

LECCIÓN 4. AUTOMATIZANDO APLICACIONES EN EXCEL

A pesar de que los ejemplos desarrollados en el curso, facilitan muchísimo los procedimientos administrativos, financieros, contables, etc., Excel posee herramientas más potentes que permiten agilizar muchos eventos de manera automática. Por ejemplo, en la aplicación Factura de la Lección 2, la función *BUSCARV* buscaba un NIT de una tabla de *Cientes*, especificando un rango de celdas para el proceso. Pero ¿qué pasa si la tabla de clientes y productos recibe nuevos registros? Tenemos que cambiar los rangos de búsqueda en cada fórmula, proceso que es dispendioso y se debe hacer manualmente. Claramente en una empresa relativamente mediana y aún pequeña, esto debe tener una solución. Es decir que cada vez que se actualice una tabla con nuevos registros, el rango de búsqueda se debe ajustar automáticamente.

Esta es la intención y el objetivo de la Lección 4. Para estimular la consulta a través de las redes, los invito a que realicen la consulta sobre las funciones *DESREF*, *CONTAR* y *CONTARA*, buscando ejemplos para que nos vayamos acostumbrando a conocer la variedad de páginas que ofrecen servicios completamente gratuitos, los cuales, recomiendo los aprovechemos al máximo. En mi caso, espero que esta afirmación no suene presuntuosa, todo lo que he logrado avanzar en Excel y Access lo he realizado de esta manera, porque no he asistido a ningún curso en la universidad sobre estos temas. Este sistema de autoaprendizaje es el que se está imponiendo con la pandemia. Estoy presto, también, a conocer sus opiniones sobre las bondades del curso y obviamente a que nos subamos al mismo coche para aprender todos.

1. EL PROCESO DE FACTURACIÓN EN UNA EMPRESA: FASE DE DISEÑO

Retomamos el ejemplo de la Ilustración 9 de la Lección 2 (recomendable tener la Ilustración 9 de la Lección 2 en su escritorio). Cargamos Excel y guardamos el libro en blanco como *Leccion4_11*. Vamos a realizar los cambios siguientes, tomando como referente la Ilustración 9 de la Lección 2, para que esta aplicación se convierta en la herramienta de facturación de una empresa o negocio.

The screenshot shows an Excel spreadsheet with the following data:

CÓDIGO	DESCRIPCIÓN	CANTIDAD	VALOR UNIT.	TOTAL
102023	LAPICEROS PAPER MATE	1	4.500	4.500
102021	RESMAS DE PAPEL CARTA	2	18.000	36.000
102027	MARCADORES SHARPIE ROJO	3	3.700	11.100
102030	MEMORIAS USB 100 GB	4	67.000	268.000
VALOR TOTAL FACTURA EN PESOS				319.600

The invoice form data is as follows:

FACTURA No.	78500
NIT	1006
EMPRESA	EL DÓLAR
MUNICIPIO	GUADALUPE

The formula in cell B9 is: `=SI(B8="";"";BUSCARV(B8;DESREF(Cientes!A1;0;0;CONTARA(Cientes!A:A);3);2;FALSO))`

Ilustración 1

- A. Cambiamos los nombres de las Hojas así: Hoja2 = *Cientes*, Hoja3 = *Productos* y habilitamos como hoja adicional la *Hoja4*. Ver Ilustración 1 y 2. Permanecemos en este Libro.
- B. En la *Hoja1*, digitamos el diseño de factura estudiado en la Lección 2, incluyendo los títulos, nombre de la papelería y NIT. No olvidar los formatos de fecha, número, ancho de celdas, etc. Ver Ilustración 1. No vamos a digitar valores, por ahora.
- C. Hoja *Cientes*: digitamos los valores y campos mostrados en la Ilustración 2 (tabla izquierda).

	A	B	C
1	NIT	EMPRESA	MUNICIPIO
2	1000	ABEJITA	AGUADA
3	1001	ALBORADA	CERRITO
4	1002	EL OSITO	CHARALÁ
5	1003	BONAIRE	OCAMONTE
6	1004	MAGICA	OIBA
7	1005	PARK	GUACA
8	1006	EL DÓLAR	GUADALUPE
9			

	A	B	C
1	CODIGO	PRODUCTO	VALOR UNIT.
2	102021	RESMAS DE PAPEL CARTA	18.000
3	102022	RESMAS DE PAPEL OFICIO	22.000
4	102023	LAPICEROS PAPER MATE	4.500
5	102024	LÁPICES MONGOL	950
6	102025	CARTULINA BLANCA PLIEGOS	1.200
7	102026	COLBÓN FCO DE 25 ML	2.500
8	102027	MARCADORES SHARPIE ROJO	3.700
9	102028	AGENDA NORMA CARTA	12.000
10	102029	CUADERNOS 100 HOJAS	2.500
11	102030	MEMORIAS USB 100 GB	67.000
12			

Ilustración 2

- D. Hoja *Productos*. Digitamos los valores y campos mostrados en la Ilustración 2 (tabla derecha).

	A	B	C	D	E	F	G	H	I	J	K	
1	PAPELERÍA EL CENTAVO MENOS					PAPELERÍA EL CENTAVO MENOS						
2	NIT: 2 098 185					NIT: 2 098 185						
3												
4												
5	FACTURA No.	78500				FACTURA No.	78500					
6												
7												
8	NIT	1006				NIT	1006					
9	EMPRESA	EL DÓLAR				EMPRESA	EL DÓLAR					
10	MUNICIPIO	GUADALUPE				MUNICIPIO	GUADALUPE					
11												
12												
13	CÓDIGO	DESCRIPCIÓN	CANTIDAD	VALOR UNIT.	TOTAL	CÓDIGO	DESCRIPCIÓN	CANTIDAD	VALOR UNIT.	TOTAL		
14	102023	LAPICEROS PAPER MATE	1	4.500	4.500	102023	LAPICEROS PAPER MATE	1	4.500	4.500		
15	102021	RESMAS DE PAPEL CARTA	2	18.000	36.000	102021	RESMAS DE PAPEL CARTA	2	18.000	36.000		
16	102027	MARCADORES SHARPIE ROJO	3	3.700	11.100	102027	MARCADORES SHARPIE ROJO	3	3.700	11.100		
17	102030	MEMORIAS USB 100 GB	4	67.000	268.000	102030	MEMORIAS USB 100 GB	4	67.000	268.000		
18	0		0			0		0				
19	0		0			0		0				
20	0		0			0		0				
21	0		0			0		0				
22	0		0			0		0				
23	0		0			0		0				
24	VALOR TOTAL FACTURA EN PESOS				319.600	VALOR TOTAL FACTURA EN PESOS				319.600		
25												

Ilustración 3

- E. La *Hoja4* va a recibir el rango E1:124 de la tabla creada en Hoja1. Para ello puede copiar el rango (*Hoja1*) y pegarlo en la misma posición de la *Hoja4*. Duplicamos este rango con el fin de utilizarlo para la impresión de la factura y

que permita copia para el cliente y copia para el negocio o empresa. Ver Ilustración 3. Ignoremos por el momento los valores que aparecen, dejemos solamente los títulos y nos concentramos solamente en el diseño de la tabla.

Tenemos listo el diseño de la aplicación. No olvide guardarlo. Ahora, viene la codificación.

2. EL PROCESO DE FACTURACIÓN EN UNA EMPRESA: FASE DE CODIFICACIÓN.

- A. HOJA 1. Recordemos que al escribir el NIT de la empresa en F8, la función *BUSCARV* encontraba el nombre y municipio en una tabla de *Cientes* que se encontraba en la Hoja1, rango A3:C9 del Libro estudiado en la Lección 2. Recordemos la fórmula:

Ilustración 4 muestra una barra de fórmulas con la siguiente fórmula: `=SI(F8="";"";BUSCARV(F8;A3:C9;2;FALSO))`. Se ven líneas azules y rojas que conectan los argumentos de la fórmula con referencias de celdas y rangos.

Ilustración 4

- Ahora, el rango a buscar se encuentra en la hoja *Cientes*, rango A1:C8. F8 de la Ilustración 4 cambia a B8. ¿Cómo resolvemos el problema del Rango A1:C8 cuando se ingresa un cliente nuevo en la fila 9? Para ello vamos a utilizar las funciones *DESREF* y *CONTARA*. La consulta en Internet se hace para rangos variables y de los ejemplos investigados, se encontró, a mi parecer la mejor respuesta en la Ilustración 5¹. Digitamos la fórmula, para B9, en la barra de fórmulas como se observa en la Ilustración 1. En un apéndice, al final de la lección se describen los fundamentos teóricos de estas funciones.

Ilustración 5 muestra una barra de fórmulas con la siguiente fórmula: `=SI(B8="";"";BUSCARV(B8;DESREF(Cientes!A1;0;0;CONTARA(Cientes!A:A);3);2;FALSO))`. Se ven líneas azules, rojas y verdes que conectan los argumentos de la fórmula con referencias de celdas y rangos.

Ilustración 5

- Para B10 la fórmula es la misma con un cambio pequeño. El valor 2 del parámetro anterior a FALSO cambia por 3. ¿Recuerda por qué? En la hoja *Cientes*, MUNICIPIO se encuentra en la columna 3 y EMPRESA en la columna 2.
- Celda B14: Descripción o nombre del Producto.** Por analogía con la fórmula de la Ilustración 5, podemos inferir la fórmula, la búsqueda se realiza en la hoja *Productos*, como se puede observar a continuación. La descripción del producto se encuentra en la columna 2 (ver parámetro antes de FALSO).

`=SI(A14="";"";BUSCARV(A14;DESREF(Productos!A1;0;0;CONTARA(Productos!A:A);3);2;FALSO))`

- Escribimos esta fórmula en B14 y la copiamos en el rango B15:B23.
- Celda D14: Valor Unitario del Producto.** Copiamos la fórmula que está en B14. El valor unitario del producto se encuentra en la columna 3 (cambiar parámetro antes de FALSO a 3).
- En el rango D15:D23 procedemos a copiar esta fórmula.
- Celda E14: Valor total del Producto.** Ver apartado 7, Lección 2, página 22. Allí se explicó el algoritmo con su respectiva fórmula. $E14 = C14 * D14$. Escribimos esta fórmula y la copiamos en el resto del rango.
- Celda E24. Valor total de la factura.** Sumamos los valores parciales del rango E14:E23. $E24 = SUMA(E14:E23)$.

¹ El estudiante usuario puede realizar la consulta y verificar en la Ilustración 5 la justificación de los parámetros utilizados.

3. EL PROCESO DE CODIFICACIÓN DE LA HOJA 4: FACTURA PARA IMPRIMIR.

Vamos a copiar los datos (aunque no tenga valores) de la Hoja 1 en la hoja 4. Ver Ilustración 3. El rango A1:E24 de las dos hojas es el mismo (los campos tienen el mismo ancho y las celdas el mismo formato). El procedimiento lo realizaremos en dos etapas:

A. Rango A1:E24 de *Hoja1* a Rango A1:E24 de *Hoja4*.

- 1) Seleccionamos *Hoja 4*.
- 2) Seleccionamos Celda A1 (donde aparece el nombre de la empresa).
- 3) Barra de fórmulas escribimos =
- 4) Seleccionamos *Hoja1*
- 5) Clic en A1
- 6) Clic en el chulito a la izquierda de *fx*
- 7) Debe aparecer en *Hoja4* la fórmula =*Hoja1!A1*

Este procedimiento lo repetimos para los rangos que contienen valores en *Hoja4*: A2, B5, B8, B9, B10 y E24. Para A14 realice el procedimiento y luego copie esta fórmula en A15:A23. Proceda de la misma forma con B14, C14, D14 y E14. Hecho esto, los valores de la *Hoja1* se copian automáticamente en el rango A1:E24 de la *Hoja 4*.

B. Estando en la *Hoja4* copiar los valores del Rango A1:E24 en G1:K24.

- 1) Seleccionamos celda G1.
- 2) En barra de fórmulas escribimos =A1

Repetimos el procedimiento para el resto de celdas. Los títulos de las celdas no los vamos a intervenir con fórmulas.

Hasta aquí tenemos la factura de venta diseñada y codificada. Si se agregan más clientes y productos las fórmulas se actualizan automáticamente. Pero nos quedan pendientes dos eventos: (1) Pulsar un Botón (lo vamos a llamar **Borrar Factura**) para borrar la factura, cuando se tiene un cliente nuevo y que el número de la factura se incremente en una unidad. (2) Pulsar un segundo botón que lo vamos a llamar **Imprimir Factura**. Previamente se tiene la impresora encendida (ON LINE) con una hoja tamaño carta y al pulsar el botón se imprime automáticamente el comprobante de pago.

Para crear los botones que estamos mencionando, nos adentramos en un proceso de automatización más elaborado, conocido como CREACIÓN DE LIBROS EXCEL HABILITADOS PARA MACROS.

4. ¿QUÉ ES UNA MACRO DE EXCEL?²

Si utilizamos Excel frecuentemente es posible que en alguna ocasión nos hayamos encontrado ejecutando una misma serie de acciones una y otra vez. Esas acciones que hacemos repetidas veces se podrían automatizar con una macro.

Una macro es un conjunto de comandos que se almacenan en un lugar especial de Excel de manera que están siempre disponibles cuando los necesitamos ejecutar. Igualmente, el código lo podemos copiar y luego pegar en Word, para poder utilizarlo en otras aplicaciones.

Por ejemplo, si todas las mañanas se crea un reporte de ventas y en ese reporte siempre se da el mismo formato a los textos, se podría crear una macro para que lo haga automáticamente por nosotros. Las macros se utilizan principalmente para eliminar la necesidad de repetir los pasos de aquellas tareas que realizamos una y otra vez.

A partir de Excel 2007, Microsoft trajo un nuevo formato de archivos de Excel, *.xlsx*, que contiene todos los datos de la hoja de cálculo, piense *.xlsx* como si fuera un archivo comprimido con varios otros archivos dentro de él. También un nuevo formato de archivo de libros de trabajo con macros habilitados con la extensión *.xlsm*. Si es nuevo en Excel, o nuevo para los nuevos formatos, conocer que datos tiene cada tipo de archivo de datos en él puede ser un poco confuso.

² <https://sites.google.com/site/aprendiendoexcelinteractivo/home/introduccion-a-macros>

Microsoft Excel utiliza el lenguaje de programación Visual Basic para Aplicaciones (VBA), para proporcionar automatización de hojas de cálculo de Excel. Cuando estos programas se unen a una hoja de cálculo en Excel 2007 o posterior, están unidos como un archivo independiente dentro de la estructura del archivo comprimido. Cuando esto ocurre, la extensión se cambia por *.xlsm. Una ventaja de este formato sobre formatos anteriores de Excel VBA es que el código puede ser eliminado sin necesidad de iniciar el archivo.

A. SEGURIDAD DE MACROS

Al abrir un libro habilitado para macros, predeterminadamente, verá una notificación sobre la barra de fórmulas diciendo "**Riesgo de seguridad: Los macros se han deshabilitado**" con un botón marcado como "Opciones". Al pulsar el botón hará que aparezca una ventana de diálogo llamada "Security Alert - Macro", que te informará acerca de la confiabilidad de la fuente de la macro, y le dará la opción de "Ayudar a protegerme de contenido desconocido" o "Habilitar este contenido". Si confía en el origen del libro habilitado para macros, seleccione este último y haz clic en "Aceptar".

B. PESTAÑA DESARROLLADOR o PROGRAMADOR

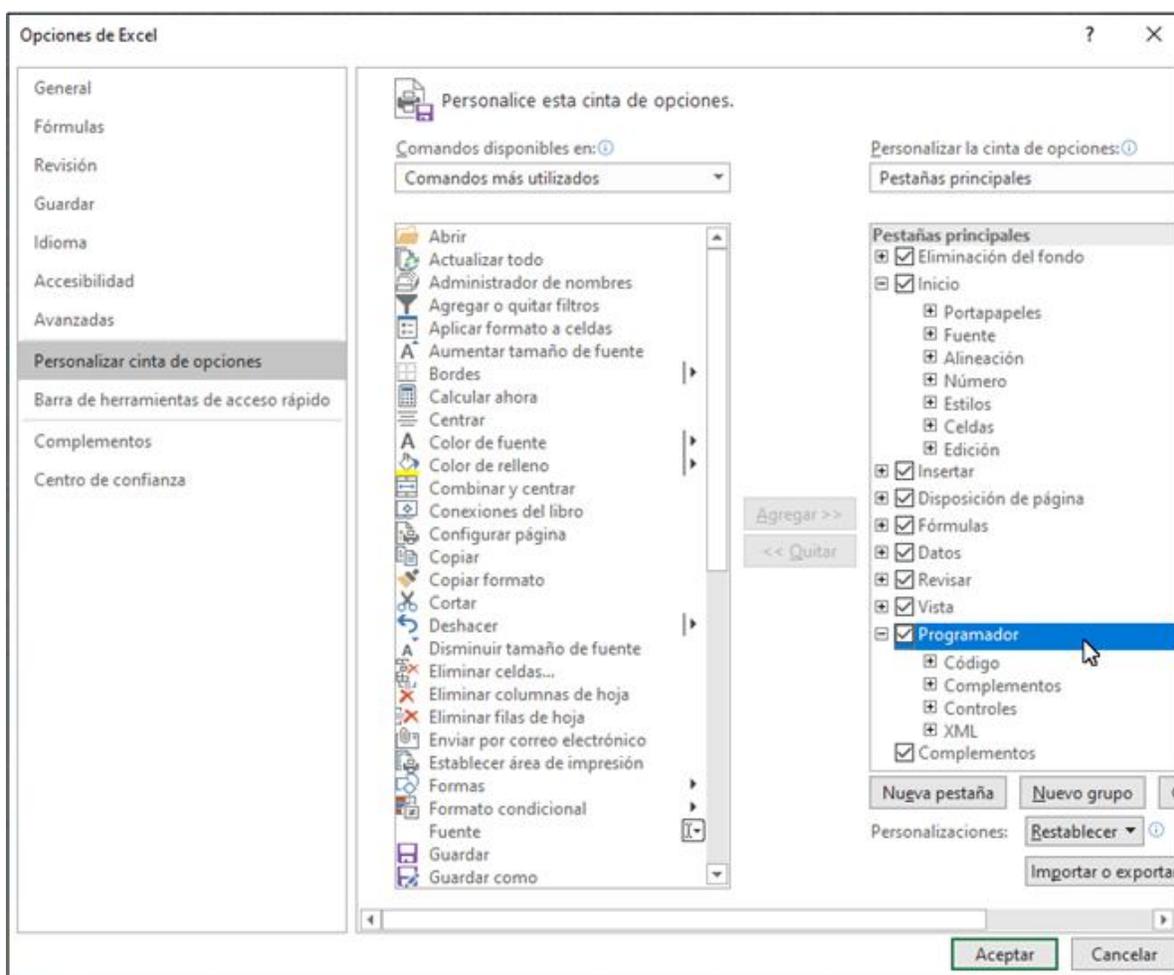


Ilustración 6

La creación de macros en Excel requiere la habilitación de una pestaña nueva en la barra de herramientas. En algunas versiones de Office la pestaña se llama *Desarrollador* y en las últimas cambió a *Programador*. Observe si su versión de Excel tiene esta pestaña que debe aparecer en la barra de herramientas.

Si no la tiene habilitada, proceda de la manera siguiente: *Archivo > Opciones > Personalizar Cinta de Opciones*. Aparece la ventana flotante de la Ilustración 6. En el recuadro titulado *Pestañas principales* hacemos clic en *Programador*. Tal como aparece en la figura nos indica que está habilitada la pestaña *Programador*. Ver Ilustración 3.

C. CAMBIO DE LA CONFIGURACIÓN DE SEGURIDAD³

Puede cambiar el comportamiento de seguridad de macros de Excel haciendo clic en la pestaña "*Desarrollador*" o "*Programador*" de Excel, y hacer clic en el grupo "*Código*". Habrá una opción de "*Seguridad de macros*" en el grupo. Sus opciones para "*Seguridad de macros*" incluyen deshabilitar todas las macros, sin notificación, esto es útil para garantizar que los macros nunca se ejecuten, el valor predeterminado para desactivar la notificación de macros, como se describió anteriormente, habilitar las macros firmadas digitalmente para ejecutarse automáticamente y habilitar todas las macros, que no es recomendable.

5. ¿CÓMO PROTEGER EN EXCEL CELDAS O RANGOS QUE TIENEN FÓRMULAS?

Puede ocurrir que seleccionemos una celda que tiene fórmulas (se dice que está *formulada*) y digitemos un valor cualquiera (letra, número, carácter especial). Esta acción borra la fórmula e inutiliza los procedimientos que ya tenemos codificados. Es decir, la aplicación deja de funcionar. Si tenemos registros de las fórmulas, o copias paralelas de los libros podemos recuperarla, pero esto claramente puede volver a repetirse, incluso por parte de otro usuario que acceda al libro para trabajar.

Excel tiene una solución y es proteger las celdas o rangos que tienen fórmulas. La protección se realiza, no para que los usuarios puedan ver y copiar las fórmulas en otras aplicaciones, sino para que no cometamos el error de borrarlas accidentalmente. Veamos el procedimiento:

- A. Nos ubicamos en la *Hoja1* del Libro que estamos trabajando. Seleccionar toda la hoja que contiene las celdas con fórmulas. Para ello nos ubicamos en la celda a la izquierda de la letra que marca la columna A y arriba del número que marca la fila 1 (debe observar un triángulo rectángulo en miniatura con un sombreado de color gris).
- B. En la pestaña *Inicio > Formato > Formato de Celdas ... > Proteger*, escogemos el rectángulo de opción *Bloqueada* y hacemos clic para desbloquear (debe desaparecer el chulito).
- C. Si este rectángulo de opción no está disponible puede ser que algunas partes de la hoja ya están bloqueadas. Debe proceder a desproteger la hoja, para lo cual debe seleccionar la pestaña *Revisar > Desproteger Hoja* y escribir el código para desproteger si es el caso.
- D. Ahora, debemos seleccionar las celdas o rangos que queremos bloquear y repetimos el paso B, seleccionando el rectángulo de opción *Bloqueado*. Debe aparecer el chulito dentro del rectángulo. Las celdas que contienen fórmulas aparecen con un pequeño rectángulo de color verde en la esquina superior izquierda, para no cometer el error de proteger celdas que no contienen fórmulas.
- E. Veamos como seleccionamos las celdas y rangos cuando estos no son adyacentes (ver Ilustración 1):
 - 1) Mantenemos hundida la tecla *Ctrl* y arrastramos el botón izquierdo del Mouse sin soltar en el Rango B9:B10. Soltamos el botón del Mouse.
 - 2) Repetimos el paso 1) para el rango B14:B23.

³ Seguir el paso A es suficiente. Cambiar la configuración de seguridad se recomienda para usuarios de Excel avanzados.

- 3) Repetimos el paso 1) para el rango D14:D23.
 - 4) Repetimos el paso 1) para el rango E14:E23.
 - 5) Repetimos el paso 1) para la celda E24.
- F. Ahora debemos escribir el código de la contraseña que va a evitar que otro usuario desproteja las celdas con fórmulas. Pestaña *Revisar* > *Proteger Hoja* > *Contraseña para desproteger la hoja* > “escribimos el código”. Se repite nuevamente cuando Excel lo solicite. Si olvida el código para desproteger una hoja no puede hacer nada. Debe arrancar de cero para codificar nuevamente su proyecto. Escriba el código en un lugar que le permite recordarlo fácilmente. No necesita inventarse un código sofisticado, como si lo hacemos para las claves de las cuentas de correo. En mi caso, casi siempre utilizo el código 12345 que es muy fácil de recordar. Aquí la intención es evitar que por error borremos las fórmulas. Si un intruso pretende conocer las fórmulas es muy fácil decirle: “consulte la lección que allí están las fórmulas”.
- G. Si es necesario acudir a la fórmula para hacerle modificaciones, se debe proceder como hemos explicado en 5C a desproteger la hoja, realice los cambios, guarde el libro y vuelva a proteger la hoja u hojas que hay estado protegidas.

Ensaye escribiendo en una celda que contiene fórmulas ya protegidas, aparece el mensaje siguiente: *“La celda o el gráfico que intenta cambiar están en una hoja protegida. Para realizar un cambio, desproteja la hoja. Es posible que se pida que escriba una contraseña”*.

La *Hoja4* contiene más fórmulas que la *Hoja1*, pero no la vamos a proteger. Por el diseño que tiene la aplicación, el usuario no interviene en las celdas escribiendo valores, porque estos son arrastrados desde la *Hoja1* automáticamente. Si hay alguna intervención es en el formato de las celdas de la hoja, configurar impresión, etc.

6. EL PROCESO DE FACTURACIÓN UTILIZANDO EL LENGUAJE PARA MACROS DE EXCEL: BOTÓN BORRAR FACTURA

Vamos a crear el botón **Borrar Factura** mencionado en los párrafos finales del apartado 3.

PASO 1. CAMBIAR EL FORMATO DEL LIBRO A .XLSM.

Tenemos el libro *Leccion4_11.xlsx* (en mi caso por cuestiones de organización del curso lo llamo *Leccion4A.xlsx*) que va cambiar a *Leccion4_11.xlsm*.

Archivo > *Guardar como* > *Examinar* (escoger la misma carpeta donde se tiene el libro actual). En *Tipo* (ver Ilustración 7) escogemos la opción *Libro de Excel habilitado para macros* y en nombre del archivo, Excel cambia automáticamente la extensión por *xlsm*. Dejamos este nombre y Guardar.

PASO 2. CREACIÓN DE LA MACRO. Primero debemos escribir en un borrador los pasos que vamos a realizar para borrar las celdas de la *Hoja1*. Se supone que borramos las celdas que no tienen fórmulas, porque si intentamos hacerlo en las que están formuladas, Excel no lo permite por estar protegidas. Tampoco vamos a borrar los títulos de las celdas. Veamos este borrador:

- 1) Borrar el rango B8.
- 2) Borrar el rango A14:A23
- 3) Borrar el rango C14:C23
- 4) Seleccionar una celda vacía sin fórmulas, para des seleccionar las celdas. Ejemplo F1.
- 5) Terminar: fin del proceso.

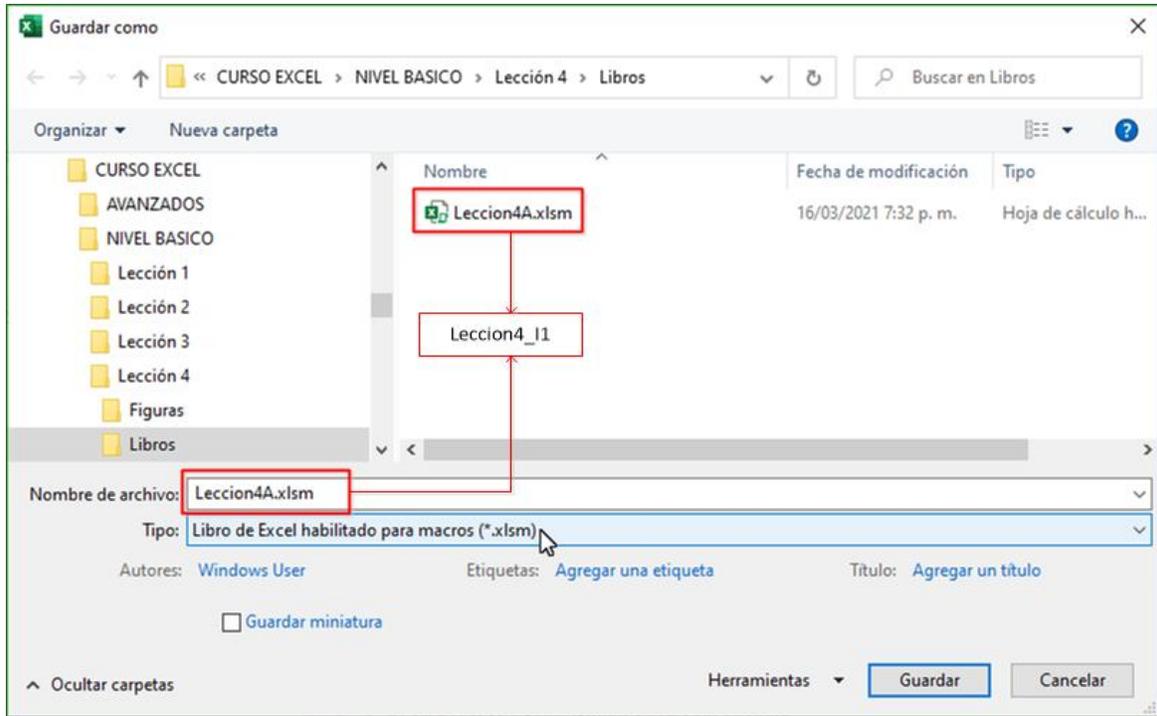


Ilustración 7

La importancia de tener el listado de los pasos a realizar: Excel trabaja las macros como una grabadora (no de voz), en donde todas las pulsaciones de teclas y los movimientos del Mouse quedan registrados, incluidos los tecleos equivocados.

Con el borrador elaborado estamos listos para grabar la macro:

Programador > *Grabar Macro* > aparece una ventana flotante (Ilustración 8). Escribimos el nombre que vamos a dar a la macro: *BorrarFactura* y tecla de método abreviado *Ctrl + b* (puede escoger otra letra). El nombre de la macro debe empezar por un carácter alfanumérico, no puede tener espacios en blanco ni caracteres especiales, puede incluir números si quiere (después de la primera letra).

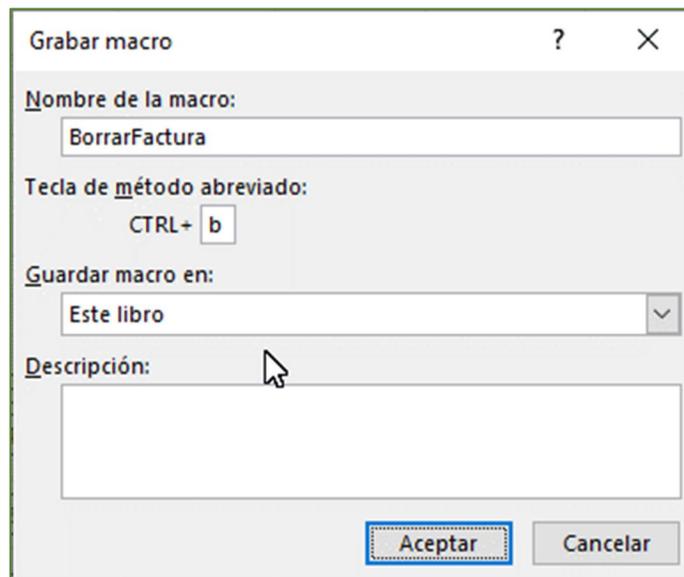


Ilustración 8

Hecho esto pulsamos *Aceptar*. Excel se encuentra esperando que sigamos los pasos del borrador.

- 1) Seleccionar rango B8. Pulsamos la tecla *Supr* o *Del* si el teclado es inglés.
- 2) Seleccionamos el Rango A14:A23 (recuerde que para seleccionar más de una celda utilizamos el Mouse arrastrando con el botón izquierdo pulsado). Pulsar *Supr*.
- 3) Seleccionamos el rango C14:C23. Pulsar *Supr*.
- 4) En ficha *Programador* clic en *Detener Grabación*.

Como decidimos que las teclas de método abreviado para borrar son *Ctrl + b*, puede proceder realizando la pulsación de las dos teclas para verificar lo que estamos buscando: borrar la factura y prepararnos para la factura siguiente. Para la tecla de método abreviado se recomienda escoger pares de telas de poco uso. Por ejemplo: no es recomendable *Ctrl + C* ni *Ctrl + V*, porque son propias para copiar y pegar. Si hay varias macros en la aplicación, puede ser complicado recordar el par de teclas a pulsar, por lo que en este caso lo más práctico es crear un botón de comando que realice las mismas operaciones dentro del procedimiento. Nosotros creamos el botón, allí copiamos el código que nos grabó Excel y si queremos, borramos la macro.

¿Podemos ver la Macro?

Claro que sí. Ficha *Programador* > *Ver Macro* (*Alt + F8*) y aparece la ventana de la Ilustración 9. Veamos:

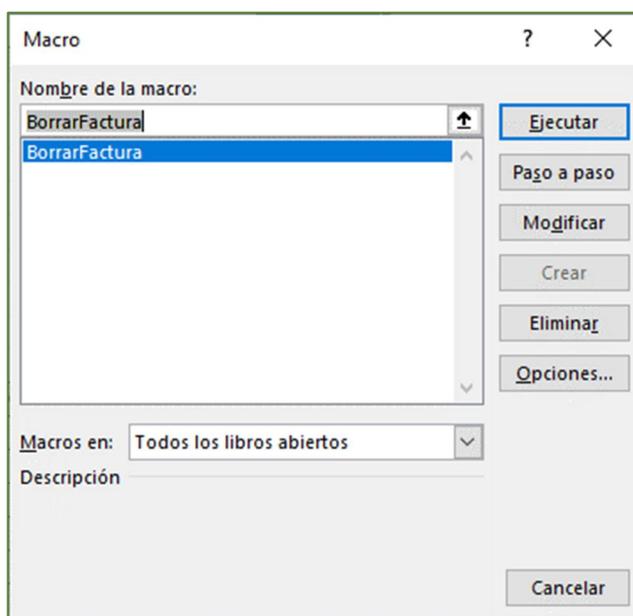


Ilustración 9

Estando en esta ventana podemos ejecutar la macro (la misma acción de *Ctrl + b*), modificarla, eliminarla y tiene otras opciones que por el momento no vamos a explorar. Si pulsamos el botón *Modificar*, aparece el listado de ilustración 10.

El código que podemos observar es propio del lenguaje Visual Basic para Aplicaciones en Excel (VBA). Este lenguaje forma parte de aplicaciones más elaboradas que un usuario experimentado puede diseñar de manera independiente, es decir, creando las macros independientemente de la grabadora de macros, aunque en el fondo esta grabadora sigue interviniendo.

La macro de la ilustración está guardada en el *Módulo 1* (ver panel izquierdo Ilustración 10). Si hacemos clic aquí y luego pulsamos *F7*, se puede ver también el código. Para el botón de comando **Borrar Factura**, el código de la ilustración 10 lo vamos a copiar en la *Hoja1*. Cuando terminemos el proceso, podemos hacer clic en *Hoja1* (ver arriba de *Módulo 1*), luego *F7* y podemos observar también el listado de este código.

El lenguaje VBA consta de instrucciones casi todas en inglés pero fácil de comprender. En la tabla 1 vamos a explicar las acciones que se realizan en algunas de las instrucciones.

Tabla 1

Range ("B8").Select	Seleccione el rango B8
Setection.ClearContents	Borre el contenido del rango seleccionado
Range ("A14:A23").Select	Seleccione el rango A14:A23
Range ("F1").Select	El cursor del Mouse se ubica en esta celda al terminar de borrar. Puede escoger otra.

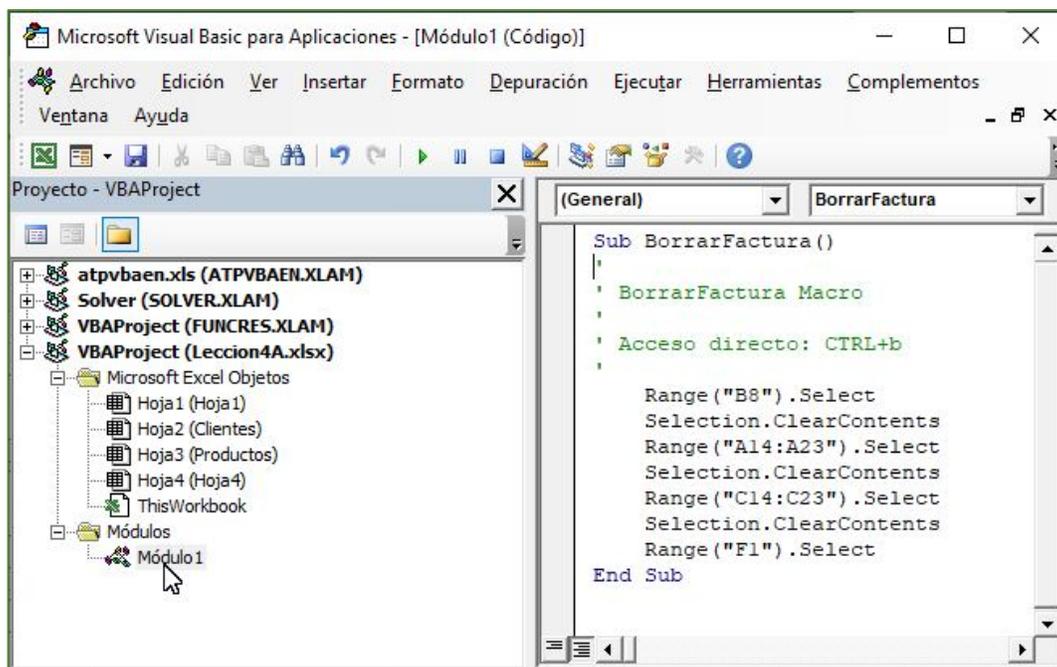


Ilustración 10

La macro de la Ilustración 10 tiene comentarios para explicar el código, pero no son instrucciones ejecutables. Están en color verde. Podemos quitar los que tienen solamente la comilla sencilla, o podemos agregar más, digitando la comilla y luego el comentario.

PASO 3. COPIAR EL CÓDIGO EN WORD (OPCIONAL). Podemos cargar Word, copiar el código (*Ctrl + C*) de la Ilustración 10 y pegarlo en Word (*Ctrl + V*) con los títulos y comentarios que consideremos pertinentes con el fin de utilizarlos en otras aplicaciones. Guardar el Archivo Word con un nombre significativo que nos permita recordar lo que allí tenemos.

PASO 4. CÓDIGO PARA QUE EL NÚMERO DE LA FACTURA SE ACTUALICE CADA VEZ QUE VAMOS A ELABORAR FACTURA NUEVA. En la Ilustración 1 aparece en B5 el valor 78500 como número de la factura actual. Es solamente un ejemplo, pero si es la primera factura, debe tener el valor 1. Queremos que, al borrar la factura actual, el número de la factura se incremente en una unidad para la factura siguiente. Por esta razón, no incluimos esta celda en la macro de *BorrarFactura*.

VBA dispone de los OPERADORES DE ASIGNACIÓN, que permiten asignar un valor a una variable. Se utiliza el símbolo de igualdad (=) con un significado un poco diferente al que conocemos en matemáticas elementales, y el valor depende del tipo de datos (numérico, alfanumérico, Fecha, etc). El formato es: *Variable = Valor*

Para nuestro caso: B5 = 78500. El valor actual de B5 es 78500. Queremos que al borrar las celdas que ya conocemos, la variable B5 se incremente en una unidad. B5 = B5 + 1. Veamos la Ilustración 11.

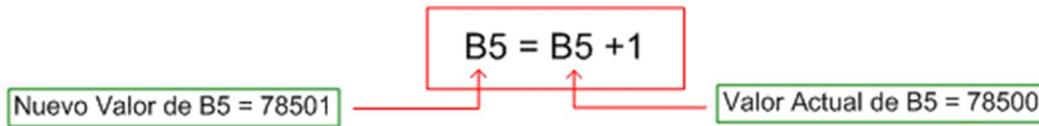


Ilustración 11

Los puristas del álgebra de Baldor me van a perdonar, pero esta no es una ecuación algebraica. El significado es: "al valor actual de la variable B5 agregarle 1 y el resultado guardarlo en B5". Recordemos siempre este operador, se utiliza con mucha frecuencia en código VBA.

La ecuación de la ilustración 11 en código VBA es:

$$\text{Range}("B5") = \text{Range}("B5").\text{Value} + 1 \quad (1)$$

La propiedad **.Value** lo que nos está diciendo es "al valor actual del rango B5 sumarle 1". Antes de escribir el código de la ecuación 1, se debe seleccionar el Rango B5: `Range("B5").Select`.

Procedemos a incluir estas dos líneas de código. Nos debe quedar así:

```
Private Sub cmdBorrarFactura_Click()
' BorrarFactura Macro
' Acceso directo: CTRL+b
Range("B8").Select
Selection.ClearContents
Range("A14:A23").Select
Selection.ClearContents
Range("C14:C23").Select
Selection.ClearContents
Range("B5").Select
Range("B5") = Range("B5").Value + 1
Range("F1").Select
End Sub
```

Guardamos estos cambios haciendo Clic en el Logo de guardar de la barra superior izquierda (una especie de disquete) y si quiere copiar y pegar en Word este nuevo código lo puede hacer en el mismo archivo que se recomendó en párrafos anteriores. Cerramos esta ventana para regresar a Excel, o hacemos clic en el logo de Excel. Nos ubicamos en la *Hoja1* para diseñar el botón de comando y luego escribirle este mismo código.

PASO 5. INSERTAR BOTÓN DE COMANDO EN HOJA 1 PARA BORRAR LA FACTURA CON UN CLIC.

- 1) Ficha Programador
- 2) Insertar

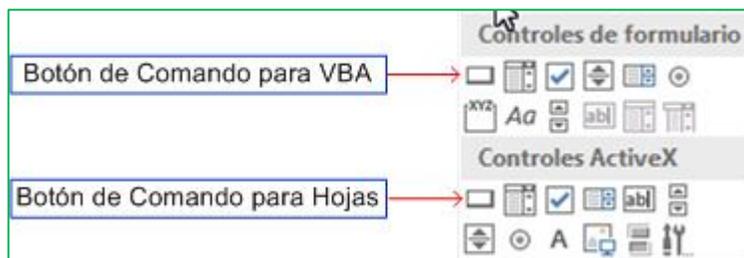


Ilustración 12

- 3) El botón de comando forma parte de un conjunto de controles propios de las aplicaciones que utilizan macros. El botón lo vamos a tener en la Hoja1 de Excel, es un control ActiveX, por lo que hacemos clic en el botón inferior. El cursor del Mouse se convierte en una cruz, que ubicamos en la parte de la factura entre las columnas F-G y filas 1-2 (si vamos a utilizar la aplicación en un portátil, queremos que el botón se puede ver en la pantalla, sin necesidad de utilizar la barra de desplazamiento horizontal). Pulsamos el botón izquierdo del Mouse y lo arrastramos hasta que aparezca un rectángulo con el nombre del botón (generalmente en inglés) al cual le vamos a cambiar algunas de sus propiedades.

CommandButton1

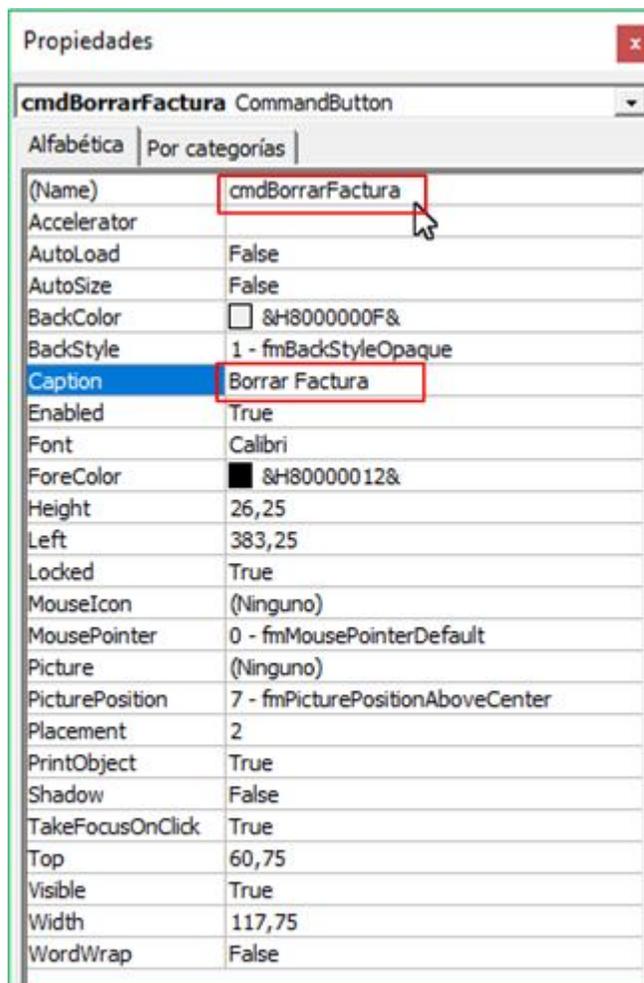


Ilustración 13

Para llegar a la ventana de propiedades del botón debemos realizar el proceso siguiente: Pestaña *Programador* > *Modo diseño* > clic en el botón (aparecen en los bordes unos círculos pequeños, prueba de que está seleccionado) > *Propiedades*. Debe aparecer en la pantalla la ventana de la Ilustración 7. Vamos a cambiar tres propiedades:

- *(Name)*. Esta propiedad, o Nombre en español, es la que se utiliza en el código de macros. El nombre debe empezar por un carácter literal, no puede llevar espacios en blanco ni los caracteres \ / & % \$ ¿ ? ¡ ! = + #. Lo vamos a llamar **cmdBorrarFactura**. Se recomienda que las dos o tres primeras letras sean minúsculas y que traten de identificar el tipo de botón (*cmd* de command).
- *Caption*. O título viene a ser lo que el usuario puede ver en la hoja. No hay restricciones para el título, así que lo vamos a llamar **Borrar Factura**.

- *Back Color*. O color de Fondo. En la segunda columna aparece el código del color actual. Clic en la parte derecha del código y aparece una paleta de colores. Clic en el color que más nos guste.
- Otras Propiedades: tipo y tamaño de fuente, ancho y alto del botón, etc. No las vamos a modificar por ahora. Una consulta en Internet sobre estas propiedades nos puede servir para aplicaciones futuras.

7. EL PROCESO DE FACTURACIÓN UTILIZANDO EL LENGUAJE PARA MACROS DE EXCEL: BOTÓN IMPRIMIR FACTURA

Los pasos para diseñar y codificar este botón son los mismos que para **Borrar Factura**. El código aquí se debe referir a la *Hoja4* que es la que vamos a imprimir.

PASO 1. CONFIGURAR LA PÁGINA DE LA HOJA4.

- Seleccionar *Hoja4*.
- Pestaña *Disposición de Página* > *Configurar página* > clic en la flecha que aparece en la ventana inferior derecha.

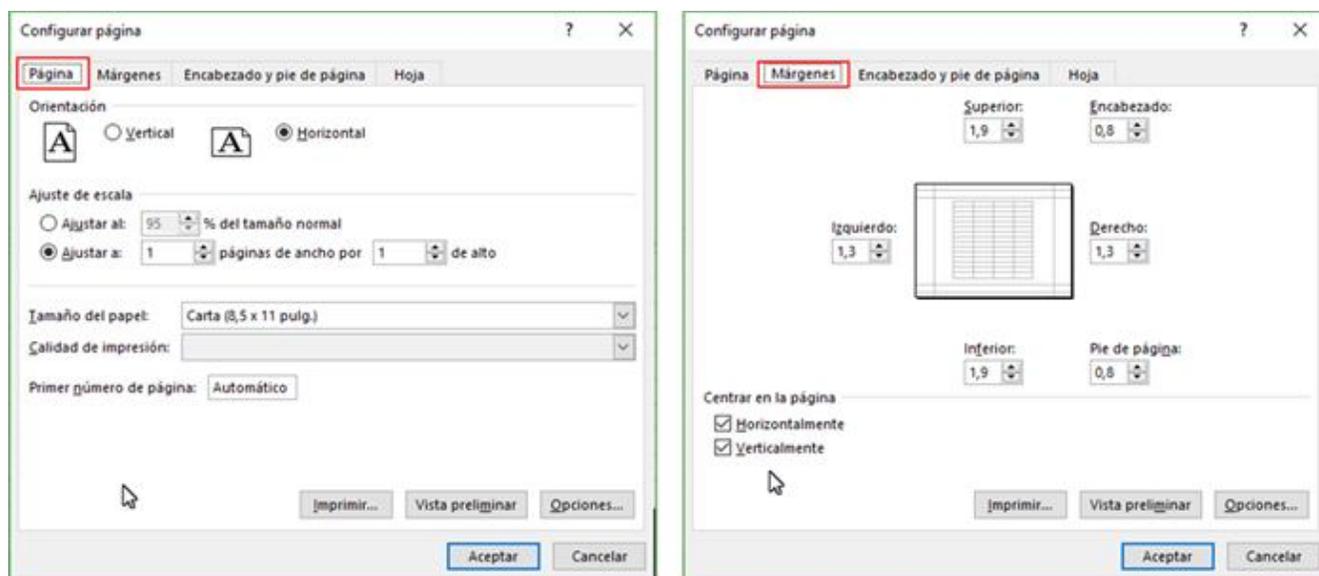


Ilustración 14

- En la ventana flotante (Ilustración 14, izquierda) escogemos: (1) Orientación Horizontal, (2) Ajustar a: 1 página de ancho por 1 de alto. El tamaño del papel debe ya estar configurado a carta, de lo contrario hay que seleccionarlo.
- Clic en la pestaña *Márgenes* (Ilustración 14, derecha). Escogemos la selección mostrada.
- Clic en la pestaña *Encabezado y pie de página* (Ilustración 15, izquierda). Escogemos la selección mostrada.
- Pestaña *Hoja* (Ilustración 15, derecha). Escogemos los valores mostrados. El área de impresión es A1:K24.
- *Vista Preliminar*: podemos hacer clic y observar cómo queda la página para la impresión. Podemos hacer los cambios que consideremos necesarios. Por ejemplo, si quiere que la página tenga un encabezado y pie de página, puede incluir la fecha, el nombre de la hoja, etc.

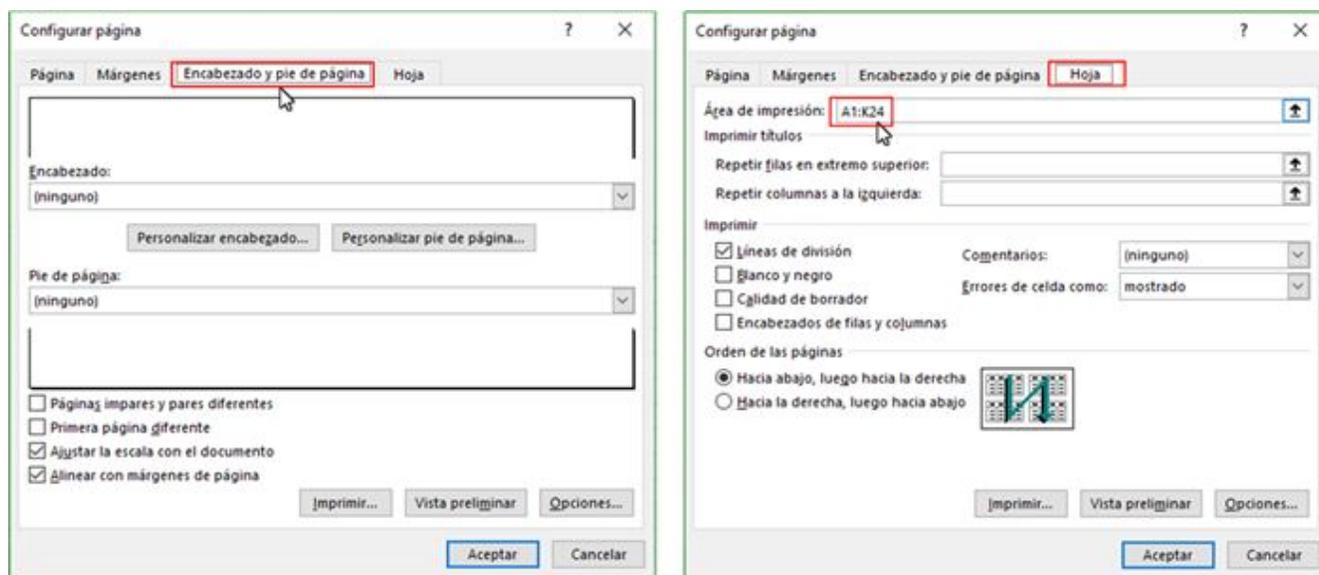


Ilustración 15

PASO 2. CREAR LA MACRO PARA IMPRIMIR LA PÁGINA.

- Borrador con los pasos: (1) Partimos de la Hoja1, porque allí vamos a ubicar el botón, (2) Seleccionar Hoja4, (3) Pestaña Archivo > Imprimir. Seleccione impresora > Imprimir (aunque debe estar ya seleccionada la impresora), coloque papel, impresora ON LINE. Pestaña Programador > Detener Grabación.
- *Programador > Grabar Macro >* aparece ventana flotante (nos vamos a apoyar en la Ilustración 8). Escribimos el nombre que vamos a dar a la macro: *ImprimirFactura* y tecla de método abreviado *Ctrl + p* (puede escoger otra letra), *Aceptar*. Iniciamos con los pasos del Borrador.

Para no ser repetitivos, ya que hemos desarrollado completamente el proceso para borrar la factura, veamos el código que genera la macro. Si prefiere, puede digitar este código en el código del botón **ImprimirFactura** si no quiere repetir el proceso con la macro. Si conoce como convertir de PDF a Word, puede hacer la conversión y cuando tenga el archivo en Word, lo copia de Word y lo pega en Excel. La macro puede estar guardada en un nuevo módulo, por lo que podemos revisar el módulo 1 o 2 para verificar.

```
Sub ImprimirFactura ()
' ImprimirFactura Macro
' Acceso directo: CTRL+p
  Sheets("Hoja4").Select
  ActiveWindow.SelectedSheets.PrintOut Copies:=1, Collate:=True, _
  IgnorePrintAreas:=False
  Application.Goto Reference:="ImprimirFactura"
  Sheets("Hoja1").Select
End Sub
```

El guión al piso (_) que se observa en la instrucción *ActiveWindow.SelectedSheets.PrintOut Copies:=1, Collate:=True, _ IgnorePrintAreas:=False*, se utiliza para pasar a la línea siguiente con la misma instrucción y facilitar la legibilidad. Puede quedar así:

ActiveWindow.SelectedSheets.PrintOut Copies:=1, Collate:=True, IgnorePrintAreas:=False

- Nos ubicamos en la *Hoja1* para crear el botón. Si hemos comprendido el proceso para el botón anterior, lo dibujamos debajo del anterior, buscando que tenga el mismo tamaño y al copiar el código (nuevamente en la Hoja1 porque allí está el botón) copiamos y pegamos el código de la macro, o el que tenemos en el listado entre *Sub ImprimirFactura ()* y *End Sub*. Tenga cuidado de que no queden repetidos los comandos *Sub ImprimirFactura ()* y *End Sub*.

PASO 3. BORRAR LAS MACROS Y DEJAR ÚNICAMENTE EL CÓDIGO DE LOS BOTONES (OPCIONAL).

Si los ensayos realizados con los botones son correctos, podemos borrar las macros que hemos creado. Para ello, nos remitimos a la Ilustración 9, seguimos los pasos para llegar a esta ventana y utilizamos el botón Eliminar. Excel pregunta si quiere conservar las macros, le decimos que no y hemos terminado de codificar la aplicación.

Recomendación:

Se han mostrado detalladamente los pasos, en algunos casos hemos sido repetitivos, pero esta etapa del curso es clave para aprender a programar en código Visual Basic para Aplicaciones. Llegar a este nivel, que podemos calificar intermedio, nos permite crear programas autónomos, intercambiando información entre formularios y las hojas de Excel. En esta etapa, podemos, si es el caso, ocultar Excel y trabajar solamente desde los formularios.

Por lo tanto, la invitación es a repetir esta lección hasta que podamos comprender todas las etapas de la misma. En la Lección anterior se prometió el desarrollo de otro ejemplo (nómina), pero lo dejaremos para lección posterior.

El diseño de una aplicación puede en ocasiones ser más dispendioso que la codificación. No hemos trabajado los adornos que tiene Excel para que el diseño sea más llamativo, con colores y marcos que por ahora, los dejaremos pendientes y a gusto del estudiante si quiere “enlucir” sus aplicaciones.

APÉNDICE 1. FUNCIÓN CONTARA

La función **CONTARA** cuenta la cantidad de celdas que no están vacías en un intervalo.

La sintaxis de la función CONTARA es: *CONTARA(lista-argumento)*

donde *lista-argumento* es un rango de celdas que usted quiere contar.

Para aplicar la función CONTARA:

- seleccione la celda donde usted quiere ver el resultado,
- pulse el icono Insertar función  que se sitúa en la barra de herramientas superior, o haga clic derecho en la celda seleccionada y elija la opción Insertar función en el menú, o pulse el icono  que se sitúa en la barra de fórmulas.
- seleccione el grupo de funciones Estadísticas en la lista,
- pulse la función *CONTARA*,
- introduzca los argumentos correspondientes separados por comas o seleccione un rango de celdas con el ratón,
- pulse el botón *Enter*.
- Observe en el ejemplo (Ilustración 5) como se utilizó para la aplicación.

APÉNDICE 2. FUNCIÓN DESREF

Devuelve una referencia a un rango que es un número de filas y de columnas de una celda o rango de celdas. La referencia devuelta puede ser una celda o un rango de celdas. Puede especificar el número de filas y el número de columnas a devolver.

Fórmula: *DESREF (ref, filas, columnas, [alto], [ancho])*

- **Referencia:** Es un argumento obligatorio. Indica la referencia en la que queremos basar el desplazamiento de las filas y las columnas. Es decir, **indica el punto de partida o el origen del que la función empezará a trabajar.**
- **Filas:** Es un argumento obligatorio. Indica el número de filas (hacia arriba o hacia abajo) que la función DESREF recorrerá desde el punto de inicio o referencia. Si el argumento filas es 5, la celda superior izquierda de la referencia pasa a estar cinco filas más abajo que la referencia. El número del argumento “Filas” **puede ser positivo o negativo.** Debemos usar un **número negativo para indicar filas superiores** a la referencia y un **número positivo para indicar filas inferiores** a la referencia.
- **Columnas:** Es un argumento obligatorio. Indica lo mismo que el argumento de “Filas” pero para las columnas. Si el argumento columnas es 5, la celda superior izquierda de la referencia pasa a estar cinco columnas hacia la derecha de la referencia. Del mismo modo que en “Filas” el argumento “Columnas” **puede ser positivo o negativo.** Debemos usar un **número positivo para indicar filas superiores** y un **número negativo para indicar filas inferiores** a la referencia.
- **Alto:** Es un argumento opcional. Indica cuantas filas de alto queremos que tenga la referencia devuelta. Se indica en número de filas. El alto debe ser un número positivo. No se refiere al alto de la celda.
- **Ancho:** Es un argumento opcional. Indica cuantas columnas de ancho queremos que tenga la referencia devuelta. Se indica en número de columnas. El ancho debe ser un número positivo. No se refiere al ancho de la celda.
- Observe en el ejemplo (Ilustración 5) como se utilizó para la aplicación.

Les recuerdo que el curso con sus respectivas lecciones se está publicando en

<https://erpoquimicainem.jimdofree.com/>