo las técnicas útiles que hemos visto en detalle en la sección de macros, como la variable ULTIMAFILA. Recuerda? para que vea que no es difícil. Si es necesario recorrer los registros, usaré Do Until en lugar de For Next, para que cuando la corra paso a paso, pueda ver por dónde se va moviendo la celda activa. Así se dará cuenta que casi todo se puede hacer con el grabador de macros, y es posible defenderse bastante bien y hacer macros por lo menos efectivas. Poco a poco, mejorará sus macros, e intentará escribirlas, como le he enseñado en el tema de MACROS, sin seleccionar las celdas. Después de depurar con macro el archivo plano de balance de comprobación por terceros, puede, también mediante macro, hacer una tabla dinámica y construir un balance general y un estado de resultados. Eliminar la tabla dinámica y organizar los totales de cuenta como le enseñé. Ahora si le gusta, puede hacer usted mismo el reporte que quiera o que le haga falta. La mayoría de los archivos planos del CG1 se arreglan de esta manera: Cartera por edades, listado auxiliar de cuentas, etc. Una vez que tenga sus macros para los mismos archivos planos clave, no tendrá que volver a "matarse" y verá como, después del cierre contable, si el gerente le ordena meter un ajuste de última hora habiendo ya Usted elaborado el Balance y el PyG., como podrá contabilizar el ajuste, generar el archivo plano, correr la macro y

tendrá en segundos nuevamente su Balance y PyG. Así que es verdad cuando le digo que Excel es su mejor asistente, su mano derecha en su día a día.

Creando la macro que depura el balance de comprobación por terceros

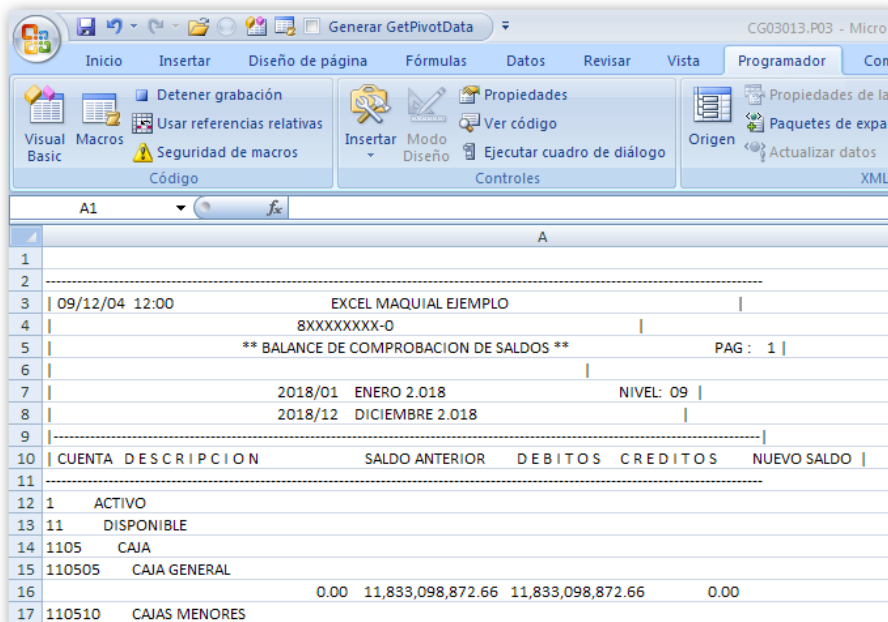
Ok, en esta macro vamos a sintetizar todo lo que hemos aprendido sobre depurar archivos planos. Le aconsejo que si no ha leído el tema anterior vaya y se lo lea, pues mucho de lo que aquí viene es un resumen de lo visto más en detalle.

Paso 1: Abrimos Excel, prendemos el grabador de macros,



Le colocamos el nombre que creamos es el más IDONEO para nuestro archivo, lo colocaremos en el libro de macros personal, o en "este" libro, luego abriremos con Archivo-Abrir de Excel, el archivo plano CG03013, que en mi caso estará ubicado en el escritorio, (o desktop), (ojo, para ver el archivo plano no olvide ver TODOS LOS ARCHIVOS), NO LO SEPARAREMOS, todo quedará empaquetado en una sola columna, los datos serán tipo Texto, y justificaremos la columna A. No es necesario para Excel pero para nosotros sí. Esta justificación luego se podría quitar, pues al ser una línea de código mas, pues es mejor si son líneas de código menos. Justificamos pues necesitamos por el momento, visualizar el contenido del archivo plano en una sola columna.

En cuanto hagamos eso, **DETENEMOS LA MACRO**, quedaremos en este momento y en el Editor de Visual Basic se generará el siguiente código:



Todo lo anterior generó el siguiente código:

```

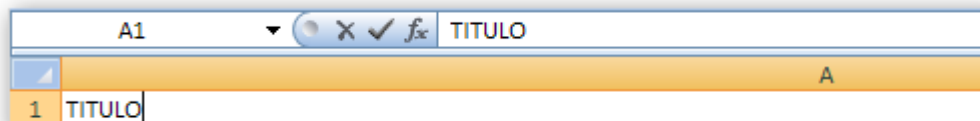
Microsoft Visual Basic - Libro1 - [Módulo1 (Código)]
Archivo  Edición  Ver  Insertar  Formato  Depuración  Ejecutar  Herramientas  Complementos  Ventana  Ayuda
Lín 16, Col 1
(General)  EjemploLibroDepuracionCG03013

Sub EjemploLibroDepuracionCG03013 ()
'
' EjemploLibroDepuracionCG03013 Macro
'
'
'
ChDir "C:\Documents and Settings\Administrador\Escritorio"
Workbooks.OpenText Filename:= _
"C:\Documents and Settings\Administrador\Escritorio\CG03013.P03", Origin:= _
xlMSDOS, StartRow:=1, DataType:=xlDelimited, TextQualifier:=xlDoubleQuote _
, ConsecutiveDelimiter:=False, Tab:=False, Semicolon:=False, Comma:= _
False, Space:=False, Other:=False, FieldInfo:=Array(1, 2), _
TrailingMinusNumbers:=True
Columns("A:A").EntireColumn.AutoFit
End Sub

```

Ok. Volvemos y prendemos el grabador de macros. (Podemos dejar esa macro en "Este libro", es decir en el archivo CG03013.P01) cuando grabemos lo que le diré a continuación y detengamos la macro, podemos cortar el código grabado que se generó en uno de los módulos del proyecto CG03013, pasarnos al modulo de la macro principal es decir, pegarlo debajo de la macro EjemploLibroDepuracionCG03013, o en la macro principal en donde estamos pegando todas las acciones grabadas.

Continuemos: Estando prendido el grabador de macros, en la hoja de cálculo selecciono la celda A1 que en este archivo plano está VACÍA y coloco cualquier palabra, como la palabra Título. Al escribir Título, **NO doy enter**, sino que escojo el CHULITO DE VISTO BUENO QUE DICE INTRODUCIR:



Porque? Si oprimos ENTER, quedará seleccionada la celda A2, Y NO DESEO SELECCIONAR A2 pues no voy a hacer nada en esa celda, y menos aún quiero que me quede grabada esa acción en mi macro. Para que no quede seleccionada A2, pues haces clic en el CHULITO en lugar de dar ENTER.

Esta palabra "Título" la coloco, porque más adelante voy a colocar un filtro. Si no denominara esa primera celda en blanco con un nombre, quedaría incluida cuando filtre los datos, pues uno de los filtros que haremos será deshacernos de las celdas en blanco... Y si eliminamos HASTA la celda que contiene el filtro, pues tendremos que volver a montar el filtro. Para eso, le colocamos un título cualquiera, pues a partir de él es que me voy a deshacer de toda la basura que está en el archivo plano.

Selecciono toda la columna A, es decir, la columna A entera, Completamente seleccionada, y activo los filtros. Le comentaba cuando depuramos manualmente el archivo, que si el archivo es MUY GRANDE, seleccionara el rango de los datos, si ese es su caso, siga esos mismos pasos mientras graba la macro.


```

Sub Macro8()
'
' Macro8 Macro
'
'
'
Windows("CG03013.P03").Activate
Windows("EjemploMacroDepurarCG03013.xlsm").Activate
Windows("CG03013.P03").Activate
ActiveSheet.Range("$A$1:$A$4578").AutoFilter Field:=1, Criteria1:="=*|*", _
    Operator:=xlAnd
Rows("3:3").Select
Range(Selection, Selection.End(xlDown)).Select
Selection.Delete Shift:=xlUp
ActiveSheet.Range("$A$1:$A$4034").AutoFilter Field:=1, Criteria1:=Array("□" _
' _
'-----" _
' _
'-----" _
' _
'□-----" _
' _
'===== " _
), Operator:=xlFilterValues
Rows("2:2").Select
Range(Selection, Selection.End(xlDown)).Select
Selection.Delete Shift:=xlUp
Selection.AutoFilter
End Sub

```

Las instrucciones que seleccioné, que se ven en color azul, se pueden eliminar, pues no hacen falta. Como esta macro, quedó directamente debajo de la otra, pues las unimos, seleccionando y quitando esta parte, así:

```

Columns("A:A").EntireColumn.AutoFit
Range("A1").Select
ActiveCell.FormulaR1C1 = "TITULO"
Columns("A:A").Select
Selection.AutoFilter
End Sub
Sub Macro8()
'
' Macro8 Macro
'
'
Windows("CG03013.P03").Activate
Windows("EjemploMacroDepurarCG03013.xlsm").Activate
Windows("CG03013.P03").Activate
ActiveSheet.Range("$A$1:$A$4578").AutoFilter Field:=1, Criteria1:="=*|*", _
    Operator:=xlAnd

```

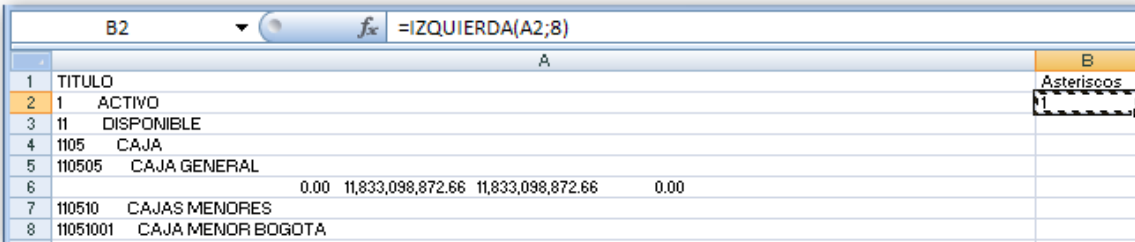
Puede ver que estoy quitando desde donde dice END Sub, que es el final de la macro parcial que grabamos primero y SUB MACRO8(), y de una vez lo que le comentaba hace un momento. Así quedan unidos los dos códigos bajo el nombre de la macro original. Puede probar a ejecutarla y verá como ya hace todo en un INSTANTE. Pero al hacerlo, **no olvide cerrar el archivo plano CG03013 SIN GUARDAR CAMBIOS. Ojo con eso.**

Ya estamos en la parte de eliminar los asteriscos que son subtotales y que son basurita también. Como sabemos que no podemos escoger todo lo que contenga asterisco, pues para Excel un asterisco puede ser cualquier carácter, lo que podemos hacer es, Grabar otra macro mediante la cual en la columna B, colocarle por título "Asteriscos", y en la celda B2, colocar la función =IZQUIERDA(), que nos saque una copia de los primeros caracteres de la parte Izquierda de cada fila de la columna A. Si no comprende bien que hace la función =Izquierda(), vaya a la sección de este libro en donde se explica bien esa función.

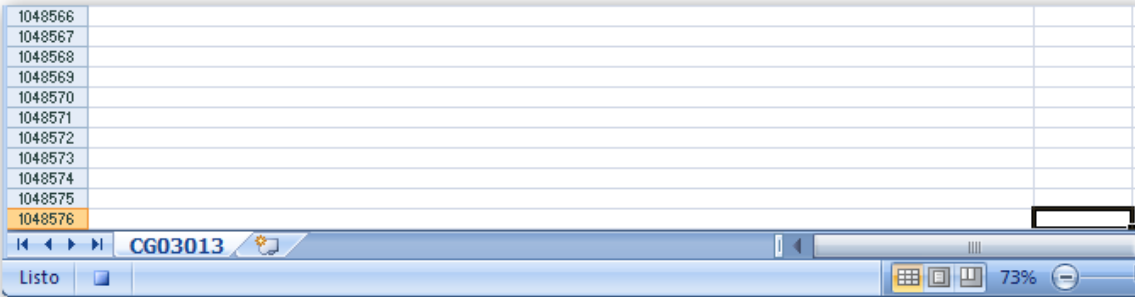
Entonces, haga lo siguiente:

Ubicado en el archivo CG03013, Prenda el grabador de macros. Seleccione la celda B1. Escriba "Asteriscos". Páse a la celda B2. Escriba la formula =Izquierda(A2;8) e introducir formula (OJO, NO ENTER). Copia la formula. Con Fin + flecha Abajo, se va hasta el final de la hoja de cálculo. Se pasa a la columna A. Otra vez, oprime Fin + flecha arriba. Se pasa a la celda de al lado, la B. Luego para seleccionar todo el rango en el cual debe quedar ingresada la formula izquierda, desde allí oprime Shift + Fin + Flecha Arriba, y da enter. La formula debe quedar copiada enteramente. Seguidamente, como tiene todo el rango seleccionado, desbaratamos la formula con Copiar, y Allí mismo, pega la copia pero con Pegado especial Solo valores.

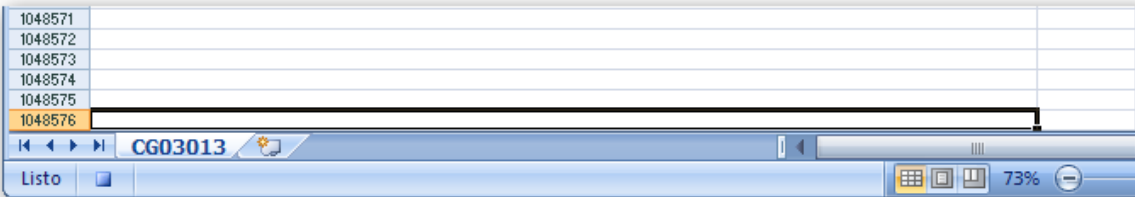
Procedo a mostrarle IMAGEN POR IMAGEN LO QUE HICE:



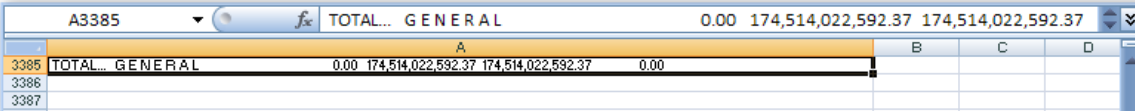
Estoy en la celda B2, escribo la formula, y hago Control C para copiarla. Luego, oprimo Fin + flecha abajo: llego a la última fila de Excel:



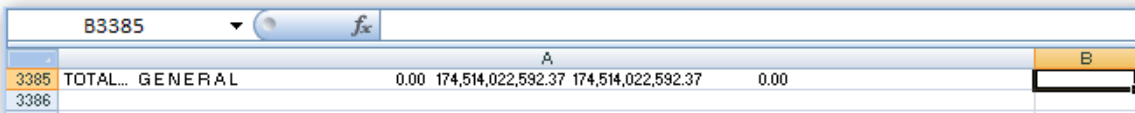
Allá abajo, me paso a la columna A:



Otra vez, oprimo Fin + Flecha hacia Arriba: Me estrello con la parte final del archivo plano:



Me paso a la columna B otra vez:



Oprimo Shift + Fin + flecha arriba para seleccionar todo el rango en donde quiero copiar la formula y oprimo ENTER para pegar la formula a lo largo de todo ese rango:

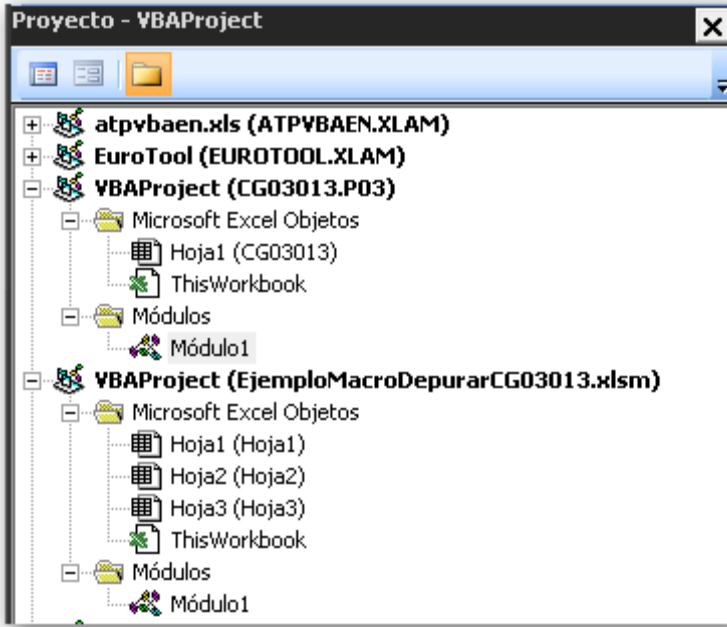
	A					B
3370	739594	TRASLADO A PROCESO				739594
3371			0.00	2,947,956,547.67	4,804,917,577.71	1,856,961,030.04CR
3372	739595	OTROS				739595
3373			0.00	36,829,679.00	36,721,574.00	108,105.00
3374	***	DIVERSOS	0.00	3,523,052,025.14	4,852,752,240.98	1,329,700,215.84CR
3375	***	DIVERSOS	0.00	3,523,052,025.14	4,852,752,240.98	1,329,700,215.84CR
3376	**	COSTOS INDIRECTOS	0.00	4,869,377,996.88	4,869,377,996.88	0.00
3377	**	COSTOS INDIRECTOS	0.00	4,869,377,996.88	4,869,377,996.88	0.00
3378						
3379						
3380	*	COSTOS DE OPERACION Y PRO.	0.00	6,722,012,631.11	6,722,012,631.11	0.00
3381	*	COSTOS DE OPERACION Y PRO.	0.00	6,722,012,631.11	6,722,012,631.11	0.00
3382						
3383						
3384	TOTAL...	GENERAL	0.00	174,514,022,592.37	174,514,022,592.37	0.00
3385	TOTAL...	GENERAL	0.00	174,514,022,592.37	174,514,022,592.37	0.00
3386						

Puede ver que quedó AL LADO, a todo lo largo de nuestro archivo plano, y que está extrayendo los asteriscos. Las dos últimas filas, ya veremos cómo quitarlas, pues son basura pero en todos los archivos son diferentes totales, pero como siempre están de últimas, se pueden eliminar fácilmente, mas adelante lo haremos.

Sigamos. Sobre ese mismo RANGO, en el cual introdujimos la función izquierda, hacemos un copiar – pegado especial – solo valores y desbaratamos la formula Izquierda. Todo este “Paso a paso” generó el siguiente código, que debemos depurar después, pues se puede ver que la línea que le señalo aquí, es una selección FIJA, y lo que habrá que decirle a Excel es que se desplace con ActiveCell.Offset una celda a la derecha, y corregido el problema. Sobre esta instrucción en la sección de macros me detuve MUCHO, para que le quedara clara su utilidad. (Mas adelante depuraremos esas cositas):

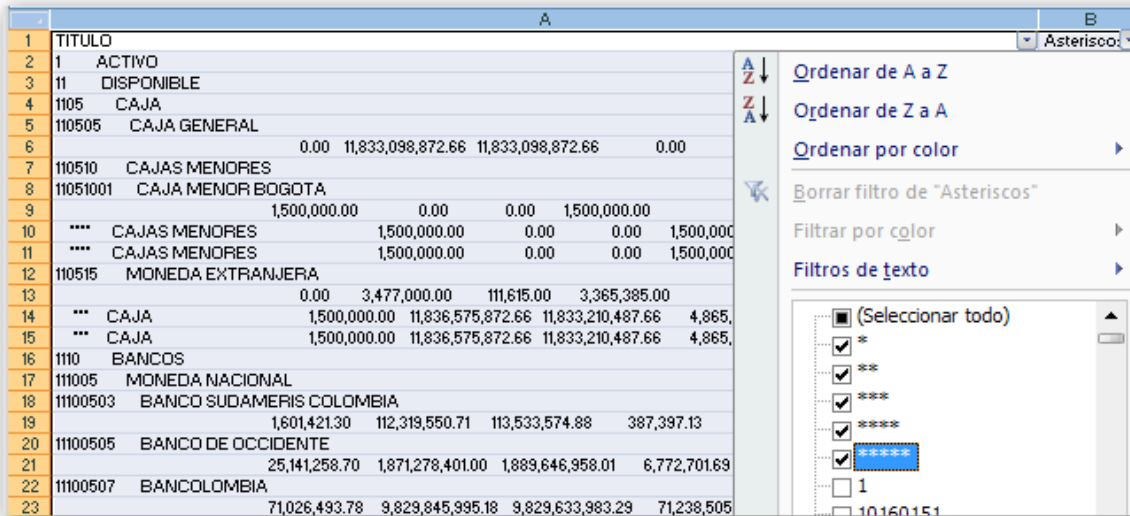
```
Sub Macro15()
'
' Macro15 Macro
'
'
'
Range("B1").Select
ActiveCell.FormulaR1C1 = "Asteriscos"
Range("B2").Select
ActiveCell.FormulaR1C1 = "=LEFT(RC[-1],8)"
Selection.Copy
Selection.End(xlDown).Select
Range("A1048576").Select
Selection.End(xlUp).Select
Range("B3385").Select
Range(Selection, Selection.End(xlUp)).Select
ActiveSheet.Paste
Application.CutCopyMode = False
Selection.Copy
Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
:=False, Transpose:=False
Application.CutCopyMode = False
End Sub
```

Nota: Si el fragmento de macro, le queda grabado dentro del modulo del otro archivo, el CG03013:



No se preocupe, ábralo, corte la macro, y péguela debajo de las demás macros del modulo1 del archivo en el cual está guardando las macros parciales que estamos uniendo, que en este caso, es el archivo "EjemploMacroDepurarCG03013.xlsm".

Sigamos. Volvemos a grabar otra macro. En ella, ahora seleccionamos las columnas A y B, colocamos un filtro y con un filtro múltiple en la columna ASTERISCOS seleccionamos todos los asteriscos, el de un asterisco, el de dos asteriscos, todos, y los eliminamos. (Hay archivos en donde aparecen más asteriscos, este trae hasta 5 ***** , a veces mas, para tener en cuenta si quiere montar una macro que le corra en todos los casos)



	A					B
1	TITULO					Asterisco
10	**** CAJAS MENORES	1,500,000.00	0.00	0.00	1,500,000.00	****
11	**** CAJAS MENORES	1,500,000.00	0.00	0.00	1,500,000.00	****
14	*** CAJA	1,500,000.00	11,836,575,872.66	11,833,210,487.66	4,865,385.00	***
15	*** CAJA	1,500,000.00	11,836,575,872.66	11,833,210,487.66	4,865,385.00	***
26	**** MONEDA NACIONAL	102,033,785.78	11,881,718,163.89	11,894,872,471.98	88,879,477.69	****
27	**** MONEDA NACIONAL	102,033,785.78	11,881,718,163.89	11,894,872,471.98	88,879,477.69	****
31	**** MONEDA EXTRANJERA	16,927,368.80	3,707,058,660.27	3,676,789,533.29	47,196,495.78	****
32	**** MONEDA EXTRANJERA	16,927,368.80	3,707,058,660.27	3,676,789,533.29	47,196,495.78	****
33	*** BANCOS	118,961,154.58	15,588,776,824.16	15,571,662,005.27	136,075,973.47	***
34	*** BANCOS	118,961,154.58	15,588,776,824.16	15,571,662,005.27	136,075,973.47	***
39	**** BANCOS	3,325,948.11	160,435,806.57	162,486,262.54	1,275,492.14	****
40	**** BANCOS	3,325,948.11	160,435,806.57	162,486,262.54	1,275,492.14	****
44	**** CORPORACIONES DE AHORRO Y VIV	354,954.57	77,703,741.90	77,426,934.31	631,702.16	****

Seleccionamos desde la segunda, hasta el final y eliminamos todas las filas, seguidamente desactivamos el filtro, eliminamos la columna B, y seleccionamos la celda A1.

Una vez realizado lo anterior, el grabador de macros nos genera este código:

```
Sub Macro17()
'
' Macro17 Macro
'
Columns("A:B").Select
Selection.AutoFilter
ActiveSheet.Range("$A$1:$B$4578").AutoFilter Field:=2, Criteria:=Array("*" _
, "****", "*****", "*****", "*****"), Operator:=xlFilterValues
Rows("10:10").Select
Range(Selection, Selection.End(xlDown)).Select
Selection.Delete Shift:=xlUp
Selection.AutoFilter
Columns("B:B").Select
Selection.Delete Shift:=xlToLeft
Range("A1").Select
End Sub
```

El único problema verdaderamente serio del código anterior, es la selección del rango, el cual es fijo. Más adelante arreglaremos esto, pues no todos los archivos tendrán la misma cantidad de registros.

El archivo ya nos va quedando así:

	A				
1	TITULO				
2	1 ACTIVO				
3	11 DISPONIBLE				
4	1105 CAJA				
5	110505 CAJA GENERAL				
6		0.00	11,833,098,872.66	11,833,098,872.66	0.00
7	110510 CAJAS MENORES				
8	11051001 CAJA MENOR BOGOTA				
9		1,500,000.00	0.00	0.00	1,500,000.00
10	110515 MONEDA EXTRANJERA				
11		0.00	3,477,000.00	111,615.00	3,365,385.00
12	1110 BANCOS				
13	111005 MONEDA NACIONAL				

Ok, asteriscos Fuera. Ya podemos ver que la información ya comienza a parecer una base de datos,

Ya estamos cerca del final. Ahora, Con otra macro, tal como lo hicimos manualmente, con mucho cuidado, con la herramienta texto en columnas, Seleccionamos la columna a, activamos la herramienta texto en columnas y vamos a partir el archivo plano. El archivo visualmente nos queda así:

	A	B	C	D	E	F	G	H
1	TITULO							
2	1	ACTIVO						
3	11	DISPONIBLE						
4	1105	CAJA						
5	110505	CAJA GENERAL						
6			0		11.833.098.872,66	11.833.098.872,66		0
7	110510	CAJAS MENORES						
8	11051001	CAJA MENOR BOGOTA						
9			1.500.000,00		0	0		1.500.000,00
10	110515	MONEDA EXTRANJERA						
11			0		3.477.000,00	111.615,00		3.365.385,00

Y el grabador de macros generó el siguiente código:

```
Sub Macro20()
'
' Macro20 Macro
'
'
Columns("A:A").Select
Selection.TextToColumns Destination:=Range("A1"), DataType:=xlFixedWidth, _
FieldInfo:=Array(Array(0, 1), Array(11, 1), Array(45, 1), Array(73, 1), Array(75, 1), _
Array(94, 1), Array(114, 1), Array(134, 1)), DecimalSeparator:=".", _
ThousandsSeparator:=",", TrailingMinusNumbers:=True
Columns("A:I").Select
Columns("A:I").EntireColumn.AutoFit
Range("A1").Select
End Sub
```

En la primera imagen puede ver que el título, que nos ayudaba en el filtro, quedó en la columna A, y que las demás columnas les hace falta un rótulo de columna o encabezado o título. Ahora, en la siguiente MACRO, a las demás columnas les colocamos un encabezado de título temporal, puede ser cualquier cosa, para que se REACTIVE la región corriente y Excel SEPA que corresponde a una TABLA. Además, en la columna I, colocaremos una fórmula que concatene, uniremos las celdas C, E, F y G con un & Ampersand que nos ayude a identificar las cuentas, para poderlas separar a un lado. Tal como hicimos en la parte manual. Seguidamente, colocaremos un número serial, que nos permita regresar al orden original, pues vamos a ordenar todo esto por la nueva columna I, que es la columna en la cual detectaremos las cuentas. Una vez realizadas estas dos acciones, el archivo queda así:

	A	B	C	D	E	F	G	H	I
1	TITULO	sd	sdf	sdfc	sdfsdfdsffsd	sdfsdfdsdfdsf	sdfsdfdsdfdsf	dsfc	reterdvfvz
2	1	ACTIVO							
3	11	DISPONIBLE							
4	1105	CAJA							
5	110505	CAJA GENERAL							
6			0		11.833.098.872,66	11.833.098.872,66			011833098872
7	110510	CAJAS MENORES							
8	11051001	CAJA MENOR BOGOTA							
9			1.500.000,00		0	0	1.500.000,00		1500000001500
10	110515	MONEDA EXTRANJERA							
11			0		3.477.000,00	111.615,00	3.365.385,00		0347700011161
12	1110	BANCOS							
13	111005	MONEDA NACIONAL							
14	11100503	BANCO SUDAMERIS COLOMBIA							
15			1.601.421,30		112.319.550,71	113.533.574,88	387.397,13		1601421,3112319

En la imagen seleccioné el rango I2 – I5, para que vea que el concatenar, A LA FIJA, nos DICE cuales son las filas que SIN TEMOR A EQUIVOCARNOS, SON CUENTAS y NO TERCEROS, valiéndonos de la consistencia que se observa, la cual es, que cuando en la fila hay una cuenta, No hay valores transaccionales.

Coloque unos títulos cualquiera, y en la celda I2, coloqué la función que concatena las celdas C, E, F y G, tal como lo hicimos en el ejercicio anterior en el cual depuramos el archivo manualmente y mostramos como era la función. Seguidamente se copia la función de I2 a todo lo largo del archivo plano, me voy hasta el final, estando allá abajo me paso a la celda de la columna G, subo hasta estrellarme con la parte final del archivo plano, y me paso nuevamente a la columna I. Estando en esa posición, selecciono TODO el rango que recorre a todo lo largo del archivo plano con toda seguridad, y pego la columna que quiero copiar. Una vez copiada, SOBRE EL MISMO RANGO, hago un nuevo Copiar, y ALLI MISMO, EN ESE MISMO RANGO, SIN MOVERME, HAGO UN PEGAR, PEGADO ESPECIAL SOLO VALORES. El grabador de macros con estas acciones, que nos dejaron el archivo plano como se ve en la imagen, generó el siguiente código:

```
Sub Macro6()
'
' Macro6 Macro
'
'
'
Range("B1").Select
ActiveCell.FormulaR1C1 = "sd"
Range("C1").Select
ActiveCell.FormulaR1C1 = "sdf"
Range("D1").Select
ActiveCell.FormulaR1C1 = "sdfds"
Range("E1").Select
ActiveCell.FormulaR1C1 = "sdfsdffsd"
Range("F1").Select
ActiveCell.FormulaR1C1 = "sdfsdffsd"
Range("G1").Select
ActiveCell.FormulaR1C1 = "sdfsdffsd"
Range("H1").Select
ActiveCell.FormulaR1C1 = "dsfds"
Range("I1").Select
ActiveCell.FormulaR1C1 = "reterdvfvx"
Range("I2").Select
ActiveCell.FormulaR1C1 = "=RC[-6] &RC[-4] &RC[-3] &RC[-2]"
Selection.Copy
Selection.End(xlDown).Select
Range("G1048576").Select
Selection.End(xlUp).Select
Range("I2753").Select
Range(Selection, Selection.End(xlUp)).Select
ActiveSheet.Paste
Application.CutCopyMode = False
Selection.Copy
Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
:=False, Transpose:=False
Application.CutCopyMode = False
Selection.End(xlUp).Select
End Sub
```

Se ve un poco largo, pero lo que lo hace largo son esas líneas con las cuales se "bautizan" los encabezados de columna. Intencionalmente he marcado la línea Range("I2753").Select, que es el único problema serio, el resto del código se puede depurar, y mucho, pero no hacerlo si bien no es indispensable, si es importante. Más adelante, cuando hagamos una revisión del código, arreglamos eso.

Ahora sigue colocar en una columna A un número serial. Porque? Porque vamos a ORDENAR por esta columna I, en la cual hicimos el concatenar, para DEJAR JUNTICAS TODAS LAS CUENTAS Y PODERLAS PASAR A UNAS COLUMNAS APARTE, PARA SEPARARLAS DE LOS NITS. Recuerde que las cuentas están

en la misma columna de los Nits, y esto, para una base de datos, NO PUEDE SER. La columna o Es campo Código de tercero o Campo Cuenta, pero no pueden ser los dos. Lo que hicimos para solucionarlo, fue que las tamizamos, las seleccionamos, las identificamos mediante el principio de que cuando hay una cuenta no hay valores transaccionales, y lo hicimos con el concatenar en la columna I.

Ok, espero que me vaya siguiendo. Sigamos. En otra macro, voy a montar el serial en la columna A. Es la lección que le enseñé al depurar la información Ordenando los datos. ESO MISMO:

Inserto UNA COLUMNA NUEVA EN LA COLUMNA A. Coloco en A1 el título "Serial", en A2 el numero 1, y en A3, el numero 3. Selecciono las dos primeras casillas y arrastro hasta el final del archivo plano.

Al realizar esta operación se genera el siguiente código:

```

Sub Macro6()
    Macro6 Macro
    Columns("A:A").Select
    Selection.Insert Shift:=xlToRight, CopyOrigin:=xlFormatFromLeftOrAbove
    Range("A1").Select
    ActiveCell.FormulaR1C1 = "Serial"
    Range("A2").Select
    ActiveCell.FormulaR1C1 = "1"
    Range("A3").Select
    ActiveCell.FormulaR1C1 = "2"
    Range("A2:A3").Select
    Selection.AutoFill Destination:=Range("&A2:&A2753"), Type:=xlFillDefault
    Range("A2:A2753").Select
    Range("A1").Select
End Sub
    
```

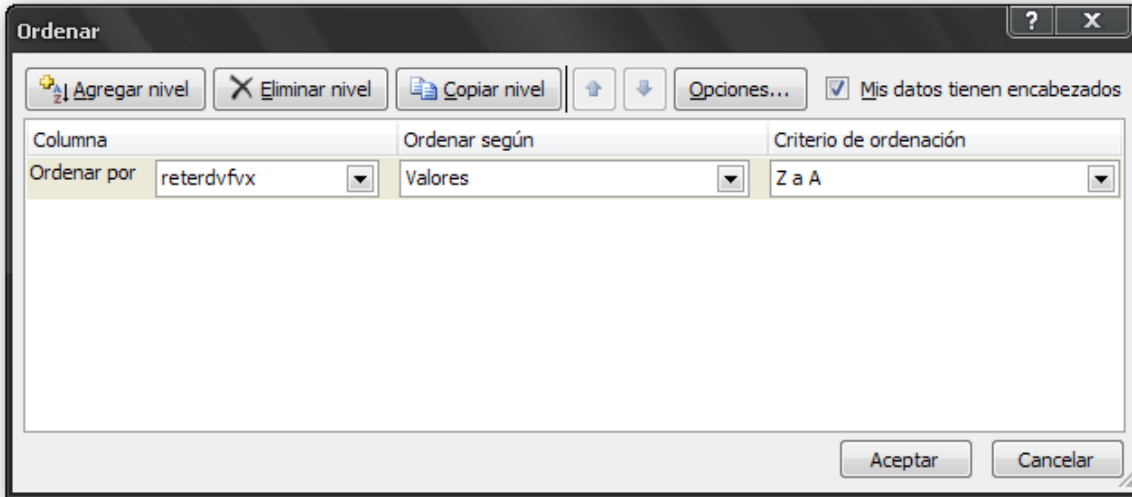
Recuerde que si la macro le queda grabada en el archivo CG03013, debe recortarla y pegarla debajo del resto de macros que luego uniremos, y esas las estamos colocando todas, una debajo de la otra en el modulo1 del archivo "EjemploMacroDepurarCG03013.xlsm"

Esto fue lo que hicimos, y grabamos:

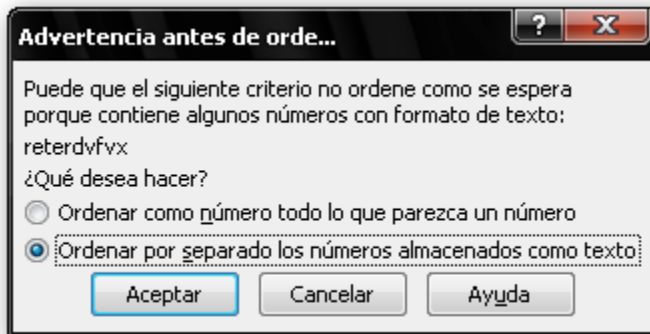
	A	B	C	D	E	F
1	Serial	TITULO	sd	sdf	sdfc	sdfsdfdsffsd
2	1	1	ACTIVO			
3	2	11	DISPONIBLE			
4	3	1105	CAJA			
5	4	110505	CAJA GENERAL			
6	5				0	11.833.098.872,66
7	6	110510	CAJAS MENORES			
8	7	11051001	CAJA MENOR BOGOTA			
9	8			1.500.000,00		0
10	9	110515	MONEDA EXTRANJERA			
11	10				0	3.477.000,00
12	11	1110	BANCOS			

Ahora, en la siguiente macro ordenamos toda la información por la columna J, que es la columna en donde hicimos el "join", es decir, en donde unimos las celdas de los valores para detectar las filas que

realmente están en blanco, y que indudablemente corresponden a las cuentas. Grabamos la macros y ordenamos por esa fila:



Ante la pregunta:



Le damos "Ordenar por separado los números almacenados como texto". De esta manera, las cuentas no quedarán en donde queremos, arriba:

	A	B	C	D	E	F	G	H	I	J
1	Serial	TITULO	sd	sdf	sdfc	sdfsdfsfsd	sdfsdfsdfsdf	sdfsdfsdfsfsf	dsfc	reterdvfx
2		1	ACTIVO							
3		2	DISPONIBLE							
4		3	CAJA							
5		4	CAJA GENERAL							
6		6	CAJAS MENORES							
7		7	CAJA MENOR BOGOTA							
8		9	MONEDA EXTRANJERA							
9		11	BANCOS							
10		12	MONEDA NACIONAL							
11		13	BANCO SUDAMERIS COLOMBIA							
12		15	BANCO DE OCCIDENTE							
13		17	BANCOLOMBIA							
14		19	BANCO DE BOGOTA CTA CTE 416-							
15		21	MONEDA EXTRANJERA							
16		22	BANCOLOMBIA MIAMI							
17		24	CUENTAS DE AHORROS							
18		25	BANCOS							
19		26	BANCO DAVIVIENDA CTA 0099 70							
20		28	CORPORACIONES DE AHORRO Y VIV							

Vé?

Ya tenemos las cuentas TODAS JUNTAS, REUNIDAS en donde queremos tenerlas. Insertamos dos nuevas columnas en B y en C, en estas columnas colocaremos las cuentas:

	A	B	C	D	E	F
2	1			1	ACTIVO	
3	2			11	DISPONIBLE	
4	3			1105	CAJA	
5	4			110505	CAJA GENERAL	
6	6			110510	CAJAS MENORES	
7	7			11051001	CAJA MENOR BOGOTA	
8	9			110515	MONEDA EXTRANJERA	
9	11			1110	BANCOS	
10	12			111005	MONEDA NACIONAL	
11	13			11100503	BANCO SUDAMERIS COLOMBIA	
12	15			11100505	BANCO DE OCCIDENTE	
13	17			11100507	BANCOLOMBIA	
14	19			11100511	BANCO DE BOGOTA CTA CTE 416-	
15	21			111010	MONEDA EXTRANJERA	
16	22			11101008	BANCOLOMBIA MIAMI	
17	24			1120	CUENTAS DE AHORROS	

Como nos quedaron juntas, solo nos resta BAJAR por la columna E, hasta que llegemos al final de todas las cuentas. Cómo? Seleccionando D1 o D2 y con fin mas flechita hacia abajo, llegas a la ultima cuenta. Allí, junto a esa última celda, la ultima cuenta, seleccionas también la descripción de esa última cuenta, y luego desde abajo seleccionas todo el rango equivalente a las cuentas, con Control + Shift + Flecha arriba:

En la imagen, descendí hasta la última cuenta:

655	2733		739537	HERRAMIENTAS	
656	2735		739540	ENVASES Y EMPAQUES	
657	2737		739545	TAXIS Y BUSES	
658	2739		739560	CASINO Y RESTAURANTE	
659	2741		739565	PARQUEADEROS	
660	2743		739594	TRASLADO A PROCESO	
661	2745		739595	OTROS	

CG03013

Listo

Luego, oprimiré, desde aquí, control + shift + flecha arriba, para tomar todo el rango correspondiente a las cuentas:

	A	B	C	D	E	F
1	Serial			TITULO	sd	sdf
2		1		11	ACTIVO	
3		2		1105	DISPONIBLE	
4		3		110505	CAJA	
5		4		110510	CAJA GENERAL	
6		6		11051001	CAJAS MENORES	
7		7		110515	CAJA MENOR BOGOTA	
8		9		1110	MONEDA EXTRANJERA	
9		11		111005	BANCOS	
10		12		11100503	MONEDA NACIONAL	

Corto la sección D y E y selecciono la Celda B1

	A	B	C	D	E	F
1	Serial			TITULO	sd	sdf
2		1		11	ACTIVO	
3		2		1105	DISPONIBLE	
4		3		110505	CAJA	
5		4		110510	CAJA GENERAL	
6		6		11051001	CAJAS MENORES	
7		7		110515	CAJA MENOR BOGOTA	
8		9		1110	MONEDA EXTRANJERA	
9		11		111005	BANCOS	
10		12		11100503	MONEDA NACIONAL	
11		13		11100503	BANCO SUDAMERIS COLOMBIA	

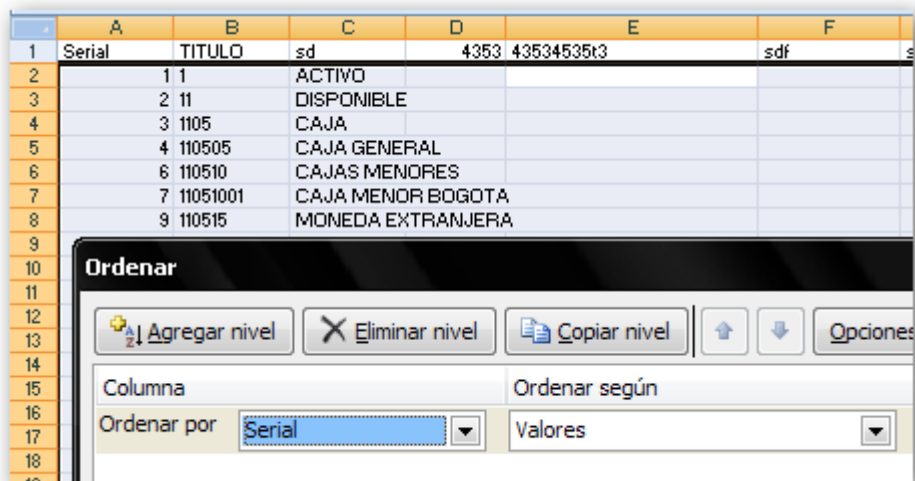
la paso a B y C:

	A	B	C	D	E
1	Serial	TITULO	sd		
2		1	ACTIVO		
3		2	DISPONIBLE		
4		3	CAJA		
5		4	CAJA GENERAL		
6		6	CAJAS MENORES		
7		7	CAJA MENOR BOGOTA		
8		9	MONEDA EXTRANJERA		
9		11	BANCOS		
10		12	MONEDA NACIONAL		
11		13	BANCO SUDAMERIS COLOMBIA		

Ahora, necesitamos de nuevo colocar unos encabezados en D y E, pues nos quedaron VACIOS, así Excel nuevamente reconocerá la región activa de todo el archivo plano. Es necesario porque vamos a ordenar nuevamente por la columna Serial.

	A	B	C	D	E	F
1	Serial	TITULO	sd	4353	43534535t3	sdf
2		1	ACTIVO			
3		2	DISPONIBLE			
4		3	CAJA			
5		4	CAJA GENERAL			
6		6	CAJAS MENORES			

He puesto cualquier cosa, pues todavía no son los encabezados definitivos. Ahora ordeno por la columna A, la del serial:



Al ordenar el archivo plano nos queda así:

31	30					354.954,57
32	31	12	INVERSIONES			
33	32	1205	ACCIONES			
34	33	120520	INDUSTRIA MANUFACTURERA			
35	34			16791503	MELGAREJO INC.	81.376.000,00
36	35	120555	ACTIV. INMOBIL., EMPRESARIALE			
37	36			18791503	SUTANEJO CORP.	9.000.000.000,00
38	37	120599	AJUSTES POR INFLACION			
39	38					101.394.302,00

Fíjese que logramos separar las cuentas de los nits y los nombres de nits!!! Sabe que es lo único que falta? Grabar en la siguiente macro la técnica de rellenar los espacios que le he enseñado, y luego eliminar las celdas de las cuentas, pues ya sabemos en donde están!!

En este punto, al detener la macro, encontramos que se ha grabado el siguiente código:

```
Sub Macro8()
    ActiveWorkbook.Worksheets("CG03013").Sort.SortFields.Clear
    ActiveWorkbook.Worksheets("CG03013").Sort.SortFields.Add Key:=Range( _
        "J2:J2753"), SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:= _
        xlSortNormal
    'xlSortTextAsNumbers
    With ActiveWorkbook.Worksheets("CG03013").Sort
        .SetRange Range("A1:J2753")
        .Header = xlYes
        .MatchCase = False
        .Orientation = xlTopToBottom
        .SortMethod = xlPinYin
        .Apply
    End With
    Columns("B:C").Select
    Selection.Insert Shift:=xlToRight, CopyOrigin:=xlFormatFromLeftOrAbove
End Sub
```

El código grabado continua en la siguiente imagen:

```

Range("D2").Select
Selection.End(xlDown).Select
Range("D661:E661").Select
Range(Selection, Selection.End(xlUp)).Select
Selection.Cut
Range("B1").Select
ActiveSheet.Paste
Range("D1").Select
ActiveCell.FormulaR1C1 = "4353"
Range("E1").Select
ActiveCell.FormulaR1C1 = "43534535t3"
ActiveWorkbook.Worksheets("CG03013").Sort.SortFields.Clear
ActiveWorkbook.Worksheets("CG03013").Sort.SortFields.Add Key:=Range( _
    "A2:A2753"), SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:= _
    xlSortNormal
With ActiveWorkbook.Worksheets("CG03013").Sort
    .SetRange Range("A1:L2753")
    .Header = xlYes
    .MatchCase = False
    .Orientation = xlTopToBottom
    .SortMethod = xlPinYin
    .Apply
End With
End Sub

```

Seguidamente, prendemos nuevamente el grabador de macros, nos ubicamos en la celda A1, bajamos hasta la última, nos pasamos a las celdas de al lado, y desde allí seleccionamos el rango, hasta A1: Estando en A1, bajo hasta la última:

2717	2744			
2718	2745	739595	OTROS	
2719	2746			
2720	2751			TOTAL...
2721	2752			TOTAL...

CG03013

Listo

Con las flechitas, me paso a la celda C:

2717	2744			
2718	2745	739595	OTROS	
2719	2746			
2720	2751			TOTAL...
2721	2752			TOTAL...

CG03013

Listo

Desde aquí, oprimo simultáneamente, con control + shift + inicio AGARRO todo el rango hasta A1:

	A	B	C	D
1	Serial	TITULO	sd	4353
2	1	1	ACTIVO	
3	2	11	DISPONIBLE	
4	3	1105	CAJA	
5	4	110505	CAJA GENERAL	
6	5			
7	6	110510	CAJAS MENORES	
8	7	11051001	CAJA MENOR BOGOTA	
9	8			
10	9	110515	MONEDA EXTRANJERA	
11	10			

Listo, ahora voy por la ficha inicio, buscar y reemplazar, ir a especial, celdas en blanco. Escribimos la formula + la celda de arriba y en lugar de Enter, hago Control+Enter, para copiar en los rellenos la información que está en la celda de arriba de las celdas en blanco:

	A	B	C	D
5	4	110505	CAJA GENERAL	
6	5	110505	CAJA GENERAL	
7	6	110510	CAJAS MENORES	
8	7	11051001	CAJA MENOR BOGOTA	
9	8	11051001	CAJA MENOR BOGOTA	
10	9	110515	MONEDA EXTRANJERA	
11	10	110515	MONEDA EXTRANJERA	
12	11	1110	BANCOS	
13	12	111005	MONEDA NACIONAL	
14	13	11100503	BANCO SUDAMERIS COLOMBIA	
15	14	11100503	BANCO SUDAMERIS COLOMBIA	
16	15	11100505	BANCO DE OCCIDENTE	

Seguidamente, marco las columnas B y C, y hago un copiar, pegado especial, solo valores.

	A	B	C	D
5	4	110505	CAJA GENERAL	
6	5	110505	CAJA GENERAL	
7	6	110510	CAJAS MENORES	
8	7	11051001	CAJA MENOR BOGOTA	
9	8	11051001	CAJA MENOR BOGOTA	
10	9	110515		
11	10	110515		
12	11	1110		
13	12	111005		
14	13	11100503		
15	14	11100503		
16	15	11100503		
17	16	11100503		
18	17	11100503		
19	18	11100503		
20	19	11100511		

Pegado especial

Pegar

Todo
 Todo utiliz

Fórmulas
 Todo exce

Valores
 Ancho de

Formatos
 Formatos

Estos pasos generan el siguiente código:

```

Sub Macro9()
'
' Macro16 Macro
'
'
Range("A1").Select
Selection.End(xlDown).Select
Range("C2721").Select
Windows("EjemploMacroDepurarCG03013.xlsm").Activate
Windows("CG03013.xlsx").Activate
Range(Selection, Cells(1)).Select
Windows("EjemploMacroDepurarCG03013.xlsm").Activate
Windows("CG03013.xlsx").Activate
Selection.SpecialCells(xlCellTypeBlanks).Select
Selection.FormulaR1C1 = "+R[-1]C"
Columns("B:C").Select
Range("B5").Activate
Selection.Copy
Application.CutCopyMode = False
ActiveWindow.SmallScroll Down:=-4
End Sub

```

Aquí estoy señalando el `Range("C2721").Select`, porque ese pedacito habrá que arreglarlo después, y decirle que, habiendo bajado a la última celda con el comando `Selection.End(xlDown).Select`, SE DESPLAZA DOS CELDAS a la derecha DESDE LA CELDA ACTIVA, que es la última que quedó seleccionada.

Estas dos instrucciones que siguen, también se pueden quitar:

```

Windows("EjemploMacroDepurarCG03013.xlsm").Activate
Windows("CG03013.xlsx").Activate

```

Sobran, pues la máquina sabe que está trabajando en el libro activo

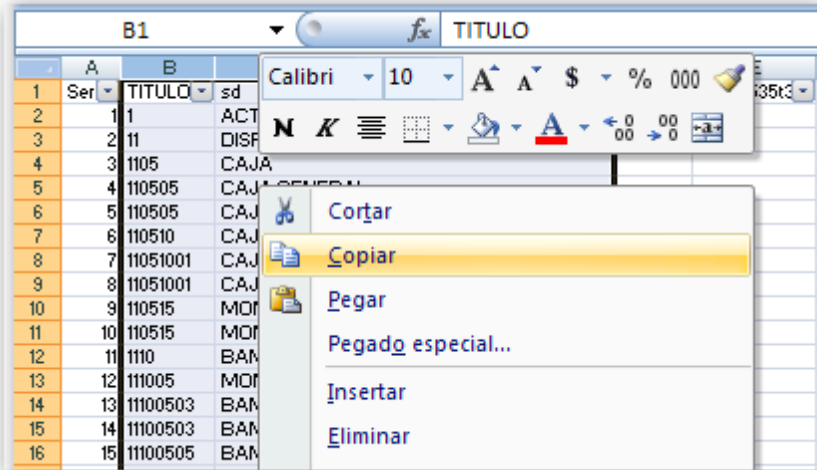
Al final se le "PELUQUEAN" esos comandos un tanto redundantes. Es que cualquier acción incluso las que no agregan valor, quedan grabadas.

Entonces, ya hicimos el relleno colocando una fórmula que no es más que decirle a las celdas vacías que JALEN el dato que está arriba de ellas. +Celda arriba. Antes de continuar, esa fórmula hay que desbaratarla con el viejo y conocido método de copiar las dos columnas y pegarlas ALLI MISMO, pero con la opción, pegar valores. Como al final de esta secuencia, las hemos dejado seleccionadas, solo nos resta hacer la macro de copiar, y luego dar pegado especial valores:

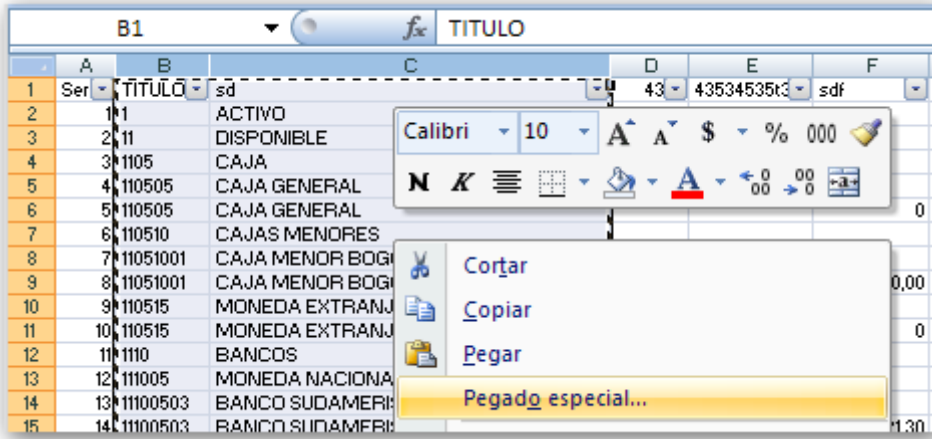
	A	B	C	D	E
1	Ser	TITULO	sd	43	43534535t3
2	1	ACTIVO			
3	2	DISPONIBLE			
4	3	1105	CAJA		

Recuerde que TIENE que seleccionar las dos columnas en las cuales quedó el relleno.

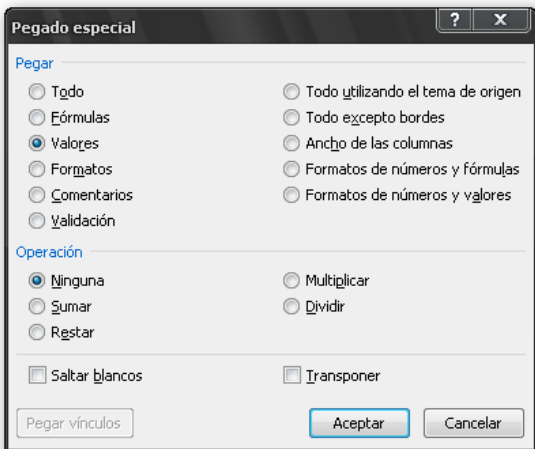
Mire que están las dos columnas seleccionadas, y hago clic derecho para que salga el menú emergente, y hago clic en Copiar.



Después ALLI MISMO hago clic en Pegado especial:



Al hacer clic allí, me saldrá una ventana que me dice QUE COSA EN ESPECIAL QUIERE QUE PEGUE: Activo la casilla SOLO VALORES, y listo, detengo la macro.



Después de dar ACEPTAR, con el teclado del computador oprimo la tecla Escape para que se me DESAPAREZCAN esas líneas INTERMITENTES que me están seleccionando el rango de las dos columnas que COPIE y PEGUE ESPECIALMENTE. Es lo que se conoce como desactivar el Clipboard, pues si deseara pegarlo en otro lado, lo podría hacer, pues Excel lo ha guardado en memoria. Como no lo quiero más, lo desactivo dando escape.

Eso genera este código:

```
Sub Macro15()
'
' Macro15 Macro
'
'
Columns("B:C").Select
Selection.Copy
Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
:=False, Transpose:=False
Application.CutCopyMode = False
End Sub
```

Cuando DI ESCAPE CON EL TECLADO, se quita la intermitencia del rango, y corresponde a la instrucción **Application.CutCopyMode = False**

Esta macrito que acabo de hacer, el numero que sea, que se generó en el libro CG03013, la corto y la pego en donde estoy pegando todas las macros.

Este archivo ya está cerca de finalizarse. Ya hemos logrado:

1. Limpiarle la basura de los encabezados del listado.
2. Detectar las filas que corresponden a las cuentas (con los blancos de la columna de la derecha)
3. Insertar dos columnas y lograr desplazar los códigos de las cuentas y sus nombres a un lado izquierdo, aparte de los códigos de los terceros, pues compartían la misma columna y para una base de datos esto es inadmisibile.
4. Rellenar, debajo de las cuentas desplazadas (no por la violencia), los datos de esas mismas cuentas y nombres de cuentas.

Que sigue? O, que nos falta para acabar?

1. Ahora sí, aprovechar las celdas que están en blanco FRUTO de la función concatenar que hicimos, recuerde que unimos el contenido de todas las celdas para detectar en donde había CUENTAS y en donde NO, para desplazarlas y rellenar ESA información al lado de los terceros. Como ya hicimos el relleno, estas celdas ahora ya no las necesitamos y antes por el contrario, ahora nos sobran, pues la meta es convertir eso en forma de base de datos. Entonces: Paso siguiente: Deshacernos de todas las celdas de la columna de la extrema derecha que quedaron en blanco. Aquí le pego una imagen para que observe:

Serial	TITULO											
1	1	ACTIVO										
2	2	DISPONIBLE										
3	3	CAJA										
4	10000	CAJA GENERAL										
5	10005	CAJAS MENORES										
6	10010	CAJAS MENORES										
7	1001000	CAJA MENOR BOGOTA										
8	1001000	CAJA MENOR BOGOTA										
9	10015	MONEDA EXTRANJERA										
10	10015	MONEDA EXTRANJERA										
11	100	BANCOS										
12	10005	MONEDA NACIONAL										
13	1000500	BANCO SUDAMERIS COLOMBIA										
14	1000500	BANCO SUDAMERIS COLOMBIA	188142138									
15	1000506	BANCO DE OCCIDENTE										
16	1000506	BANCO DE OCCIDENTE	2514125828									
17	1000507	BANCOLOMBIA										
18	1000507	BANCOLOMBIA	*****									

La que pinté de grisecito de la derecha, es la que vamos a USAR para deshacernos de las filas que ya nos sobran. Para mayor claridad, en esta siguiente imagen, le estoy mostrando cuales son las filas que ya no nos sirven y que hay que borrar, pues para que la información sea trabajable debe ser en forma de base de datos. Esta imagen que le presento es la misma que esta de arriba, en esta le estoy señalando las filas que vamos a proceder a borrar:

Serial	TITULO	D	E	F	G	H	I	J	K	L
1	ACTIVO	4353	43534353	rd	rd	rd	rd	rd	rd	rd
2	DISPONIBLE									
3	CAJA									
4	CAJA GENERAL									
5	CAJAS MENORES			0	11833.098.872,68	11833.098.872,68		0	81833888872,68833098872,68	
6	CAJA MENOR BOGOTA									
7	CAJA MENOR BOGOTA			1900.000,00				1.588.888,00	7900000000000000	
8	MONEDA EXTRANJERA									
9	MONEDA EXTRANJERA			0	3.477.000,00	11615,00		3.385.385,00	5347700000000000	
10	BANCOS									
11	BANCO NACIONAL									
12	BANCO SUOMERIS COLOMBIA			1481421,30	12.789.550,71	10.523.574,88		567.387,13	6048131610050,789323874,88887387,13	
13	BANCO SUOMERIS COLOMBIA									
14	BANCO OCCIDENTE			25.941.250,70	1471270.400,88	1.029.646.958,88		6.772.781,63	25941250,707940889640889,086772701,63	
15	BANCO DE OCCIDENTE									
16	BANCO COLOMBIA									
17	BANCO COLOMBIA				3.829.045.965,38	3.829.632.382,28		11.238.505,67	7808483,789829645985,89629632382,287238505,67	

Si se fija en esta imagen, las filas que YA NO NECESITAMOS CORRESPONDEN CON EL HECHO, DE QUE EN LA COLUMNA L ESTAN VACIAS O ESTAN EN BLANCO. Ahora, NO NOS VAMOS A PONER A ELIMINAR ESTAS FILAS UNA A UNA, NOOOO. Las vamos a eliminar con la técnica que hemos venido indicando en el libro, con un filtro. Entonces: Vamos a grabar la siguiente macro que nos eliminará estas filas que ya no queremos:

Encienda nuevamente el grabador de macros por la opción ficha programador, grabar macro, en este libro. No se preocupe si el libro no es el mismo en el cual esta guardando todas las macros, luego con el editor de visual basic, va a la macro grabada en ese libro, la corta como cuando corta texto en Microsoft Word, se pasa EN EL VISUAL BASIC al módulo del archivo en donde está acumulando todas las macros parciales, que en este caso es "EjemploMacroDepurarCG03013.xlsm" y la pega debajo de las demás.

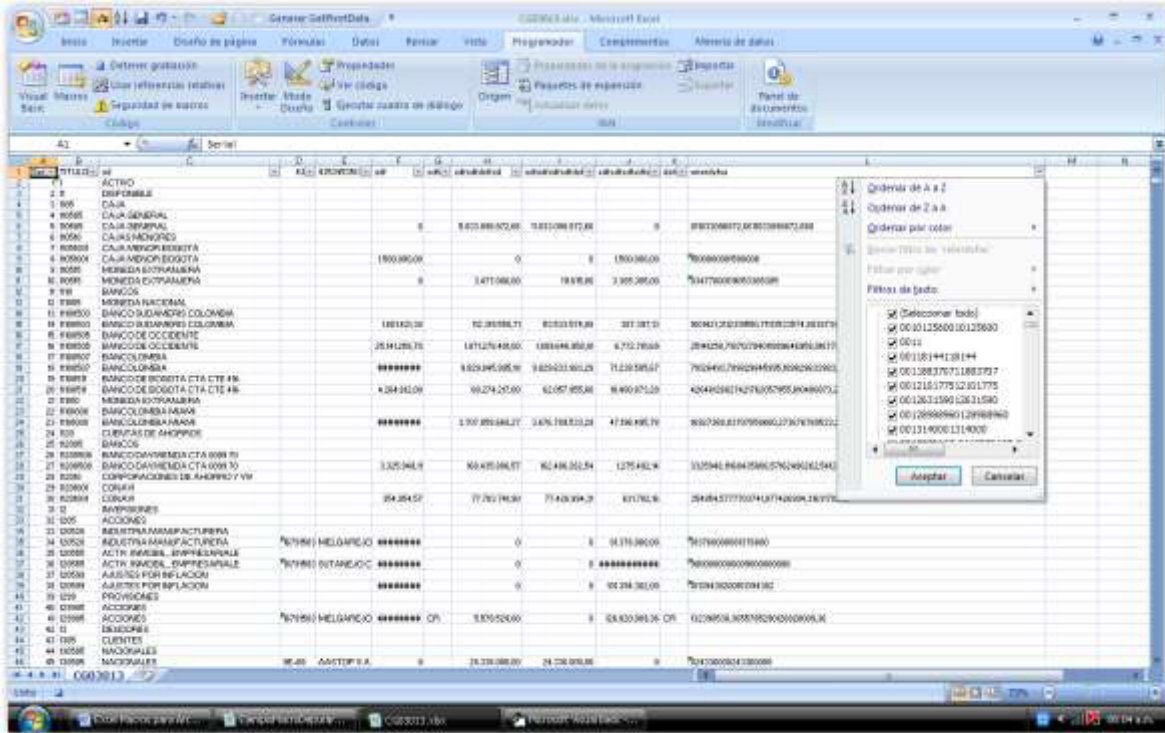
Entonces, recapitulando: prende el grabador de macros, y pone un filtro. Dado que el archivo ya tiene títulos en los encabezados de columna, y tiene un número serial para todas las filas, Excel reconocerá TODA la tabla, o región corriente sin necesidad de seleccionar todo el rango.

Entonces, recapitulando: prende el grabador de macros y pone un filtro. VA A SELECCIONAR EL FILTRO DE LA COLUMNA QUE ESTA EN GRIS, que en mi caso se llama "reterdvfx". Recuerde que he puesto encabezados solo para efectos de que los filtros trabajen. Más adelante le identificaremos correctamente los encabezados. No los identifico aún, porque estas columnas NO SON CON LAS QUE NOS VAMOS A QUEDAR. Pues también hay que arreglar los valores, y eso ya hemos visto como hacerlo en otras partes de este libro.

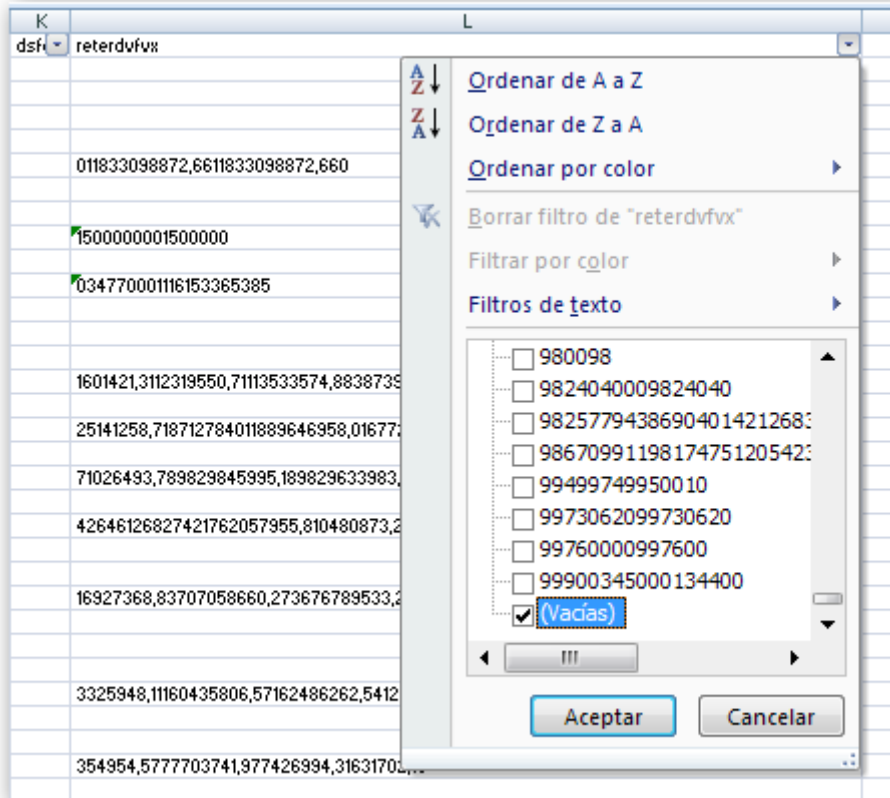
Entonces, vuelvo y recapitulo que es lo que vamos a hacer: Prende el grabador de macros, pone un filtro, selecciona la columna donde están los blancos, y **DES-selecciona TODO**, busca las que están en BLANCO y las selecciona. Que ocurrirá? Que Excel NOS DEJARÁ JUNTICAS TODAS LAS CELDAS QUE YA NO NECESITAMOS. TENIENDOLAS JUNTAS, PROCEDEMOS A ELIMINAR TODAS LAS FILAS.

Entonces: Le muestro en imágenes todo lo anterior:

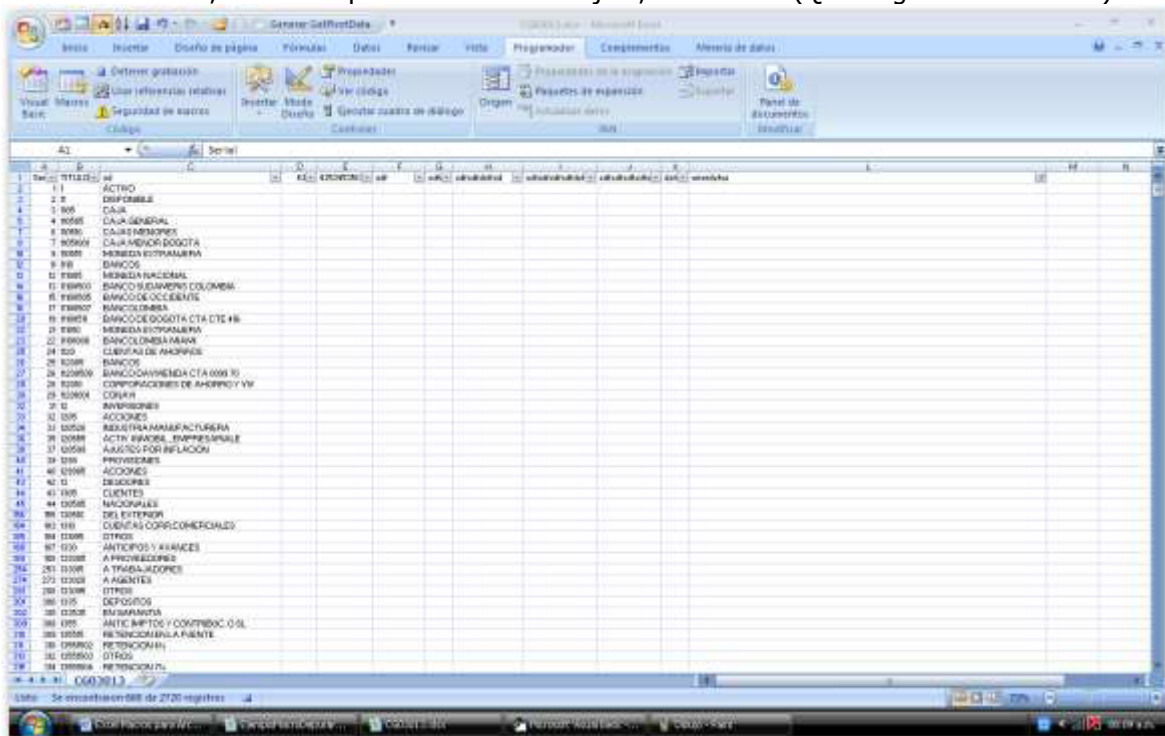
Paso 1: GRABA ESCENA MACRO SIGUENTE (como en las películas), Pongo el filtro:



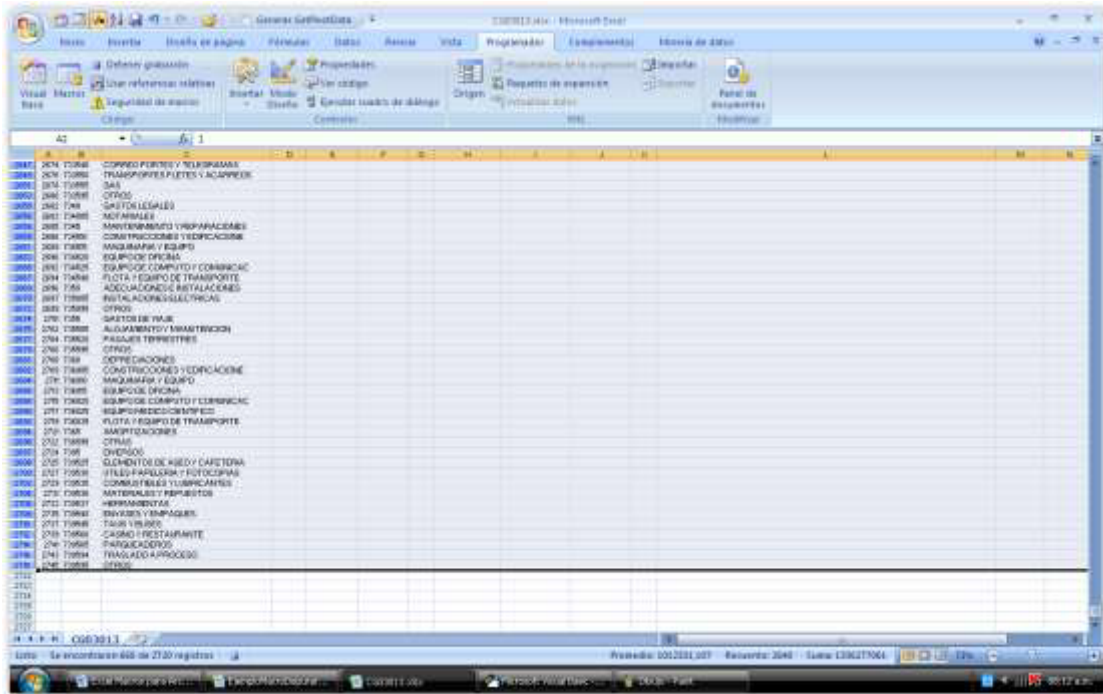
UBICADO EN EL FILTRO, Desactivo la casilla de verificación Donde dice Seleccionar todo, Bajo con la barra de desplazamiento DEL FILTRO hasta la última parte, esta: y ACTIVO LAS VACIAS, ASI:



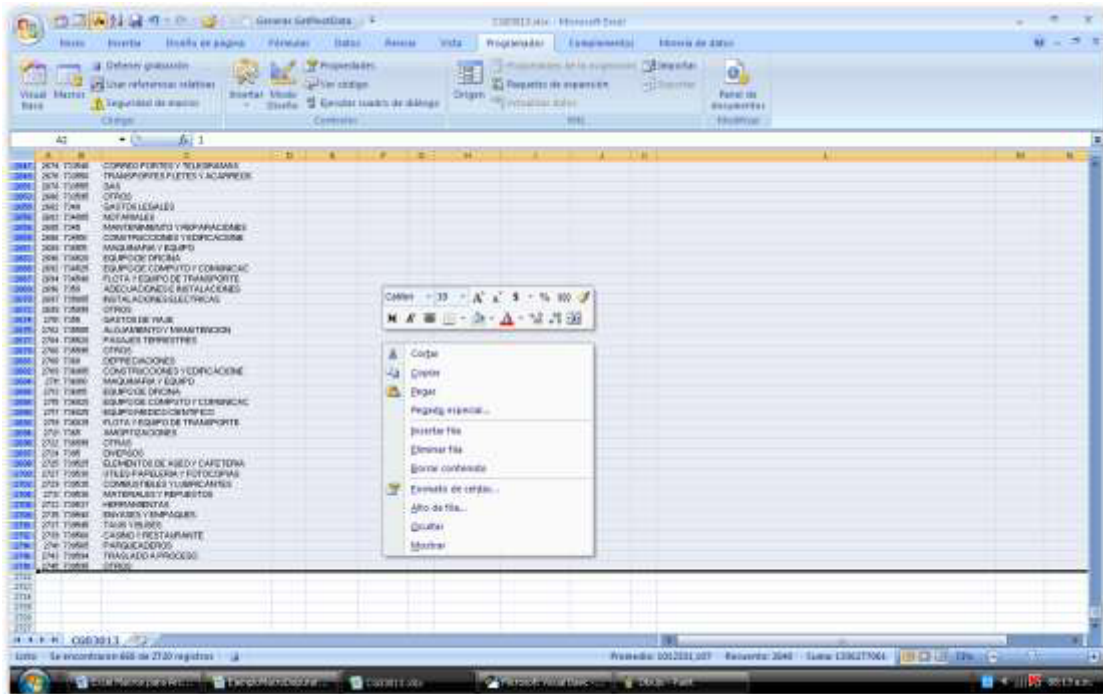
En cuanto LO HAGA, sucederá que Excel solo nos dejará, Las vacías: (Que valga la redundancia).



Puede darse cuenta, como el filtro nos hace la tareíta en un segundo, y lo mejor de todo, es que NADA DE TODO ESTO TENDREMOS QUE VOLVERLO A HACER PORQUE ESTAMOS GRABANDO TODOS LOS PASOS EN UNA MACRO. Sigue: Selecciona todas las filas, desde la segunda visible, hasta la última:



Luego Le da BOTON DERECHO (aparecerá un menú emergente), y ELIMINAR FILA:



Para que se vea más claro, aquí le pongo un ACERCAMIENTO:

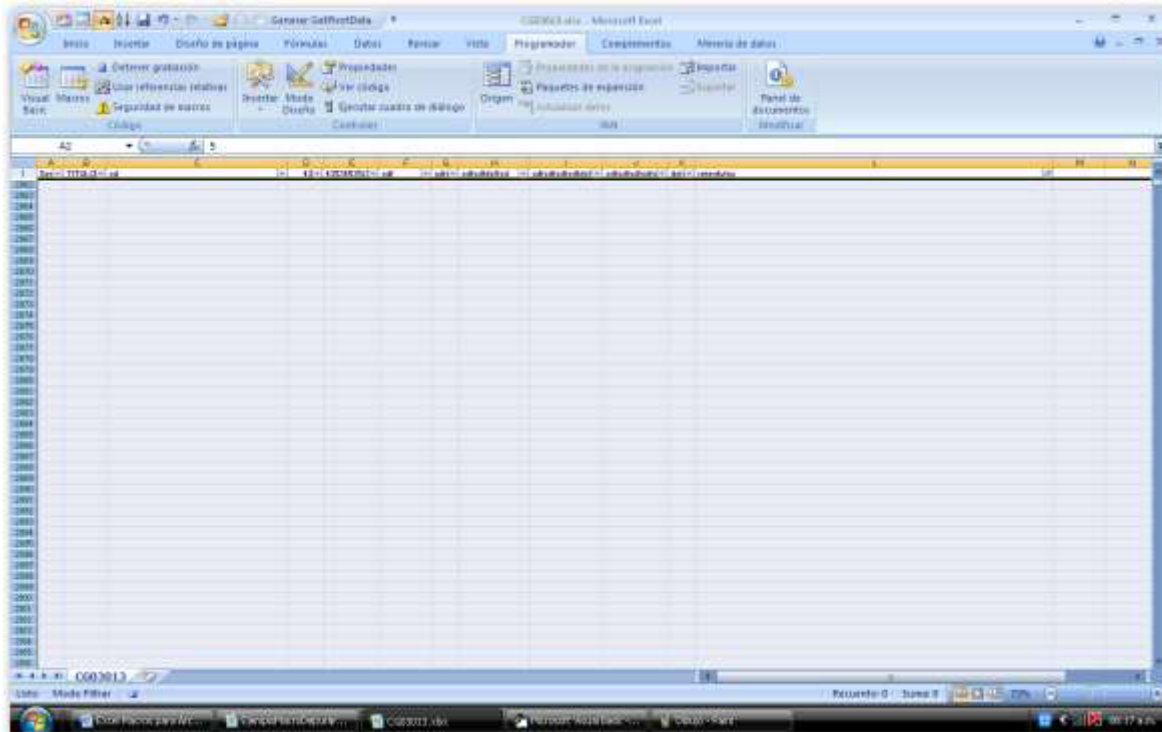
2665	2692	734525	EQUIPO DE COMPUTO Y COMUNICAC
2667	2694	734540	FLOTA Y EQUIPO DE TRANSPORTE
2669	2696	7350	ADECUACIONES E INSTALACIONES
2670	2697	735005	INSTALACIONES ELECTRICAS
2672	2699	735095	OTROS
2674	2701	7355	GASTOS DE VIAJE
2675	2702	735505	ALOJAMIENTO Y MANUTENCION
2677	2704	735520	PASAJES TERRESTRES
2679	2706	735595	OTROS
2681	2708	7360	DEPRECIACIONES
2682	2709	736005	CONSTRUCCIONES Y EDIFICACIONE
2684	2711	736010	MAQUINARIA Y EQUIPO
2686	2713	736015	EQUIPO DE OFICINA
2688	2715	736020	EQUIPO DE COMPUTO Y COMUNICAC
2690	2717	736025	EQUIPO MEDICO CIENTIFICO
2692	2719	736035	FLOTA Y EQUIPO DE TRANSPORTE
2694	2721	7365	AMORTIZACIONES
2695	2722	736595	OTRAS
2697	2724	7395	DIVERSOS
2698	2725	739525	ELEMENTOS DE ASEO Y CAFETERIA
2700	2727	739530	UTILES PAPELERIA Y FOTOCOPIAS
2702	2729	739535	COMBUSTIBLES Y LUBRICANTES
2704	2731	739536	MATERIALES Y REPUESTOS
2706	2733	739537	HERRAMIENTAS
2708	2735	739540	ENVASES Y EMPAQUES
2710	2737	739545	TAXIS Y BUSES
2712	2739	739560	CASINO Y RESTAURANTE
2714	2741	739565	PARQUEADEROS
2716	2743	739594	TRASLADO A PROCESO
2718	2745	739595	OTROS
2722			
2723			
2724			

Calibri 10 A \$ % 000

N K [Bulleted List] [Numbered List] [Decrease Indent] [Increase Indent] [Align Center] [Align Right] [Text Color] [Background Color]

- Cortar
- Copiar
- Pegar
- Pegado especial...
- Insertar fila
- Eliminar fila
- Borrar contenido
- Formato de celdas...
- Alto de fila...
- Ocultar
- Mostrar

En cuanto elimine las filas, se le desaparecerá todo. NO SE PREOCUPE, o NO SE ASUSTE:



Fíjese que hay un filtro, UN EMBUDITO en la columna L:

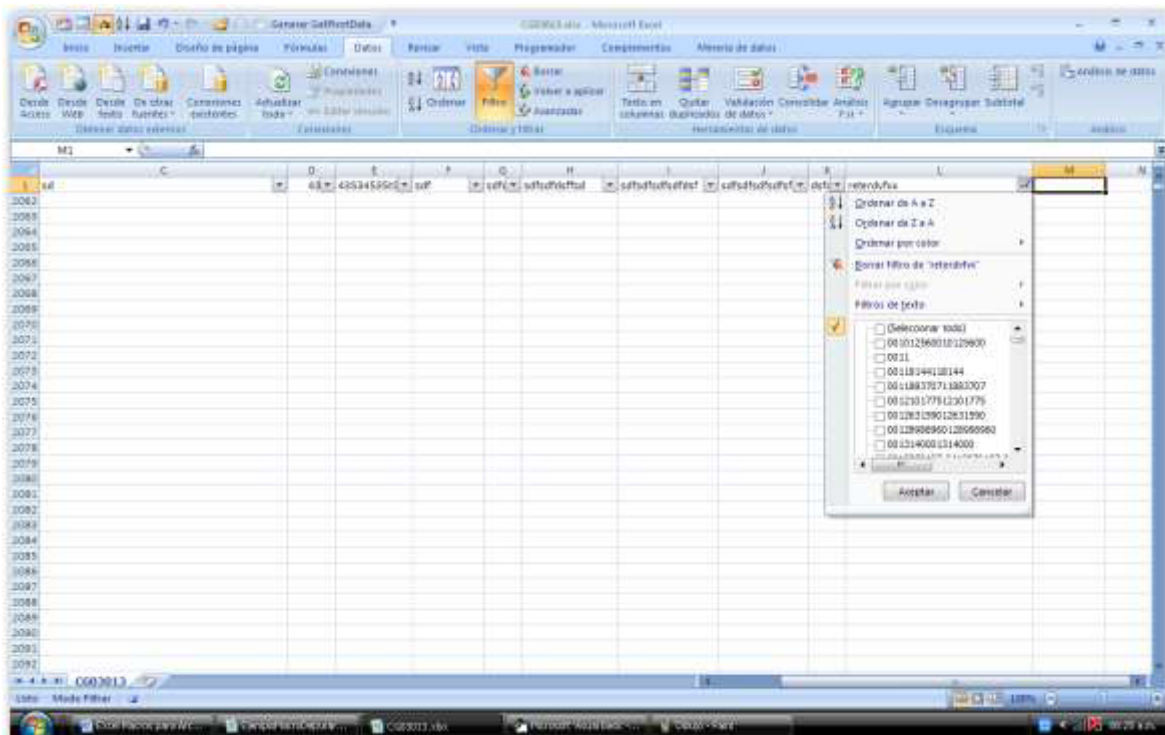
	K	L
1	dsfds	reterdvfvx
2062		

Lo ve? Aquí le hago otro ZOOM:

ESTE que se ve a la derecha con una flechita hacia abajo:

L
reterdvfvx

Cada vez que una hoja tenga filtros, FIJESE EN DONDE ESTAN LOS EMBUDITOS, así sabrá que esas columnas Están filtrando algún dato, dado algún criterio del usuario. En este caso, nuestro criterio fue que nos seleccionara todas las celdas en blanco, Como las acabamos de eliminar, pues ya no nos muestra nada. Lo que tenemos que hacer, es Hacer clic allí y darle clic a Seleccionar todo y Aceptar:



Esta operación de eliminar las filas que no nos interesan, nos generó el siguiente código:

```
Sub Macro16()
'
' Macro16 Macro
Selection.AutoFilter
ActiveSheet.Range("$A$1:$L$2721").AutoFilter Field:=12, Criteria1:=""
Rows("2:2").Select
Range(Selection, Selection.End(xlDown)).Select
Selection.Delete Shift:=xlUp
End Sub
```

Aquí le estoy señalando el número 2721, que corresponde a la última fila del archivo. Esa Celda, esa instrucción habrá que corregirla, pues seguramente otro archivo podría tener más o menos filas. Eso lo arreglaremos con una variable, la variable UltimaFila que en la sección de macros aprendimos a hacer. Recuerda? Espero que sí, si no, le agradezco retorne a la sección de macros y repase.

Ok. Mire como queda todo después de haber eliminado las filas que nos sobran: Fíjese que la columna L, la de la derecha, que tanta utilidad nos ha dado, sigue allí, ya no tiene celdas en blanco.

Ser	TITULO	sd	43	43534535t3	sdf	s	sdfsdffsd	sdfsdffsd	sdfsdffsd	sdfsdffsd	sdfsdffsd	sdfsdffsd	sdfsdffsd
5	110505	CAJA GENERAL				0	11.833.098.872,66	11.833.098.872,66	0	0	0	0	0118330988726611833098872660
8	11051001	CAJA MENOR BOGOTA			1.500.000,00	0	0	0	1.500.000,00	1.500.000,00	0	0	150000001500000
10	110515	MONEDA EXTRANJERA			0	3.477.000,00	0	0	3.477.000,00	3.477.000,00	0	0	034770001116150965885
14	11100503	BANCO SUDAMERIS COLOMBIA			1.601.421,30	112.319.550,71	113.533.574,88	387.397,13	1.601.421,30	1.601.421,30	112.319.550,71	113.533.574,88	387.397,13
16	11100505	BANCO DE OCCIDENTE			25.141.258,70	1.871.278.401,00	1.889.646.958,01	6.772.701,69	25.141.258,70	25.141.258,70	1.871.278.401,00	1.889.646.958,01	6.772.701,69
18	11100507	BANCOLOMBIA			71.026.493,78	9.829.845.995,18	9.829.633.983,29	71.238.505,67	71.026.493,78	71.026.493,78	9.829.845.995,18	9.829.633.983,29	71.238.505,67
20	11100511	BANCO DE BOGOTA CTA CTE 416-			4.264.612,00	68.274.217,00	62.057.955,80	10.480.873,20	4.264.612,00	4.264.612,00	68.274.217,00	62.057.955,80	10.480.873,20
23	11101008	BANCOLOMBIA MIAMI			16.927.368,80	3.707.058.660,27	3.676.789.533,29	47.196.495,78	16.927.368,80	16.927.368,80	3.707.058.660,27	3.676.789.533,29	47.196.495,78
27	11200508	BANCO DAVIVIENDA CTA 0099 70			3.325.948,11	160.435.806,57	162.486.262,54	1.275.492,14	3.325.948,11	3.325.948,11	160.435.806,57	162.486.262,54	1.275.492,14
30	11201001	CONAVI			354.954,57	77.703.741,90	77.426.994,31	631.702,16	354.954,57	354.954,57	77.703.741,90	77.426.994,31	631.702,16
34	120520	INDUSTRIA MANUFACTURERA	16791503	MELGAREJO INC.	81.376.000,00	0	0	81.376.000,00	81.376.000,00	81.376.000,00	0	0	813760000081376000
36	120555	ACTIV. INMOBIL. EMPRESARIALE	18791503	SUTANEJO CORP.	9.000.000.000,00	0	0	9.000.000.000,00	9.000.000.000,00	9.000.000.000,00	0	0	9000000000009000000000
38	120599	AJUSTES POR INFLACION			101.394.302,00	0	0	101.394.302,00	101.394.302,00	101.394.302,00	0	0	10139430200101394302
41	129905	ACCIONES	16791503	MELGAREJO INC.	132.398.538,36	CR	5.578.529,00	0	132.398.538,36	132.398.538,36	CR	5.578.529,00	0

Le muestro esta otra imagen para que vea más claramente como este archivo plano ya se va pareciendo más a una base de datos:

Ser	TITULO	sd	43	43534535t3	sdf	s	sdfsdffsd	sdfsdffsd	sdfsdffsd	sdfsdffsd	sdfsdffsd	sdfsdffsd	sdfsdffsd
5	110505	CAJA GENERAL				0	11.833.098.872,66	11.833.098.872,66	0	0	0	0	0
8	11051001	CAJA MENOR BOGOTA			1.500.000,00	0	0	0	1.500.000,00	1.500.000,00	0	0	0
10	110515	MONEDA EXTRANJERA			0	3.477.000,00	0	0	3.477.000,00	3.477.000,00	0	0	0
14	11100503	BANCO SUDAMERIS COLOMBIA			1.601.421,30	112.319.550,71	113.533.574,88	387.397,13	1.601.421,30	1.601.421,30	112.319.550,71	113.533.574,88	387.397,13
16	11100505	BANCO DE OCCIDENTE			25.141.258,70	1.871.278.401,00	1.889.646.958,01	6.772.701,69	25.141.258,70	25.141.258,70	1.871.278.401,00	1.889.646.958,01	6.772.701,69
18	11100507	BANCOLOMBIA			71.026.493,78	9.829.845.995,18	9.829.633.983,29	71.238.505,67	71.026.493,78	71.026.493,78	9.829.845.995,18	9.829.633.983,29	71.238.505,67
20	11100511	BANCO DE BOGOTA CTA CTE 416-			4.264.612,00	68.274.217,00	62.057.955,80	10.480.873,20	4.264.612,00	4.264.612,00	68.274.217,00	62.057.955,80	10.480.873,20
23	11101008	BANCOLOMBIA MIAMI			16.927.368,80	3.707.058.660,27	3.676.789.533,29	47.196.495,78	16.927.368,80	16.927.368,80	3.707.058.660,27	3.676.789.533,29	47.196.495,78
27	11200508	BANCO DAVIVIENDA CTA 0099 70			3.325.948,11	160.435.806,57	162.486.262,54	1.275.492,14	3.325.948,11	3.325.948,11	160.435.806,57	162.486.262,54	1.275.492,14
30	11201001	CONAVI			354.954,57	77.703.741,90	77.426.994,31	631.702,16	354.954,57	354.954,57	77.703.741,90	77.426.994,31	631.702,16
34	120520	INDUSTRIA MANUFACTURERA	16791503	MELGAREJO INC.	81.376.000,00	0	0	81.376.000,00	81.376.000,00	81.376.000,00	0	0	81.376.000,00
36	120555	ACTIV. INMOBIL. EMPRESARIALE	18791503	SUTANEJO CORP.	9.000.000.000,00	0	0	9.000.000.000,00	9.000.000.000,00	9.000.000.000,00	0	0	9.000.000.000,00
38	120599	AJUSTES POR INFLACION			101.394.302,00	0	0	101.394.302,00	101.394.302,00	101.394.302,00	0	0	101.394.302,00
41	129905	ACCIONES	16791503	MELGAREJO INC.	132.398.538,36	CR	5.578.529,00	0	132.398.538,36	132.398.538,36	CR	5.578.529,00	0

Ok, esa columna L, que tanto nos sirvió, ahora ya no la necesitamos. Prenda el grabador de macros, seleccione esa columna y elimínela. Al hacerlo, se generará este pequeño código:

```
Sub Macro17()
'
' Macro17 Macro
Columns("L:L").Select
Selection.Delete Shift:=xlToLeft
End Sub
```

La macrito generada, al igual que las demás, córtela y péguela debajo de las otras.

Que falta? Muy Poco. Qué? Queda un problemita por resolver. Fíjese como se presentan los valores crédito en este archivo plano:

	A	B	C	D	E	F	G	H	I	J	K
1527	1829	280505	DE CLIENTES	8E+08	TEJAF		0	12.500.000,00	12.500.000,00		0
1528	1830	280505	DE CLIENTES	2E+07	TRIAF		0	11.848.000,00	11.848.000,00		0
1529	1831	280505	DE CLIENTES	8E+08	VALO		0			2.019.960,00	CR
1530	1840	310505	CAPITAL AUTORIZADO	9E+08	CARD	56.842.100,00	CR		0	56.842.100,00	CR
1531	1841	310505	CAPITAL AUTORIZADO	9E+08	CARD				12.631.590,00		12.631.590,00
1532	1842	310505	CAPITAL AUTORIZADO	2E+07	CARD	12.631.590,00	CR	12.631.590,00			0
1533	1843	310505	CAPITAL AUTORIZADO	8E+08	CARD	145.263.200,00	CR		0	145.263.200,00	CR
1534	1844	310505	CAPITAL AUTORIZADO	9E+08	CARD	28.421.090,00	CR		0	28.421.090,00	CR
1535	1845	310505	CAPITAL AUTORIZADO	1E+07	CUEV	24.043.170,00	CR		0	24.043.170,00	CR
1536	1846	310505	CAPITAL AUTORIZADO	3E+07	DOM	9.618.260,00	CR		0	9.618.260,00	CR
1537	1847	310505	CAPITAL AUTORIZADO	9E+08	GEN I	300.000.000,00	CR		0	300.000.000,00	CR
1538	1848	310505	CAPITAL AUTORIZADO	9E+08	GRUF	113.684.210,00	CR		0	113.684.210,00	CR
1539	1849	310505	CAPITAL AUTORIZADO	8E+08	HURY	113.684.210,00	CR		0	113.684.210,00	CR

Tienen una columna adyacente en la cual, en los casos en los que son valores Crédito, pone un "CR". Este balance, como cualquier otro, al final debe de darnos CERO, prueba **IMPAJARITABLE** (Agradezco sepa excusar mis modismos), de que el balance esta cuadrado o más que eso, de que a nosotros nos ha quedado Bien depurado. Aunque en la práctica, casos se ven en donde **DE UNA se pillan** balances descuadrados, aunque el mismo archivo plano, si se fija, por lo general al final muestra unos totales generales, y cuando un balance está descuadrado, allí se puede ver, que en las columnas saldo inicial y final, no están en ceros. Por otra parte, si el archivo plano reporta cuanto suman los débitos y los créditos totales, a ese valor deberemos llegar y así sabremos que el archivo, por lo menos en cuanto valores generales, nos ha quedado correcto.

Otra situación con la que me he encontrado es que, dado que en esta depuración nos estamos quedando con información de las cuentas a nivel de terceros, y al analizarla e irla mayorizando por clase, grupo, subgrupo, cuenta, en ocasiones no corresponde con las sumatorias a nivel de terceros. Es una prueba que podría uno llamar de "Integralidad de la información", en la cual se detectan inconsistencias en la acumulación de los saldos del mismo sistema. Eso se puede revisar, y podríamos cruzar, rápidamente con una tabla dinámica, contra los totales que nos mostraban los asteriscos al comienzo, en el archivo plano, pues por lo general, en estos archivos del CG1, un asterisco son los totales por clase, dos asteriscos totales por grupo, etc.

Ok. Nos falta el código que nos arregle los valores. Vamos a grabarlo: ya sabemos que nos podemos servir de las columnas en donde están los CR y aparte, en una nueva columna colocar una función =SI(), y copiarla a todo lo largo del rango. Es una manera, de solucionarlo, que a mí me gusta pues resulta ser más rápida que si me pusiera a recorrer la columna y en donde aparezca el CR multiplicar por menos uno o algo así. Créame que hacer la función SI() en una columna adyacente, a la derecha, y copiar las formulas hacia abajo es muchísimo más rápido que obligar Excel a que recorra cada columna arreglando los valores uno a uno.

Ok. En Excel, nos ubicamos en el archivo plano. Prendo nuevamente el grabador de macros. Me ubico en la celda en la cual voy a montar la función =SI(), que en este caso será la L2, y en ella introduciremos la siguiente función:

```
=SI(G2="CR";-VALOR(F2);VALOR(F2))
```

La cosa es que, cada vez que aparece un CR, el valor es crédito, es decir, NEGATIVO, por eso, lo que hacemos es traer el valor negativo, utilizo la función valor, POR SI DE PRONTO sigue apareciendo algún numero que Excel lo está tomando como Texto. De esta manera me REASEGURO de que definitivamente quede negativo. En caso de que no haya un CR, solo traeré el valor de la celda, usando también, la función Valor.

En M2 introduciremos : =VALOR(H2) que corresponde a los débitos del archivo plano

En N2 introduciremos : =VALOR(I2) son los créditos del archivo plano

En O2 introduciremos : =L2+M2-N2 el saldo final, podemos optar por sumar las funciones anteriores. Detenemos la macro.

Ahora, antes de ver el código que generó Excel, veamos gráficamente la introducción de la función =Si():

	E	F	G	H	I	J	K	L
1	43	sdf	s	sdfsdfdsffsd	sdfsdfdsdfds	sdfsdfdsdfsfsc	ds	
2			0	11.833.098.872,66	11.833.098.872,66		0	0
3		1.500.000,00		0	0	1.500.000,00		
4		0		3.477.000,00	111.615,00	3.365.385,00		

Esa es la función que está en L2, que en este caso devuelve un cero. Si en L2 le damos F2 Editar, podemos ver las celdas que está tomando, o ver mejor como está montada la formula:

	E	F	G	H	I	J	K	L	M	N
1	43	sdf	s	sdfsdfdsffsd	sdfsdfdsdfds	sdfsdfdsdfsfsc	ds			
2			0	11.833.098.872,66	11.833.098.872,66		0	=SI(G2="CR";-VALOR(F2);VALOR(F2))		
3		1.500.000,00		0	0	1.500.000,00				

Ok. Después de haber ingresado esas funciones y haber detenido la macro, este es el código que se generó:

```
Sub Macro20 ()
'
' Macro20 Macro
'
Range("L2").Select
ActiveCell.FormulaR1C1 = "=IF (RC[-5]=""CR"",-VALUE (RC [-6] ) , VALUE (RC [-6] ) )"
Range("M2").Select
ActiveCell.FormulaR1C1 = "=VALUE (RC [-5] )"
Range("N2").Select
ActiveCell.FormulaR1C1 = "=VALUE (RC [-5] )"
Range("O2").Select
ActiveCell.FormulaR1C1 = "=RC [-3] +RC [-2] -RC [-1] "
End Sub
```

Ok, ahora en la siguiente macro que grabaremos, vamos a tomar estas cuatro celdas, copiarlas y pegarlas a todo lo largo del rango del archivo plano.

Una manera de hacerlo, puede ser:

1. Seleccionar las celdas, copiarlas, pasarnos a la celda A1 o a2, con el comando FIN mas flecha hacia abajo, iremos hasta la última fila del archivo, nuevamente con FIN mas flecha a la derecha iremos a la esquina inferior derecha, nos desplazaremos una casilla mas a la derecha, y en ese punto, ya estaremos ubicados en el punto del rango final. Desde este punto final

estaremos correctamente ubicados y seleccionando, desde este punto final, hacia arriba, tomaremos todo el rango de celdas que queremos pegar.

El inconveniente es que, si nos pasamos a A1 o A2, para que la técnica del FIN + flechita de dirección, funcione, deben de existir datos a todo lo largo de la dirección sobre la que nos estamos moviendo. Si llegara a haber un espacio en blanco, Fin + Flechita podría detenerse a mitad de camino, y es lo que sucederá en este caso, pues el archivo tiene muchas celdas en blanco, como es el caso de la columna G, columna que contiene o no los caracteres CR, y sobre ella pasaremos cuando al final demos Fin + flechita a la derecha.

2. Otra manera es, en lugar de pasarnos a la celda A1 y bajar hasta el final, pasarnos solo hasta la columna J, que representa el saldo final, y deslizarlos hasta el final por allí, luego nos desplazamos dos celdas a la derecha y ya tendremos el punto final del rango sobre el cual pegaremos estas cuatro celdas con sus formulas.

El inconveniente? Igualmente no podemos arriesgarnos a que haya un espacio en blanco que es poco probable que lo haya, pero puede existir incluso por error, y podría detenerse a mitad de camino. Entonces, tampoco. Busquemos otra opción:

3. Considero que la mejor opción es, arrancar desde la última celda, en la columna A, de la hoja de cálculo, la 65.536 si es Excel 2003 o la 1.048.576 si estamos en Excel 2007. En la sección de macros aprendimos un comando que sirve para ambos casos, pero en este caso queremos hacerlo usando el grabador de macros, por lo que debemos de saber con certeza en que versión de Excel estamos trabajando. Así pues, la mejor opción, para encontrar la última fila, es ir hasta la última celda de Excel, y subir hasta estrellarnos con el último dato de la región corriente, por la columna A. ¿Porque la A y no la J la columna del saldo final? Respuesta: Porque hay casos en que las demás columnas no tienen la misma longitud que la primer columna, y queremos estar exentos de error. Luego, **INEVITABLEMENTE** tendremos que escribir **MANUALMENTE** el código que desplaza la celda activa hasta la columna L. Recuerde que comenzamos a escribir la función =SI() en L2, así que la columna L-Ultima fila, será el punto final del rango sobre el cual pegaremos la función que estamos copiando.

Vamos a hacerlo:

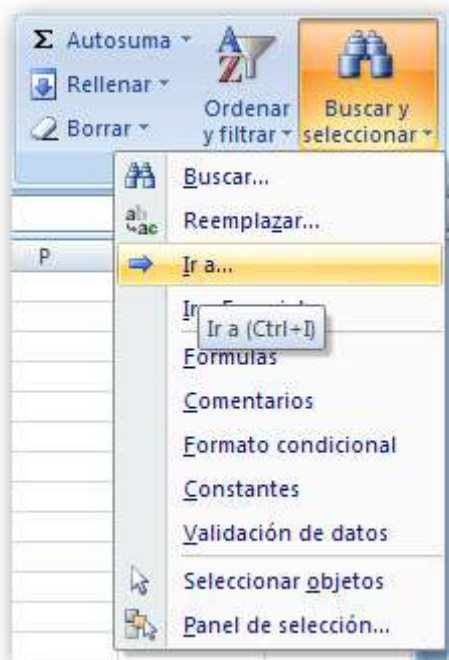
Prendemos otra vez el grabador de macros. Selecciono las cuatro celdas que contienen las formulas:

J	K	L	M	N	O
sdfsdfsdfsdfsfsfc	ds				
0		0	11833098873	11833098873	0
1.500.000,00					

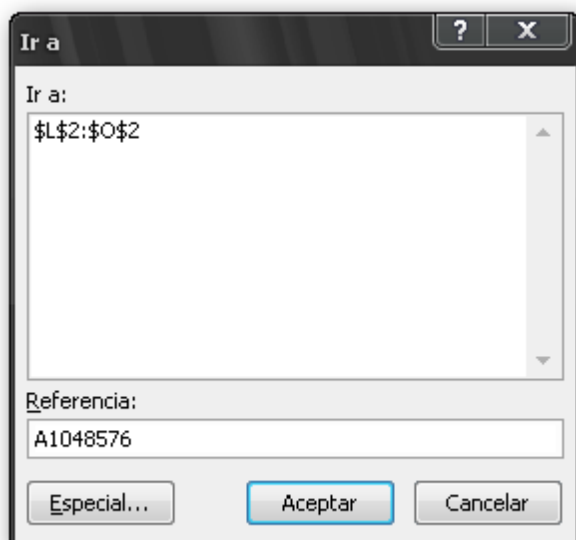
Las copio:

J	K	L	M	N	O	P
sdfsdfsdfsdfsfsfc	ds					
0		0	11833098873	11833098873	0	
1.500.000,00						

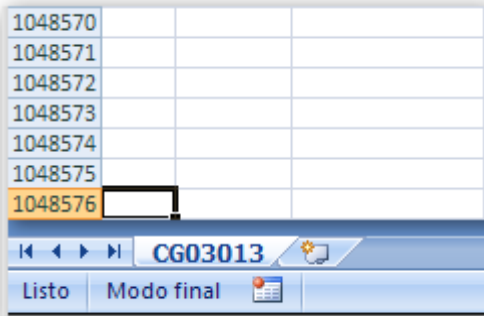
Me voy por la ficha INICIO – Buscar y seleccionar – Ir a (puede ver que hay un short-cut o atajo de teclado que Excel nos está diciendo que también podemos usar para evitar esos Clic adicionales de ratón, control + I)



En cuanto salga la ventana, escribiremos las coordenadas o referencia de la última celda en la columna A, esto es A1.048.576:



Clic en Aceptar. Una vez que llegemos a la ultima celda, oprimiremos FIN + celda hacia arriba:



Puede ver que al lado de "Listo" dice "Modo Final", quiere decir que ya oprimí la tecla FIN y solo flat oprimir la flecha hacia arriba para salir disparado y estrellarme sin lugar a dudas con la ultima celda:

	A	B	C
2061	27521739595		OTROS
2062			
2063			
2064			

En este punto, detenemos la macro y vamos al código que se generó, y completarlo:

```
Sub Macro23 ()
'
' Macro23 Macro
'
'
'
Range("L2:O2").Select
Selection.Copy
Application.Goto Reference:="R1048576C1"
Selection.End(xlUp).Select
End Sub
```

No se fije en el número de macro por la que voy, dado que como estoy escribiendo el libro mientras hago las macros, y grabo las imágenes, no me están quedando en secuencia, pero eso no es problema, solo pegue la macro, que esté correcta. Observe el código, mire que todo eso a lo que le hemos "botado corriente" sobre como copiar y pegarlo en el rango correcto sin error, queda reducido esas cuatro líneas.

Este pequeño código ya tiene la instrucción de copiar e ir hasta la última celda de la columna A, del archivo plano. Solo necesitamos que se SELECCIONE la columna L. La columna L es la columna número 12, o la letra numero 12, si cuenta desde A, hasta L.

La instrucción que debe escribir, debajo de esta macro, es la instrucción `ActiveCell.Offset` que se explicó ampliamente en la parte teórica. Solo se necesita escribir esta línea:

`ActiveCell.Offset(0, 11).Select`

Porque escribo 11 si la columna L es la columna 12? Porque si se mueve, desde A, doce veces, quedará ubicado en la columna M, en lugar de L. Cuando usted camina y comienza a contar pasos, NO CUENTA el punto inicial, cuenta desde el primer paso, el paso que lo llevará a la posición o punto 2. Entonces es lo mismo aquí, es cuestión de cuantos pasos o celdas se debe mover Excel. Cero filas, 11 columnas a la derecha.

Recuerde que las MINIMACROS como esta, se pueden probar, ejecutar paso a paso, así que si se equivoca en la cantidad de posiciones, pues rectifique, hasta que encuentre la referencia o coordenada de desplazamiento correcta. Es imprescindible que para una mejor comprensión de esta parte haya leído la parte teórica que hacía referencia al tema de ActiveCell.Offset en el libro.

Una vez incorporada la línea en el código, y comprobada queda así:

```
Sub Macro23 ()
'
' Macro23 Macro
'
'
Range("L2:O2").Select
Selection.Copy
Application.Goto Reference:="R1048576C1"
Selection.End(xlUp).Select
ActiveCell.Offset(0, 12).Select
End Sub
```

Este es, en la práctica, de las pocas cositas en las cuales se debe "meterle la mano" al código. Ok, ahora, desde este punto podemos seguir usando el grabador de macros, para la parte que nos falta. Entonces, esta macro la corremos, y cuando termina fíjese que nos está pidiendo que seleccionemos el destino y que peguemos. Ubicados aquí, en donde nos deja, grabaremos el pedazo de macro que nos falta.

	I	J	K	L	M	N	O
2053	925.719,44	113.858.389,02					
2054	0	26.701.641,40					
2055	0	3.428.400,00					
2056	0	8.648.929,00					
2057	0	1.339.660,00					
2058	4.804.917.577,71	1.856.961.030,04	CR				
2059	36.721.574,00	108.105,00					
2060	174.514.022.592,37	0					
2061	174.514.022.592,37	0					
2062							
2063							
2064							
2065							
2066							

Ubicados en el punto final del rango, prendemos el grabador de macros y lo que hacemos es oprimir las teclas Shift + FIN + flecha arriba. En cuanto se seleccione todo el rango, se da ENTER. Sucederá que quedará pegada, a todo lo largo del rango, las formulas de las cuatro celdas que estamos copiando. Seguidamente, de una vez allí, con TOODOO el rango seleccionado, de nuevo copiamos y pegamos, esta vez con el propósito de DESBARATAR las formulas, daremos PEGADO ESPECIAL SOLO VALORES. Puesto que hace un momento hicimos esa operación de copiar y pegar solo valores, no considero necesario volver a pegar las imágenes.

Una vez hecho lo anterior, esto es lo que nos queda:

I	J	K	L	M	N	O
sdfsdfsdfsdfs	sdfsdfsdfsdfsfc	ds				
11.833.098.872,66	0	0	11833098873	11833098873	0	0
0	1.500.000,00	1500000	0	0	1500000	0
111.615,00	3.365.385,00	0	3477000	111615	3365385	0
113.533.574,88	387.397,13	1601421,3	112319550,7	113533574,9	387397,13	0
1.889.646.958,01	6.772.701,69	25141258,7	1871278401	1889646958	6772701,69	0
9.829.633.983,29	71.238.505,67	71026493,78	9829845995	9829633983	71238505,67	0
62.057.955,80	10.480.873,20	4264612	68274217	62057955,8	10480873,2	0
3.676.789.533,29	47.196.495,78	16927368,8	3707058660	3676789533	47196495,78	0
162.486.262,54	1.275.492,14	3325948,11	160435806,6	162486262,5	1275492,14	0
77.426.994,31	631.702,16	354954,57	77703741,9	77426994,31	631702,16	0
0	81.376.000,00	81376000	0	0	81376000	0
0	9.000.000.000,00	9000000000	0	0	9000000000	0
0	101.394.302,00	101394302	0	0	101394302	0
0	126.820.009,36 CR	-132398538	5578529	0	-126820009	0
24.336.000,00	0	0	24336000	24336000	0	0
15.243.750,00	0	15243750	0	15243750	0	0

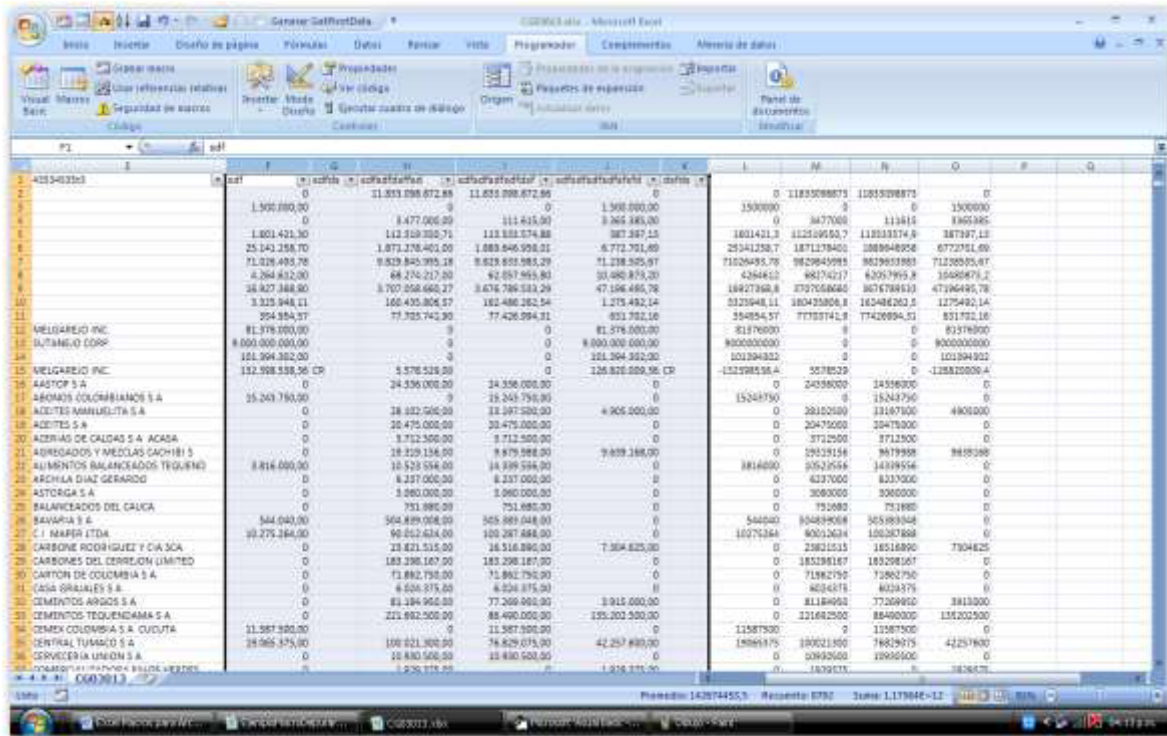
Fíjese que los 126.820.009,36 que son CREDITO, quedaron negativos. Quiere decir que la función opera correctamente y cumple su cometido. Esta operación de seleccionar el rango, pegar, volver a copiar y pegar solo valores, generó este código, que deberá ir debajo del otro que manipulamos manualmente:

```
Sub Macro24()
'
' Macro24 Macro
'
'
Range(Selection, Selection.End(xlUp)).Select
ActiveSheet.Paste
Application.CutCopyMode = False
Selection.Copy
Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
:=False, Transpose:=False
Application.CutCopyMode = False
End Sub
```

Después de esto, YA NO NOS HACEN FALTA LAS COLUMNAS DESDE LA F HASTA LA COLUMNA K. Estas son las columnas viejas del archivo plano que representaban el saldo inicial, la columna de los CR, los débitos, los créditos y la columna de los CR. Puesto que ya organizamos los valores y los convertimos en negativos cuando son CR'S (créditos), podemos proceder a Eliminar esas columnas.

Entonces, nuevamente prenda el grabador de macros, seleccione esas columnas y elimínelas. Detenga la macro y pegue ese código debajo de los que venimos haciendo. En esta imagen estoy seleccionando las columnas a ELIMINAR:

Excel aplicado a la Auditoría, la Contaduría y la Administración



En esta, ya las he eliminado, y ha quedado así:

	E	F	G	H	I	J	K
1	43534535t3						
2		0	11833098873	11833098873	0		
3		1500000	0	0	1500000		
4		0	3477000	111615	3365385		
5		1601421,3	112319550,7	113533574,9	387397,13		
6		25141258,7	1871278401	1889646958	6772701,69		
7		71026493,78	9829845995	9829633983	71238505,67		
8		4264612	68274217	62057955,8	10480873,2		
9		16927368,8	3707058660	3676789533	47196495,78		
10		3325948,11	160435806,6	162486262,5	1275492,14		
11		354954,57	77703741,9	77426994,31	631702,16		
12	MELGAREJO INC.	81376000	0	0	81376000		
13	SUTANEJO CORP.	9000000000	0	0	9000000000		

Eso generará este código, como podrá ver es muy sencillo de entender, pero no por eso tiene que escribirlo, siempre que pueda use el grabador de macros:

```
Sub Macro25 ()
'
' Macro25 Macro
'
'
Columns("F:I").Select
Selection.Delete Shift:=xlToLeft
End Sub
```

Fíjese como va quedando el archivo ya depurado:

	A	B	C	D	E	F	G	H	I
1	5	TITULO	sd	43	43534535t3				
2	5	110505	CAJA GENERAL			0	11833098873	11833098873	0
3	8	11051001	CAJA MENOR BOGOTA			1500000	0	0	1500000
4	10	110515	MONEDA EXTRANJERA			0	3477000	111615	3365385
5	14	11100503	BANCO SUDAMERIS COLOMBIA			1601421,3	112319550,7	113533574,9	387397,13
6	16	11100505	BANCO DE OCCIDENTE			25141258,7	1871278401	1889646958	6772701,69
7	18	11100507	BANCOLOMBIA			71026493,78	9829845995	9829633983	71238505,67
8	20	11100511	BANCO DE BOGOTA CTA CTE 416-			4264612	68274217	62057955,8	10480873,2
9	23	11101008	BANCOLOMBIA MIAMI			16927368,8	3707058660	3676789533	47196495,78
10	27	11200508	BANCO DAVIVIENDA CTA 0099 70			3325948,11	160435806,6	162486262,5	1275492,14
11	30	11201001	CONAVI			354954,57	77703741,9	77426994,31	631702,16
12	34	120520	INDUSTRIA MANUFACTURERA	16791503	MELGAREJO INC.	81376000	0	0	81376000
13	36	120555	ACTIV. INMOBIL., EMPRESARIALE	18791503	SUTANEJO CORP.	9000000000	0	0	9000000000
14	38	120599	AJUSTES POR INFLACION			101394302	0	0	101394302

Solo nos falta colocar los encabezados de columna y colocarle el formato a los números. Prenda el grabador de macros, seleccione las columnas de los valores, esto es, de la F hasta la I, haga clic derecho para que salga el menú emergente, y coloque el formato de celdas NUMERO de su predilección, o el que considere necesario:

E	F	G	H	I
43534535t3				
	0,00	11.833.098.872,66	11.833.098.872,66	0,00
	1.500.000,00	0,00	0,00	1.500.000,00
	0,00	3.477.000,00	111.615,00	3.365.385,00
	1.601.421,30	112.319.550,71	113.533.574,88	387.397,13
	25.141.258,70	1.871.278.401,00	1.889.646.958,01	6.772.701,69
	71.026.493,78	9.829.845.995,18	9.829.633.983,29	71.238.505,67
	4.264.612,00	68.274.217,00	62.057.955,80	10.480.873,20
	16.927.368,80	3.707.058.660,27	3.676.789.533,29	47.196.495,78
	3.325.948,11	160.435.806,57	162.486.262,54	1.275.492,14
	354.954,57	77.703.741,90	77.426.994,31	631.702,16
MELGAREJO INC.	81.376.000,00	0,00	0,00	81.376.000,00
SUTANEJO CORP.	9.000.000.000,00	0,00	0,00	9.000.000.000,00
	101.394.302,00	0,00	0,00	101.394.302,00
MELGAREJO INC.	(132.398.538,36)	5.578.529,00	0,00	(126.820.009,36)

Este es el código que se genera, el cual también vamos cortando y llevando para el otro lado:

```
Sub Macro26()
'
' Macro26 Macro
'
'
Columns("F:I").Select
Selection.NumberFormat = "#,##0.00_"; [Red] (#,##0.00)
End Sub
```

Ahora, la macro de los encabezados. Solo es prender el grabador de macros y colocar los encabezados a todas las columnas:

	A	B	C	D	E	F	G	H	I
1	Seri	Cuenta	Nombre de la cuenta	Tercero	Descripcion	Saldo inicial	Debitos	Creditos	Saldo final
2	5	110505	CAJA GENERAL			0,00	11.833.098.872,66	11.833.098.872,66	0,00
3	8	11051001	CAJA MENOR BOGOTA			1.500.000,00	0,00	0,00	1.500.000,00
4	10	110515	MONEDA EXTRANJERA			0,00	3.477.000,00	111.615,00	3.365.385,00
5	14	11100503	BANCO SUDAMERIS COLOMBIA			1.601.421,30	112.319.550,71	113.533.574,88	387.397,13
6	16	11100505	BANCO DE OCCIDENTE			25.141.258,70	1.871.278.401,00	1.889.646.958,01	6.772.701,69
7	18	11100507	BANCOLOMBIA			71.026.493,78	9.829.845.995,18	9.829.633.983,29	71.238.505,67

Genera este código:

```
Sub Macro27()
'
' Macro27 Macro
'
'
ActiveCell.FormulaR1C1 = "Serial"
Range("B1").Select
ActiveCell.FormulaR1C1 = "Cuenta"
Range("C1").Select
ActiveCell.FormulaR1C1 = "Nombre de la cuenta"
Range("D1").Select
ActiveCell.FormulaR1C1 = "Tercero"
Range("E1").Select
ActiveCell.FormulaR1C1 = "Descripcion"
Range("F1").Select
ActiveCell.FormulaR1C1 = "Saldo inicial"
Range("G1").Select
ActiveCell.FormulaR1C1 = "Debitos"
Range("H1").Select
ActiveCell.FormulaR1C1 = "Creditos"
Range("I1").Select
ActiveCell.FormulaR1C1 = "Saldo final"
End Sub
```

Podemos decir que ya hemos finalizado. Solo falta unir todas las macros y mejorarla en aquellas partes en donde quedaron celdas fijas y necesitamos que se mueva con el ActiveCell.Offset. También debemos definir la variable Ultimafila para remitirnos siempre en verdad a la última fila y no a una última fila fija de un solo archivo plano. Recuerde que si no se hacen esos arreglos la macro no le operará bien si le llega a salir un archivo diferente.

Si usted afina bien esta macro, le puede servir para cualquier compañía que maneje este tipo de archivos de CG1, el CG03013. Si es versión CG1 versión 8.5, es el archivo UFCG1041, el cual encontrará

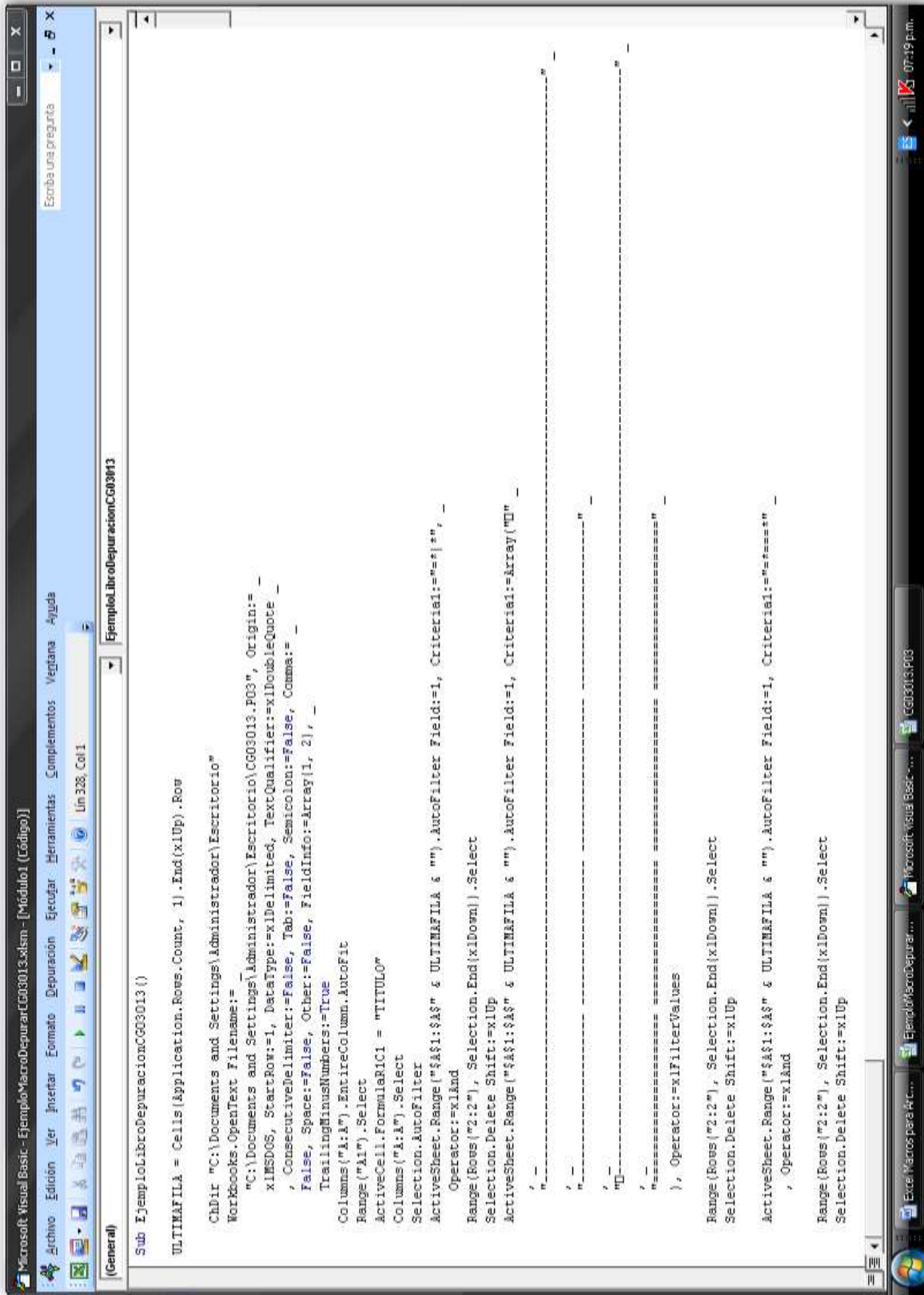
que es muy similar a este, por no decir que es casi la misma cosa, pero esta macro le servirá para el CG03013. Tendrá que construir la que corre para el UFCG1041.

Código Mejorado.

Este es el código con los cambios mínimos necesarios para que funcione. No se requiere que haya una superdepuración para que trabaje. Igual lo hará. Con el tiempo, si encuentra que le gusta el tema, le irá quitando toda la redundancia que tiene por doquier, y seguramente lo escribirá mucho mejor.

Una buena manera de unir todas las macros que se fueron colocando una debajo de la otra, es colocarles un apóstrofo. Cuando se coloca un apóstrofo, el programa ignora las líneas, pues reconoce que solo son líneas informativas. Así, el código de la primera macro seguirá derecho en la segunda y así sucesivamente. Por ello verá que los encabezados de las macros quedan, para que nemotécnicamente uno recuerde todas las macros que se construyeron para realizar toda la tarea.

He preferido colocar las imágenes del código tal como se ve en el editor del visual basic, si lo desea puedo enviarle esta macro a su correo, previa verificación de que haya realizado la compra legal del libro.




```

Microsoft Visual Basic - EjemploMacroDepurarC03013.xlsm - [Módulo1 (Código)]
Archivo Edición Ver Insertar Formato Depuración Ejecutar Herramientas Complementos Ventana Ayuda
Escriba una pregunta
EjemploMacroDepuraciónC03013

Columns("B:B").Select
Selection.Delete Shift:=xlToLeft
Range("A1").Select

'End Sub
'Sub Macro20()
'Macro20 Macro

Columns("A:A").Select
Selection.TextToColumns Destination:=Range("A1"), DataType:=xlFixedWidth,
FieldInfo:=Array(Array(0, 1), Array(11, 1), Array(45, 1), Array(73, 1), Array(75, 1),
Array(94, 1), Array(114, 1), Array(134, 1)), DecimalSeparator:=".",
ThousandsSeparator:=",", TrailingMinusNumbers:=True

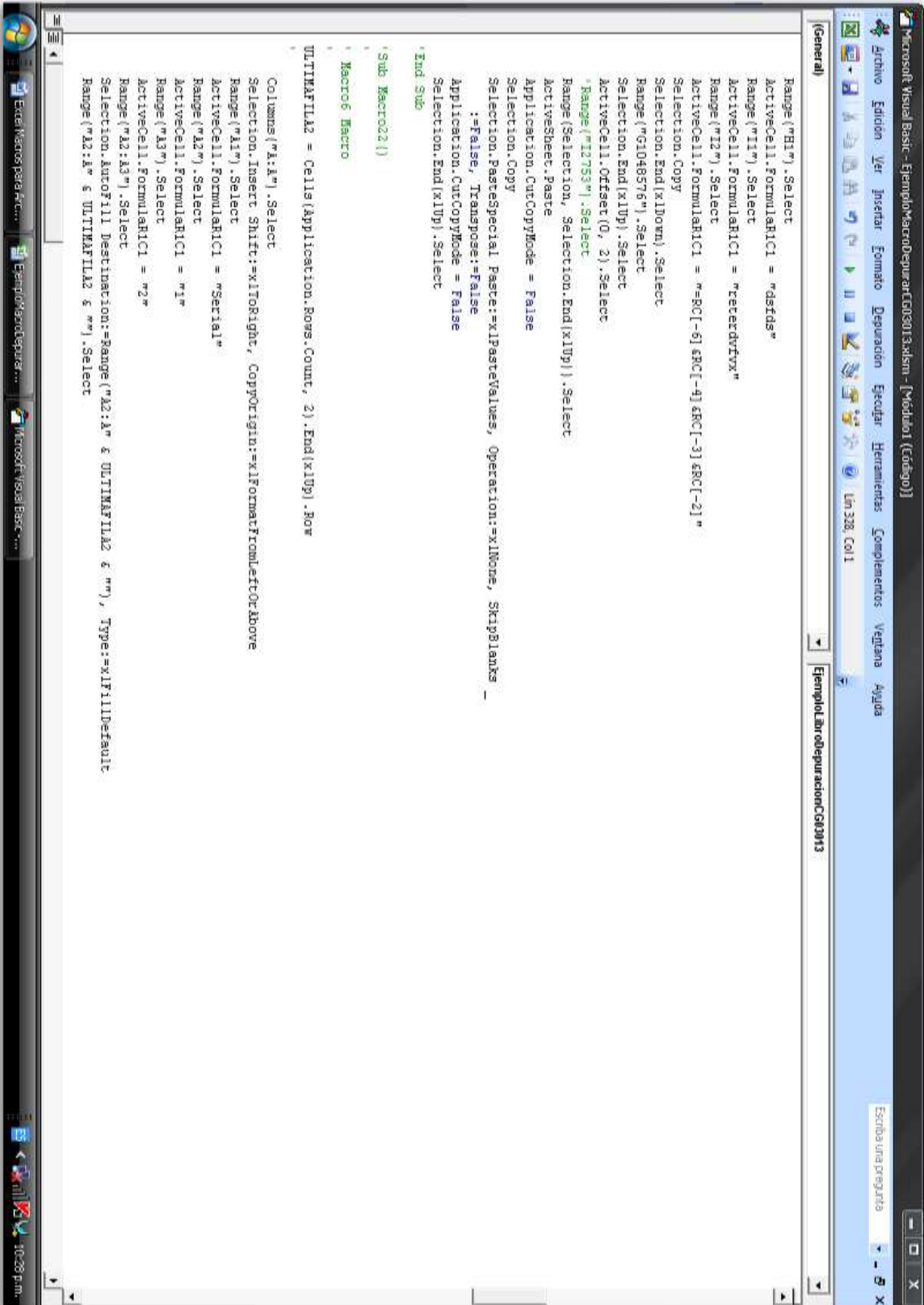
Columns("A:I").Select
Columns("A:I").EntireColumn.AutoFit
Range("A1").Select

'End Sub

'Sub Macro21()
'Macro2 Macro

Range("B1").Select
ActiveCell.FormulaR1C1 = "sd"
Range("C1").Select
ActiveCell.FormulaR1C1 = "sdf"
Range("D1").Select
ActiveCell.FormulaR1C1 = "sdfs"
Range("E1").Select
ActiveCell.FormulaR1C1 = "sdfsdfsf"
Range("F1").Select
ActiveCell.FormulaR1C1 = "sdfsdfsfdsf"
Range("G1").Select
ActiveCell.FormulaR1C1 = "sdfsdfsfdsfsfd"

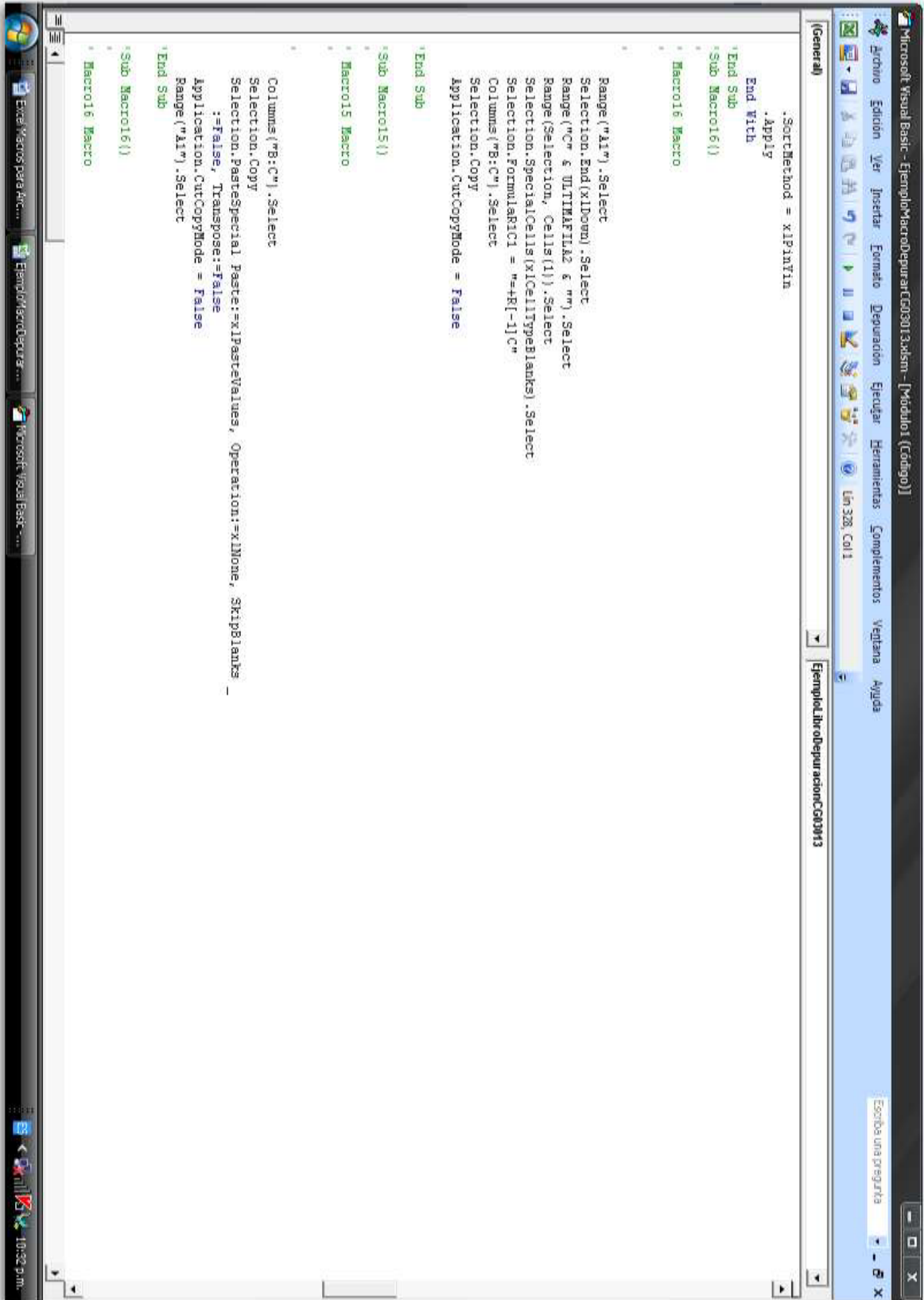
```

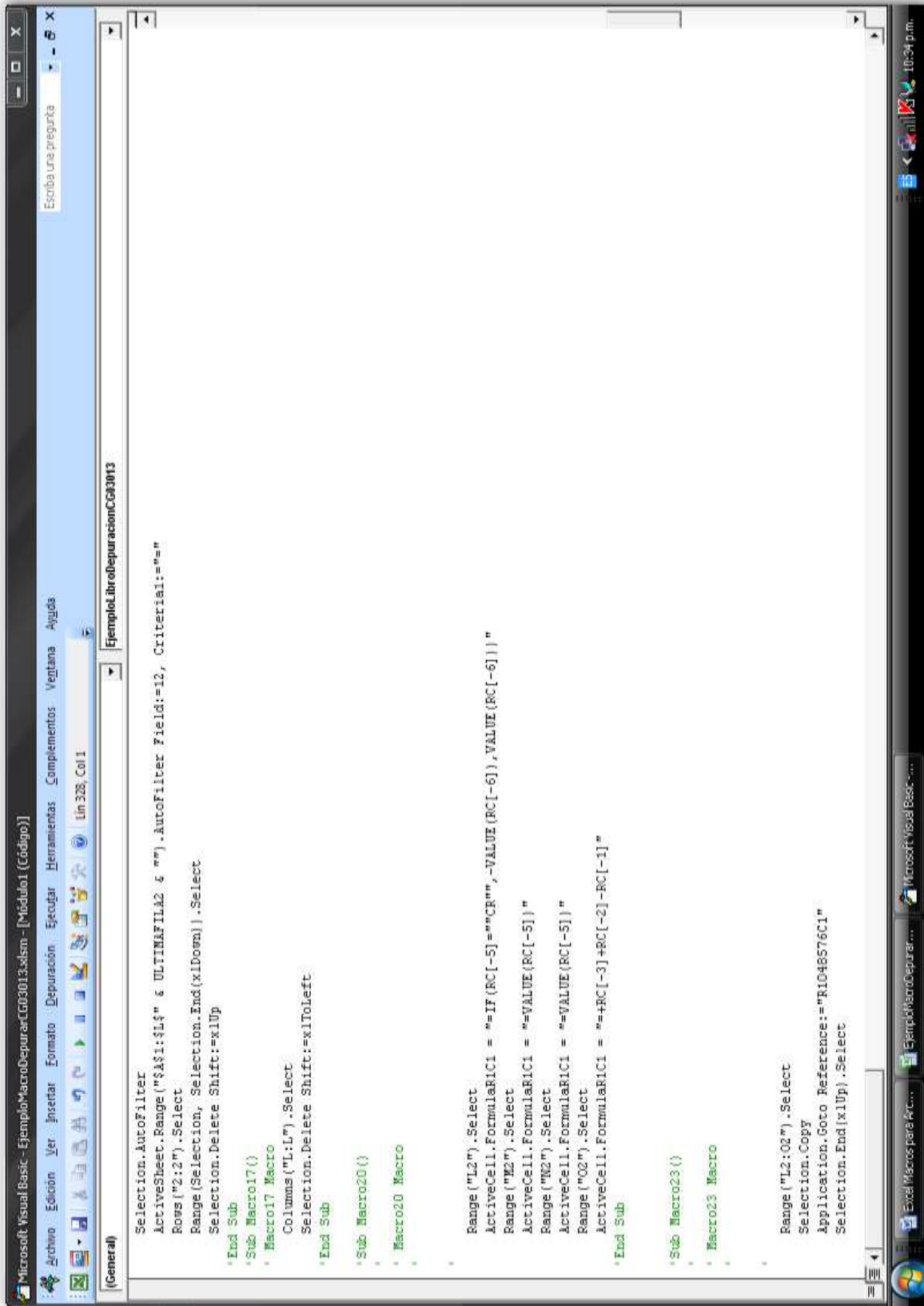


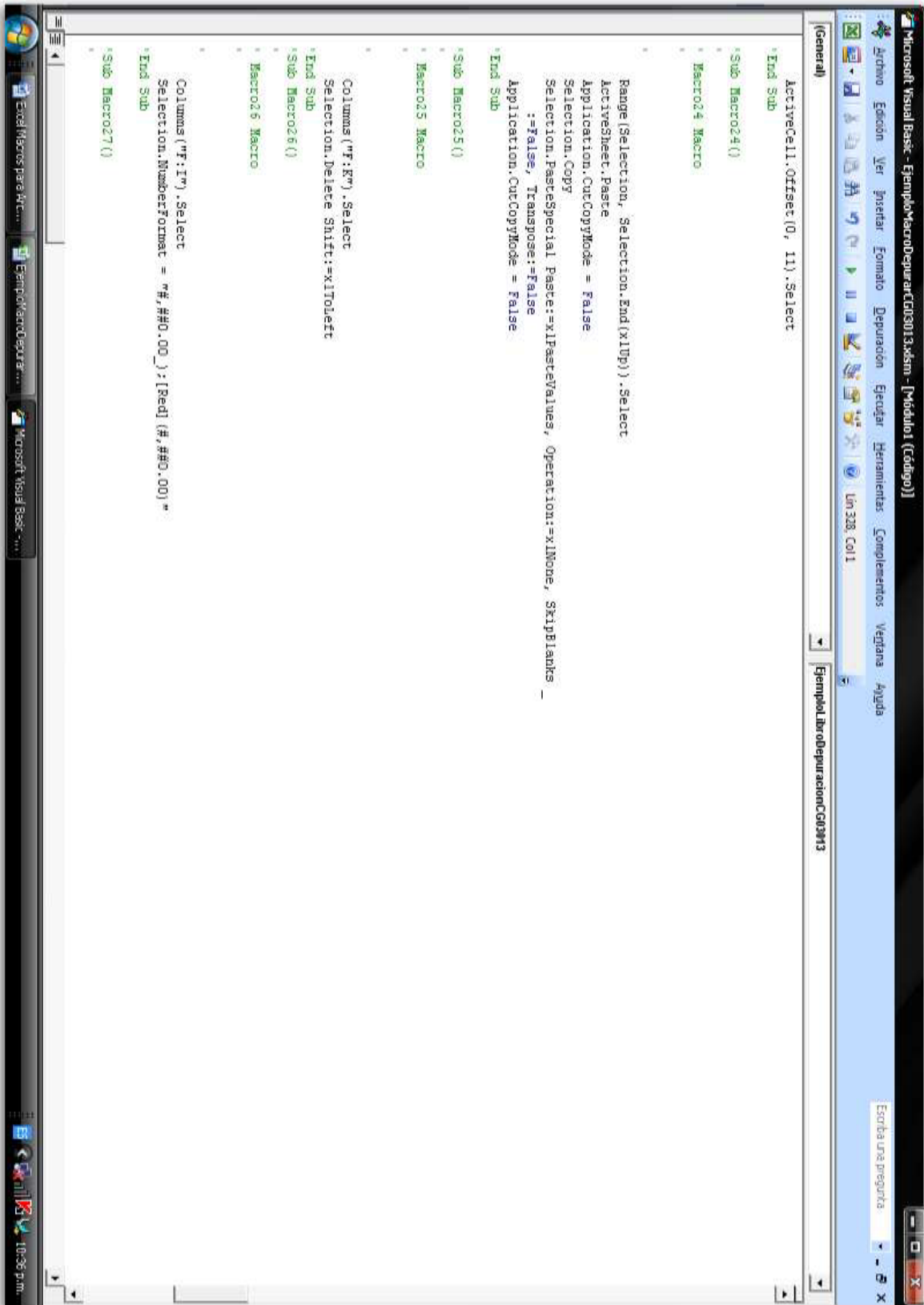
```

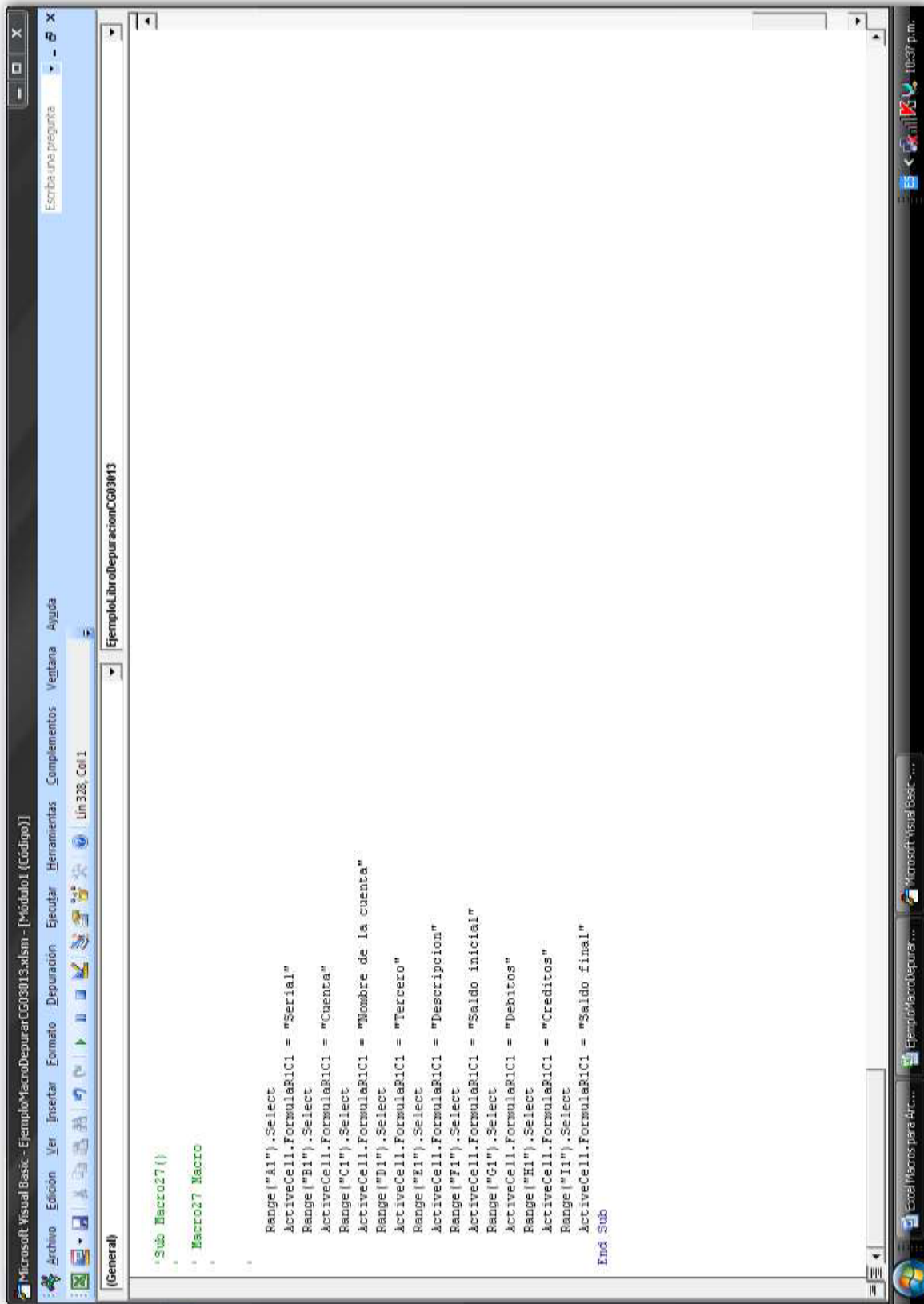
Microsoft Visual Basic - EjemploMacroDepurarCG03013.xlsm - [Módulo1 (Código)]
Archivo Edición Ver Insertar Formato Depuración Ejecutar Herramientas Complementos Ventana Ayuda
Escriba una pregunta
EjemploLibroDepuracionCG03013
[General]
Range("A1").Select
End Sub
Sub Macro23 ()
ActiveWorkbook.Worksheets("CG03013").Sort.SortFields.Clear
ActiveWorkbook.Worksheets("CG03013").Sort.SortFields.Add Key:=Range(
"J2:J" & ULTIMAFILA2 & ""), SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:=
xlSortNormal
xlSortTextAsNumbers
With ActiveWorkbook.Worksheets("CG03013").Sort
.SetRange Range("A1:J" & ULTIMAFILA2 & "")
.Header = xlYes
.MatchCase = False
.Orientation = xlTopToBottom
.SortMethod = xlPinYin
.Apply
End With
Columns("B:C").Select
Selection.Insert Shift:=xlToRight, CopyOrigin:=xlFormatFromLeftOrAbove
Range("F2").Select
Selection.End(xlDown).Select
Range(ActiveCell, ActiveCell.Offset(-1, -2)).Select
Range(Selection, Selection.End(xlUp)).Select
Selection.Cut
Range("E1").Select
ActiveSheet.Paste
Range("D1").Select
ActiveCell.FormulaR1C1 = "4353"
Range("E1").Select
ActiveCell.FormulaR1C1 = "43534535t3"
ActiveWorkbook.Worksheets("CG03013").Sort.SortFields.Clear
ActiveWorkbook.Worksheets("CG03013").Sort.SortFields.Add Key:=Range(
"A2:A" & ULTIMAFILA2 & ""), SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:=
xlSortNormal
With ActiveWorkbook.Worksheets("CG03013").Sort
.SetRange Range("A1:I" & ULTIMAFILA2 & "")
.Header = xlYes
.MatchCase = False
.Orientation = xlTopToBottom

```









```
Sub Macro27()  
    Range("A1").Select  
    ActiveCell.FormulaR1C1 = "Serial"  
    Range("B1").Select  
    ActiveCell.FormulaR1C1 = "Cuenta"  
    Range("C1").Select  
    ActiveCell.FormulaR1C1 = "Nombre de la cuenta"  
    Range("D1").Select  
    ActiveCell.FormulaR1C1 = "Tercero"  
    Range("E1").Select  
    ActiveCell.FormulaR1C1 = "Descripcion"  
    Range("F1").Select  
    ActiveCell.FormulaR1C1 = "Saldo inicial"  
    Range("G1").Select  
    ActiveCell.FormulaR1C1 = "Debitos"  
    Range("H1").Select  
    ActiveCell.FormulaR1C1 = "Creditos"  
    Range("I1").Select  
    ActiveCell.FormulaR1C1 = "Saldo final"  
End Sub
```

A continuación, el código que acabamos de ver, se depura un poco:

Sub EjemploLibroDepuracionCG03013Depurada()

Excel aplicado a la Auditoría, la Contaduría y la Administración

```
Application.CutCopyMode = False
Selection.End(xlUp).Select
ULTIMAFILA2 = Cells(Application.Rows.Count, 2).End(xlUp).Row
Columns("A:A").Insert Shift:=xlToRight, CopyOrigin:=xlFormatFromLeftOrAbove
Range("A1") = "Serial"
Range("A2") = 1
Range("A3") = 2
Range("A2:A3").Select
Selection.AutoFill Destination:=Range("A2:A" & ULTIMAFILA2 & ""), Type:=xlFillDefault
Range("A2:A" & ULTIMAFILA2 & "").Select
Range("A1").Select
ActiveWorkbook.Worksheets("CG03013").Sort.SortFields.Clear
ActiveWorkbook.Worksheets("CG03013").Sort.SortFields.Add Key:=Range( _
    "J2:J" & ULTIMAFILA2 & ""), SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:= _
    xlSortNormal
    xlSortTextAsNumbers
With ActiveWorkbook.Worksheets("CG03013").Sort
    .SetRange Range("A1:J" & ULTIMAFILA2 & "")
    .Header = xlYes
    .MatchCase = False
    .Orientation = xlTopToBottom
    .SortMethod = xlPinYin
    .Apply
End With
Columns("B:C").Insert Shift:=xlToRight, CopyOrigin:=xlFormatFromLeftOrAbove
Range("F2").End(xlDown).Offset(-1, -2).Select
Range(ActiveCell, ActiveCell.Offset(0, 1)).Select
Range(Selection, Selection.End(xlUp)).Select
Selection.Cut
Range("B1").Select
ActiveSheet.Paste
Range("D1") = "4353"
Range("E1") = "43534535t3"
ActiveWorkbook.Worksheets("CG03013").Sort.SortFields.Clear
ActiveWorkbook.Worksheets("CG03013").Sort.SortFields.Add Key:=Range( _
    "A2:A" & ULTIMAFILA2 & ""), SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:= _
    xlSortNormal
With ActiveWorkbook.Worksheets("CG03013").Sort
    .SetRange Range("A1:L" & ULTIMAFILA2 & "")
    .Header = xlYes
    .MatchCase = False
    .Orientation = xlTopToBottom
    .SortMethod = xlPinYin
    .Apply
End With
Range("A1").End(xlDown).Select
Range("C" & ULTIMAFILA2 & "").Select
Range(Selection, Cells(1)).Select
Selection.SpecialCells(xlCellTypeBlanks).Select
Selection.FormulaR1C1 = "=+R[-1]C"
Columns("B:C").Select
Selection.Copy
Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
    :=False, Transpose:=False
Application.CutCopyMode = False
Range("A1").Select
Selection.AutoFilter
ActiveSheet.Range("$A$1:$L$" & ULTIMAFILA2 & "").AutoFilter Field:=12, Criteria1:=""
Range(Rows("2:2"), Selection.End(xlDown)).Select
Selection.Delete Shift:=xlUp
Columns("L:L").Delete Shift:=xlToLeft
Range("L2").Select
ActiveCell.FormulaR1C1 = "=IF(RC[-5]=""CR"",-VALUE(RC[-6]),VALUE(RC[-6]))"
Range("M2").Select
ActiveCell.FormulaR1C1 = "=VALUE(RC[-5])"
Range("N2").Select
ActiveCell.FormulaR1C1 = "=VALUE(RC[-5])"
Range("O2").Select
ActiveCell.FormulaR1C1 = "=+RC[-3]+RC[-2]-RC[-1]"
Range("L2:O2").Select
Selection.Copy
Application.Goto Reference:="R1048576C1"
Selection.End(xlUp).Select
ActiveCell.Offset(0, 11).Select
Range(Selection, Selection.End(xlUp)).Select
ActiveSheet.Paste
```

```
Application.CutCopyMode = False
Selection.Copy
Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
:=False, Transpose:=False
Application.CutCopyMode = False
Columns("F:K").Select
Selection.Delete Shift:=xlToLeft
Columns("F:I").Select
Selection.NumberFormat = "#,##0.00_);[Red](#,##0.00)"
Range("A1") = "Serial"
Range("B1") = "Cuenta"
Range("C1") = "Nombre de la cuenta"
Range("D1") = "Tercero"
Range("E1") = "Descripcion"
Range("F1") = "Saldo inicial"
Range("G1") = "Debitos"
Range("H1") = "Creditos"
Range("I1") = "Saldo final"
```

End Sub

Para finalizar:

Puede ver que todo esto se ha construido básicamente usando el grabador de macros. Usamos la técnica de definir la ultima variable un par de veces al definir la última fila, y uno que otro `activecell.offset`, un par de claves que una vez conocidas son muy sencillas. No hubo necesidad ni siquiera, de usar algún procedimiento de recorrido como `For next` o `Do until`.

Espero que todo lo que hay aquí le sea de utilidad, puede escribirme a mi email, **alejoquiceno@gmail.com**

Éxitos y bendiciones

Alejandro Quiceno García

Para saber más:

Pagina web de Chip Pearson
Pagina web de Ozgrid.Com
Pagina web de Mr Excel.Com
Pagina web de David McRitchie

Archivo de Internet "xlfdic.xls" Diccionario de Formulas de Peter Noneley.

Referencias de estos sitios webs fueron traducidas al español por el Autor, adaptadas y enriquecidas con las situaciones del día a día contable.

Agradecimientos

La dificultad más grande que tuve que superar para sacar este libro adelante fue el condicionamiento que el medio ejerce sobre uno. Es terrible como el prejuicio hacia el no, hacia la negación reina casi en toda parte. Me parece tenaz como en cada una de las fases, mucha gente que uno considera y la tiene en su opinión como muy importante en orden de estimación, son a veces las primeras en desanimarte, personas que en lugar de allanar el camino, pretenden mostrarte todas las dificultades, sin proporcionar valor agregado. Hubo de todo: pensamientos malevos que atacan tu ego y tu capacidad, hasta ideas tales como que el mundo se acaba el 23 de diciembre de 2012.

Este libro fue sacado contra viento y marea, cuando llegué al momento en el cual había terminado el manuscrito necesitaba un o una mecenas. Alguien que creyera en mí, pero que no me viera con ánimo de lucro, solo que me extendiera una mano. Cosa difícil hoy en día, pues es tremenda la cultura del "**CVY**": "Como voy yo". Y no es un "como voy yo" de una simple comisioncita, es una porción grande que quieren usufructuar de ti. Solo la amistad podía suplir mi necesidad, y así fue.

Tengo que agradecer a **Nelsy Gomez Solarte**, quien desde el momento que le conté que ya tenía el libro listo, ha creído en este proyecto a ojo cerrado. Casi no tener amigos es muy triste, pero tener amigos que de verdad tienen la capacidad de extender una mano y ayudar, como dice la propaganda de Máster Card, no tiene precio. La lucha de los emprendedores solo se ve iluminada, irrigada por el poder de la amistad y la solidaridad, valores intangibles que en un panorama económico poco se estiman y sin los cuales, quien sabe como estaría la economía en realidad.