

CONTROLES, MACROS Y FUNCIONES

CONTENIDO

Controles, macros y funciones	1
Contenido	1
Controles y macros.....	2
Ideas previas	2
Grabación de macros	5
Asignación de una macro a un objeto	7
Programación de una macro sencilla.....	8
Uso de botones	14
Un caso práctico	16
Definición de funciones.....	17
Función con varios argumentos	20
Otros controles.....	21
Botones de opción	21
Cuadros de grupo.....	23
Casillas de verificación	24
Macros de apertura.....	25

CONTROLES Y MACROS

IDEAS PREVIAS

¿Qué es una macro?

Una macro es un conjunto de operaciones de Hoja de Cálculo que el programa memoriza para repetir ese conjunto cada vez que se desee.

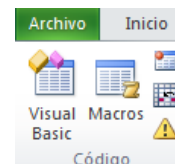
En el uso de macros podemos distinguir tres operaciones:

- Usar macros que han programado otras personas.
- Grabar y ejecutar nuestras propias macros. Asignarlas a objetos.
- Aprender a programar macros mediante el lenguaje Basic de Excel

En esta guía estudiaremos algunos detalles de cada uso

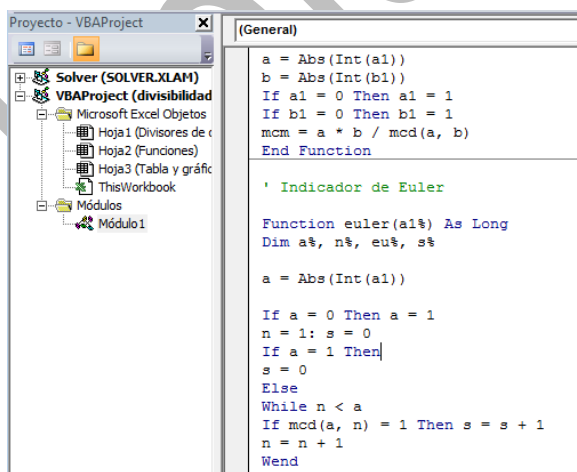
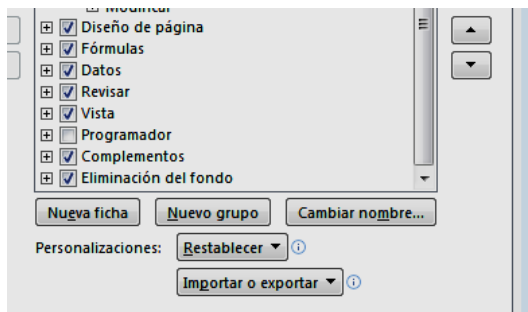
¿Dónde se guardan las macros?

Las macros están alojadas en unos contenedores llamados **módulos**. Pueden pertenecer a las carpetas generales de Excel, en cuyo caso se pueden ejecutar en cualquier hoja de cálculo, o bien estar destinadas sólo a una hoja, y estas serán las que nos interesen en este momento, porque las otras es peligroso tocarlas.



Abre la ficha **Programador** y usa el primer botón de la izquierda **Visual Basic** para ver qué macros tiene definidas tu hoja de cálculo.

Pudiera ser que no tengas a la vista esa ficha **Programador**. En ese caso abre la ficha **Archivo** y en el apartado **Ayuda** elige **Opciones**, y dentro de ellas, **Personalizar cinta de opciones**. Verás que la ficha **Programador** no está activada. Pulsa sobre ella para incorporarla a la cinta general



En la imagen se pueden ver algunas macros de las pertenecientes a la hoja **divisibilidad.xls**, contenida en esta página web.

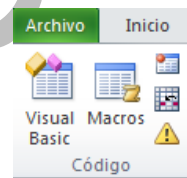
Estudia bien esta ventana. A la izquierda figuran las carpetas generales, que no debes tocar. Debajo figura la hoja que está abierta, **divisibilidad.xls**. En su carpeta **Módulos** figura el módulo

Module1 que el autor ha creado para contener todas sus macros. El código de estas macros figura a la derecha. En la imagen se puede ver el código de la función phi de Euler. que se ha implementado en las macros.

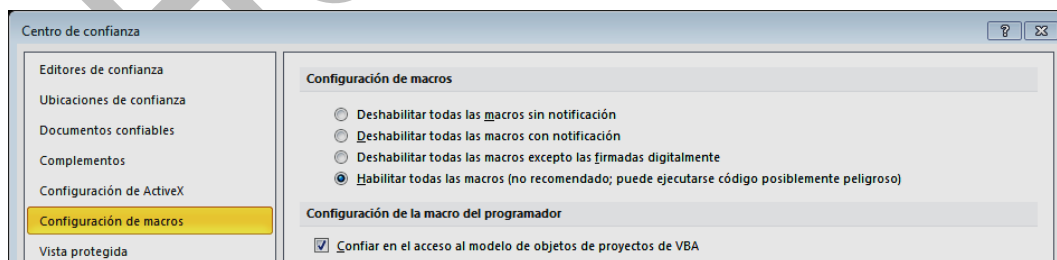
Los códigos de macros, si no tienes experiencia en programación, te parecerán ininteligibles. Cierra sin cambiar nada y no creas que hay que dominar esto para usar macros, porque seguidamente aprenderás a grabarlas automáticamente.

Seguridad de las macros

Como una macro puede contener virus, Excel define varios niveles de seguridad para el uso de las mismas. Puedes cambiar el nivel con el botón **Seguridad de macros**, incluido en la ficha **Programador**.



Si sólo vas a abrir hojas con macros de procedencia fiable puedes elegir el nivel más bajo, pero en caso contrario, vuelve a niveles altos de seguridad cuando trabajes con otras hojas.



GRABACIÓN DE MACROS

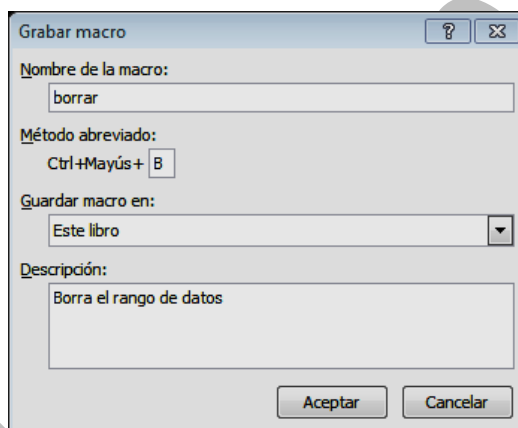
Para que te inicies en la grabación de macros, crearemos una que borre un rango de celdas, por ejemplo B4:D12

Para crear una macro mediante grabación has de seguir esta secuencia:

1. En la ficha **Programador** elige el botón **Grabar macro**



2. Se abrirá una ventana que contiene tres preguntas importantes:



Nombre: Puedes darle el de **borrar**

Método abreviado: No es imprescindible responder. Si eliges una combinación de teclas que ya esté ocupada, Excel te cambiará las de prefijo.

Guardar macro en: Lo normal es que le indiques "Este libro". Es lo más prudente, para no crear conflictos con otras hojas.

La descripción es optativa.

3. Realizas manualmente las operaciones que deseas que automatice la macro, en este caso borrar el rango B4:D12
4. Decides finalizar la grabación, para lo que usas el mismo botón de grabación, que ahora habrá cambiado a “Detener grabación”

La macro se guardará sin consultarte más.

Intenta recorrer esa secuencia con el ejemplo propuesto:

1. Recuerda bien qué celdas deseas borrar. En este caso, desde B4 hasta D12

Pulsa el botón **Grabar Macro**.

2. Asigna nombre y destino a la macro. Por ejemplo: “Borrar” y guardar “en este libro”

3. Realiza manualmente las operaciones para borrar los datos:

(a) Selecciona los datos B4:D12

(b) Pulsa la tecla **Supr**

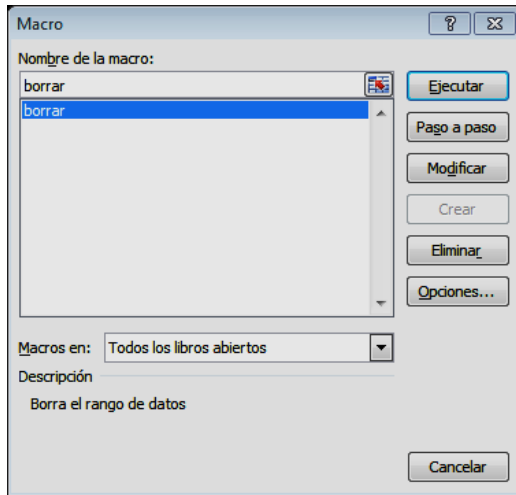
4. Pulsa sobre el botón de **Detener grabación**

5. Ejecuta la macro **Borrar**.

Escribe varios números en el rango B4:D12, para probar tu macro.

Si le has asignado una combinación de teclas, como **Ctrl+Mayús.+B**, úsala y verás cómo se borra todo. Si no le

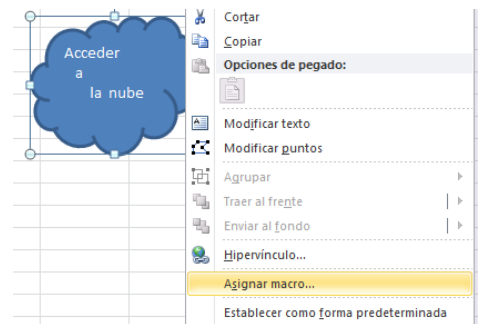
has asignado teclas, usa el botón **Macros**, que te abre la lista de las existentes. Señala “Borrar” y usa el botón “Ejecutar”



Inmediatamente se borrarán los datos del rango.

ASIGNACIÓN DE UNA MACRO A UN OBJETO

Para ejecutar una macro de forma inmediata un procedimiento rápido es asignarla a una imagen, forma, cuadro de texto o cualquier otro objeto que lo admita. Una vez creada la imagen o la forma, basta usar el botón derecho para acceder a la orden de asignar una macro.



En la imagen hemos insertado una nube con el texto adecuado y con el botón derecho hemos accedido a la opción de asignarle una macro.

En la lista de macros se elige la adecuada y se acepta.

Así, cada vez que pulsemos sobre la nube, se ejecutará la macro que hemos asignado.

PROGRAMACIÓN DE UNA MACRO SENCILLA

Crearemos alguna macro sencilla, y simultáneamente aprenderemos algunas operaciones sobre celdas.

Incremento del valor de una celda

Escribe un número cualquiera en la celda B4 y algún rótulo de presentación sobre ella. Por ejemplo así

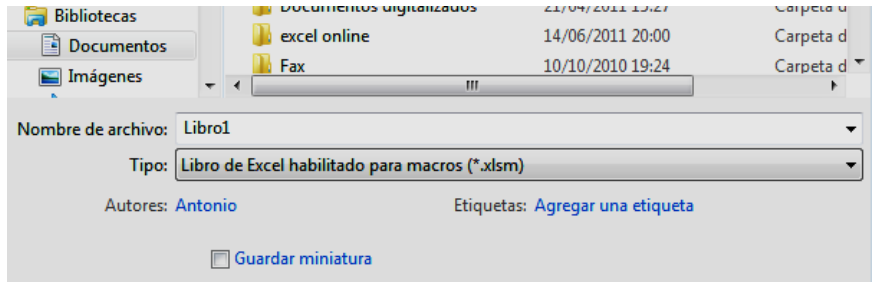
	A	B	C
1			
2			
3		Cambiamos B4	
4		348	
5			

Inserta a su lado una forma parecida a un botón

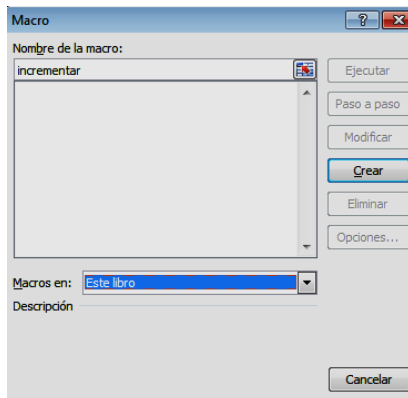
	Cambiamos B4	Incremento
	348	

Guarda la hoja con el nombre **prueba** u otro similar.

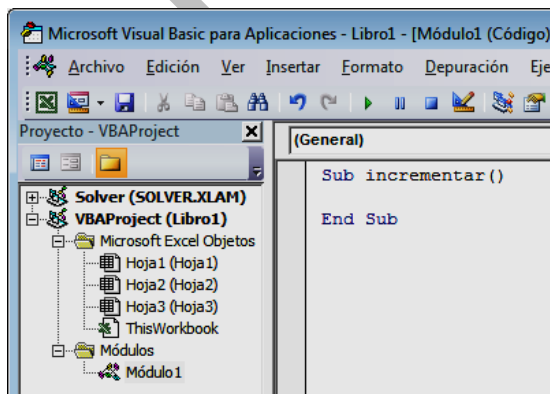
Un detalle importante: Debes guardarlo como libro *habilitado para macros*.



Accede a la edición de una macro nueva pulsando el botón **Macros** de la ficha **Programador**. En la ventana correspondiente asigna el nombre de **incrementar** a la macro, indica que se guarde “En este libro” y pulsa el botón de **Crear**.



Así accederás al Editor de Visual Basic, que te habrá escrito ya la cabecera y el final del código de macro. Simultáneamente te habrá creado un módulo contenedor.



Ves que ha escrito:

Sub incrementar()

End sub

El código de una macro ha de comenzar con la palabra clave **Sub** seguida del nombre asignado y de paréntesis para posibles argumentos (aquí no hay) y ha de terminar con **End sub**.

Nuestra macro está creada, pero vacía de contenido, no se sabe qué va a hacer. Para concretarlo has de escribir órdenes propias del lenguaje Basic entre la cabecera y el final. Para saber qué hacer necesitas un curso de lenguaje Basic, pero aquí usaremos algunas órdenes sencillas.

Escribe en medio este código de macros (puedes copiarlo desde esta misma página web con **CTRL-C** y **CTRL-V**), aunque todavía no lo entiendas:

Sub incrementar

Dim v

v=ActiveWorkbook.Sheets(1).Cells(4,2).Value

v=v+1

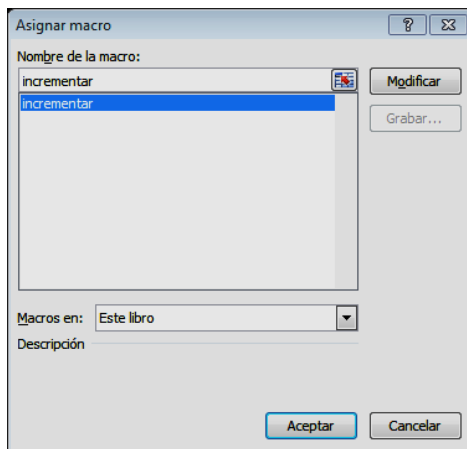
ActiveWorkbook.Sheets(1).Cells(4,2).Value=v

End sub

Esta macro te incrementa de 1 en 1 el valor de la celda B4. Después la explicaremos. Guarda la hoja que estás creando.

Ahora viene lo más fácil: Señala la imagen del botón y pulsa el botón derecho. Elige la opción de **Asignar**

macro... Se te abrirá este cuadro de diálogo y en él eliges la macro **incrementar**. Selecciónala y pulsa **Aceptar**.



Con esto, cada vez que pulses sobre la imagen (el puntero del ratón se transformará en un dedo señalador) se incrementará el valor de la celda B4.

Explicación del código

Aunque este documento no pretende enseñarte a programar, te vendrá bien saber interpretar el código que has copiado:

Sub incrementar

Dim v

v=ActiveWorkbook.Sheets(1).Cells(4,2).Value

v=v+1

ActiveWorkbook.Sheets(1).Cells(4,2).Value=v

End sub

Lo explicamos por partes:

Todas las macros han de comenzar con la palabra clave **Sub** seguida del nombre de la macro (en este caso **incrementar**) y han de terminar con la línea **End sub**.

La orden **Dim v** reserva memoria para una nueva variable a la que llamamos **v**. Si suprimes este tipo de órdenes la macro también funciona, pero si sigues por el camino de la programación es bueno que te acostumbres a dimensionar todo lo que uses.

En la tercera línea y en la quinta se usan tres palabras que conviene interpretar bien. Son nombres de objetos de Excel.

ActiveWorkbook: Se refiere al archivo actual de OpenOffice.org que tienes abierto.

Sheets(1): Se refiere a la hoja en la que se actuará, en este caso, la primera.

Cells(4,2).Value: Es la referencia al objeto celda B4. El 2 se refiere a la columna B, que es la segunda, y el 4 se refiere a la cuarta fila. La palabra Value lee el contenido de la celda.

Explicado esto, se entiende mejor lo que ejecuta la macro:

Línea 1: Comienzo obligatorio con la palabra **Sub** y el nombre de la macro.

Línea 2: Se dimensiona la variable

Línea 3: Se lee el valor de la celda B4 (palabra **value**) y el resultado se guarda en una variable llamada **v**.

Línea 4: Se incrementa **v** en una unidad.

Línea 5: Se copia el valor de **v** en la celda B4 de nuevo, pero ya incrementado.

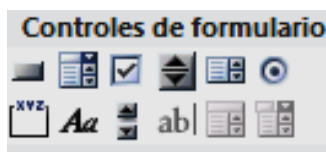
Línea 6: Sentencia obligatoria **end sub**

La explicación anterior sólo pretende que te vayas acostumbrando a la forma de trabajar en Basic. Un aprendizaje más profundo requiere un manual sistematizado.

Hojamat.es

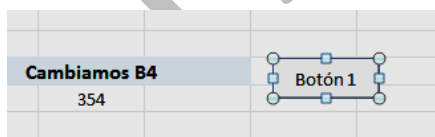
USO DE BOTONES

Se llama **control** a cualquier objeto activo, es decir, con sus propiedades y métodos, que se inserte en la hoja de cálculo para actuar interactivamente entre los usuarios y la hoja. Son controles los botones de acción, las casillas de verificación, las listas desplegables, etc. Estos elementos los has visto en cualquier programa de ordenador, y también se pueden usar en Excel.



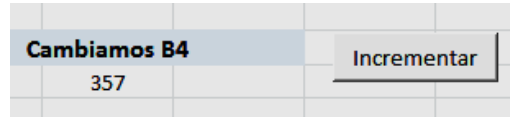
Para ver cuántos controles puedes usar, abre de nuevo a ficha del **Programador** y busca el botón **Insertar**. Ahora nos interesarán los controles de formulario entre los que se encuentran los botones.

Abre la hoja que has creado para incrementar la celda B4 borra el botón que dibujaste. En su lugar busca el control **botón** en la opción **Insertar** del **Programador** (suele ser el primero) e insértalo abriéndolo con el ratón hasta que tenga el tamaño que desees.



Cámbiale el nombre de **botón 1** por Incrementar. Si no sabes cómo, usa el botón derecho y elige **Modificar texto**. Puede ocurrir que también tengas que alterar el tamaño del botón tirando con el ratón de uno de los anclajes del borde (aparecen con el botón derecho).

También con el botón derecho accede a la asignación de una macro, si es que no te lo ha propuesto antes. Elige ***Incrementar*** y ya estará listo para funcionar. Púlsalo reiteradamente para observar su funcionamiento.



Hojamat.es

UN CASO PRÁCTICO

Diseñaremos un botón tal que al pulsar sobre él aparezca una ventana nueva en la que figure escrita la fecha actual. Los pasos que hemos de dar son:

Preparar una macro

Repite los pasos que diste para definir la macro del apartado anterior, la que incrementaba el valor de la celda B4. Crea un nuevo libro de Excel y guárdalo habilitado para macros llamado, por ejemplo, **fecha.xlsm**

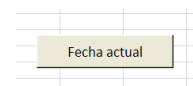
Accede al editor de macros como ya has aprendido, pero ahora escribe lo siguiente:

```
Sub fecha_actual  
msgbox(date$)  
End Sub
```

Guarda el archivo que has creado.

Diseñar un botón

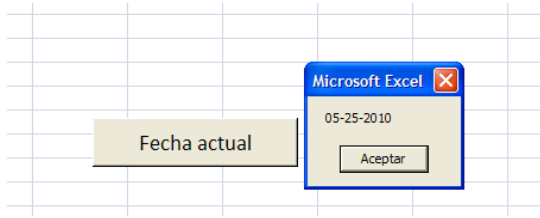
Abre el catálogo de controles de la forma que has aprendido. Elige botón e insértalo en la hoja. Cámbiale el rótulo a **Fecha actual**.



Asignación de la macro

Para que el botón sea útil deberemos asignarle una macro que se ejecute al pulsarlo. Con el botón derecho del ratón elige Asignar macro y le asignas **fecha_actual**

Con estas operaciones has terminado de configurar el botón de acción. Ahora debe funcionar: cada vez que pulses sobre el botón que has creado, aparecerá un mensaje con la fecha actual.



Práctica

Intenta repetir todo el trabajo creando un botón nuevo titulado "Hora actual" y en el código de la macro, en lugar de la variable `date$` escribe `time$`, para que te devuelva la hora en lugar de la fecha.

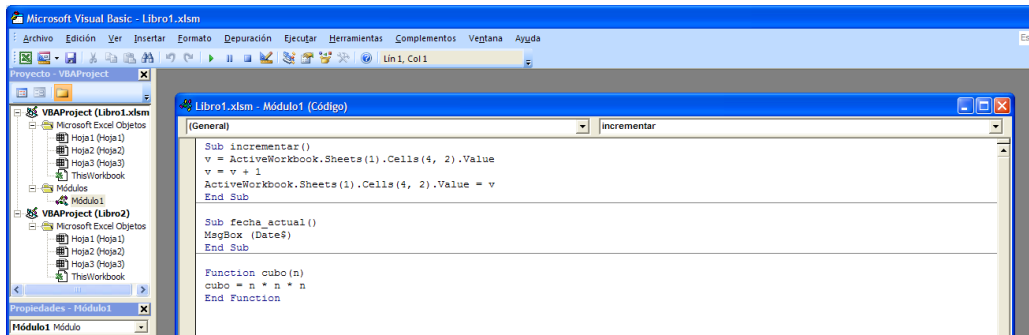
DEFINICIÓN DE FUNCIONES

En ocasiones desearás definir funciones que la hoja de cálculo no ofrece. Por ejemplo, en Electricidad nos puede convenir definir la resistencia equivalente a otras dos situadas en paralelo, o en Geometría, la función que devuelve una hipotenusa en función de los dos catetos. Mediante un uso sencillo de las macros puedes lograrlo.

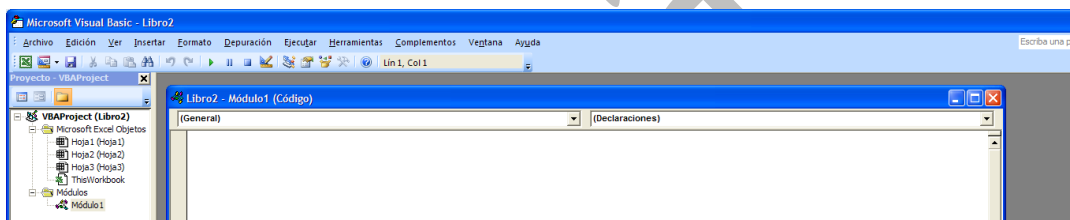
Secuencia para definir tus propias funciones

1) Abrir el Editor de Basic

En la ficha **Programador** usa el primer botón **Visual Basic**. Si ya tienes macros, se abrirá directamente el editor y podrás escribir



Si no has programado nada, no aparecerá ventana de edición, y tendrás que pedir **Insertar** y **Módulo**. De esta forma se abrirá una ventana en blanco.



2) Escritura del código

Terminada la secuencia anterior, comienza a escribir el código de una función.

Debes comenzar con

Function nombre de la función (argumento)

y terminar con

End function

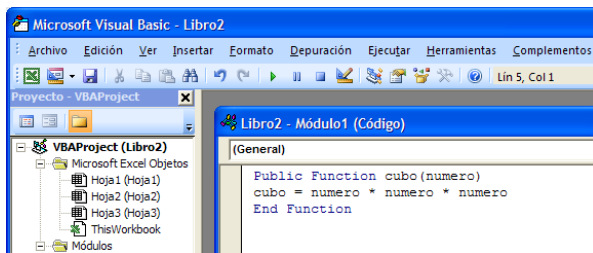
y entre ambas, el código de la función.

Para un mejor uso, delante de la palabra **function** es bueno escribir **Public**, para que sea accesible desde las hojas.

En ese código debemos usar el nombre de la función seguida del signo igual y de su definición. Es mejor verlo con un ejemplo:

```
Public Function cubo ( numero )  
cubo=numero*numero*numero  
End function
```

En el ejemplo, el nombre de la función es **cubo**, y su argumento **numero** (lo traduciríamos como "Cubo de un número"). Después volvemos a escribir cubo, el signo igual, y su definición.



3) Uso de la función

Una vez escrito el código, cierra el Editor de Basic y usa tu función en cualquier celda. En la imagen puedes ver una celda definida mediante la nueva función CUBO.

The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D
1				
2				
3		Número	4	
4		Su cubo	64	
5				
6				

The formula bar above the spreadsheet shows the formula in cell C4: `=cubo(C3)`.

Con esto ya tienes definida la función.

Con la técnica explicada, esa función sólo estará activa en la hoja de cálculo en la que la has creado, no en otras. Al cerrar la hoja ya no podrás usarla.

FUNCIÓN CON VARIOS ARGUMENTOS

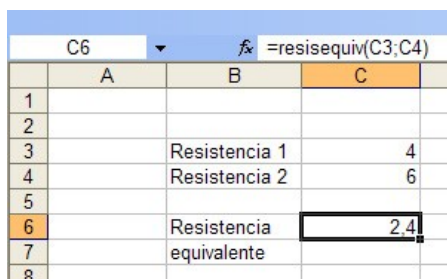
Una función puede actuar sobre varios argumentos, por ejemplo la resistencia equivalente a la que se aludía en el primer párrafo. En ese caso, se deberán separar mediante una coma:

Public Function resisequiv(r1, r2)

resisequiv = r1 * r2 / (r1 + r2)

End Function

Cuando uses esta función en una celda, debes sustituir la **coma** por un **punto y coma**, por ejemplo **resisequiv(4;6)**. Estudia el ejemplo de la imagen:



	A	B	C
1			
2			
3		Resistencia 1	4
4		Resistencia 2	6
5			
6		Resistencia equivalente	2.4
7			
8			

Variables auxiliares

En una definición puedes usar las estructuras del Basic: FOR...NEXT, SELECT CASE, etc. Aquí sólo usaremos DIM, para crear variables auxiliares. Observa este ejemplo

Function area6(lado)

Dim perimetro, apotema

perimetro = 6 * lado

apotema = lado * Sqr(3) / 2

area6 = perimetro * apotema / 2

End Function

que devuelve el área de un hexágono en función del lado. El nombre de la función, en este caso **area6**, debe figurar en la definición, aunque uses otras variables.

OTROS CONTROLES

Además de los botones, puedes usar todo un catálogo de controles. No vamos a estudiarlos todos, pero incluiremos dos tipos para que te animes a investigar por tu cuenta.

BOTONES DE OPCIÓN

Estos botones, casi siempre agrupados en conjuntos, nos sirven para elegir entre varias posibilidades. Para conocerlos mejor nos plantearemos un conversor de un número decimal a las expresiones binaria, hexadecimal y octal.

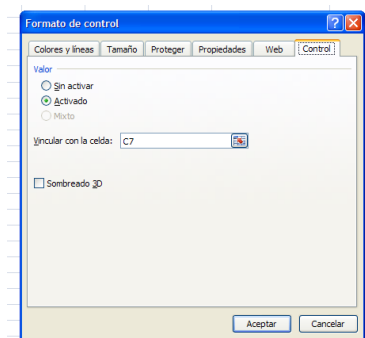
Un número en distintas bases			
			Número 18273
Binario	<input type="radio"/>	2	
Hexadecimal	<input checked="" type="radio"/>		Expresión 4761
Octal	<input type="radio"/>		

En la imagen puedes observar el grupo de botones de opción que servirá para elegir la expresión deseada. Según se pulse sobre uno u otro, aparecerá una expresión distinta. Es claro que la activación de uno de ellos desactiva a los demás.

Para construir este modelo de cálculo deberemos seguir estos pasos:

Construcción de los botones

Como en el caso de los botones, usa el botón **Insertar** de la ficha del **Programador**, pero ahora elige **botón de opción**. Inserta tres botones similares uno debajo del otro y adáptales el texto según la imagen anterior (hemos borrado el texto y después lo escribimos en otras celdas).



Excel necesita saber qué botón se ha pulsado en cada momento. Para ello selecciona los botones uno a uno con

el botón derecho del ratón y elige **Formato de control**. En la pestaña **Control** asigna a los tres **una misma celda** (en la imagen la D6)

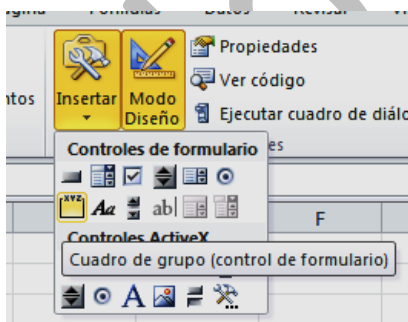
De esta forma, si se pulsa el primer control, D6 valdrá 1, si el segundo, 2, y 3 si se pulsa el último. Esto nos permite decidir qué conversión aplicamos al número. Podemos usar esta expresión:

```
=SI(D6=1;DEC.A.BIN(F5);SI(D6=2;DEC.A.HEX(F5);SI(D6=3;DEC.A.OCT(F5);"")))
```

Intenta algo similar a lo explicado para ver si funciona.

CUADROS DE GRUPO

A veces te puede interesar disponer de varios juegos de botones de opción que se relacionen por separado en varios grupos. Para conseguirlo rodea cada grupo de un **cuadro de grupo**, que lo insertas con el mismo procedimiento que los botones



Con la ayuda del ratón rodeas tus botones de opción con uno de esos cuadros y así los aíslas de cualquier otro grupo que crees.

En la construido

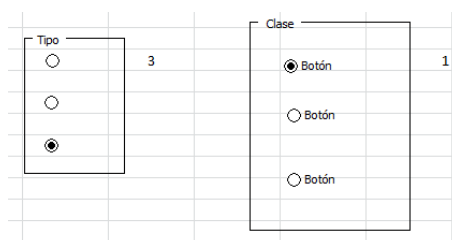


imagen se han dos grupos, uno

llamado Tipo y otro llamado Clase, que funcionan por separado y actúan sobre celdas distintas.

CASILLAS DE VERIFICACIÓN

Son controles que te permiten asignar a una celda los valores VERDADERO o FALSO. Si reúnes varias casillas de este tipo en una hoja, funcionan de forma independiente: el estado de una no influye en el de las demás.

Un ejemplo para que lo entiendas. Un mayorista aplica tres tipos distintos de descuento y según el cliente decidirá cuáles considera y cuáles no.

					Total factura
		Descuentos			26.300,00 €
FALSO	<input type="checkbox"/>	Por pronto pago	10%		0
VERDADERO	<input checked="" type="checkbox"/>	Rebajas	15%		3945
VERDADERO	<input checked="" type="checkbox"/>	Por pedido grande	5%		1315
				A pagar	21.040,00 €

Se insertan como los anteriores, pasando por la ficha del **Programador** y después el botón **Insertar**. Los buscas y los dibujas con el ratón. Con **Modificar texto** puedes borrar el que traen y después los asignas a celdas distintas. En la imagen son las que están a la izquierda de los controles.

Si se activa uno de ellos, con la función SI decidimos si se aplica el descuento o no. Basta una fórmula del tipo

=SI(A7=VERDADERO;E7*G5;0)

Si te decides, puedes construir un modelo de cálculo similar.

Los rótulos de VERDADERO y FALSO pueden resultar molestos. Si quieres que no se vean, asigna a sus celdas como color de texto **el blanco**, y así no se verán. De esta forma los hemos ocultado en el anterior modelo:

Descuentos	
<input type="checkbox"/>	Por pronto pago
<input checked="" type="checkbox"/>	Rebajas
<input checked="" type="checkbox"/>	Por pedido grande

Así ya no se ven a la izquierda de los controles.

MACROS DE APERTURA

En ocasiones podemos desear que se ejecute cualquier operación al abrir una hoja de cálculo, como borrar un rango, abrir una hoja determinada, dar un valor a una celda, etc.

Para lograrlo debes, en primer lugar, escribir o grabar una macro con las operaciones que desees. Una vez escrita, el procedimiento para que se ejecute al abrir una hoja es el siguiente:

Escribes o grabas una macro y le das el nombre de **auto_open**. Lo que escribas como código se ejecutará al abrir la hoja.

Por ejemplo, entra en el editor de Visual Basic, inserta un módulo nuevo y escribe

```
Sub auto_open()  
Msgbox("Hola")  
End sub
```

Guarda el archivo y ciérralo. Cuando lo vuelvas a abrir, te saludará con un "Hola" al comenzar.

Hojamat.es