

Macros en Excel

Ejemplos prácticos



Pablo E. Fernández Casado

Macros en Excel

Ejemplos prácticos

Pablo E. Fernández Casado



Ra-Ma®

edü®

BOGOTÁ - MÉXICO, D.F.

Casado Fernández, Pablo E, *et. al.*

Macros en Excel. Ejemplos prácticos / Pablo E. Fernández Casado--. Bogotá: Ediciones de la U, 2024

200 p. ; 24 cm

ISBN 978-958-792-653-8 e-ISBN 978-958-792-654-5

1. Informática 2. Excel I. Tít.

5 ed.

*Edición original publicada por © Editorial Ra-ma (España)
Edición autorizada a Ediciones de la U para Colombia*

Área: Informática

Primera edición: Bogotá, Colombia, enero de 2024

ISBN. 978-958-792-653-8

- © Pablo E. Fernández Casado
- © Ra-ma Editorial. Calle Jarama, 3-A (Polígono Industrial Igarsa) 28860 Paracuellos de Jarama
www.ra-ma.es y www.ra-ma.com / E-mail: editorial @ra-ma.com
Madrid, España
- © Ediciones de la U - Carrera 27 #27-43 - Tel. (+57) 601 6455049
www.edicionesdelau.com - E-mail: editor@edicionesdelau.com
Bogotá, Colombia

Ediciones de la U es una empresa editorial que, con una visión moderna y estratégica de las tecnologías, desarrolla, promueve, distribuye y comercializa contenidos, herramientas de formación, libros técnicos y profesionales, e-books, e-learning o aprendizaje en línea, realizados por autores con amplia experiencia en las diferentes áreas profesionales e investigativas, para brindar a nuestros usuarios soluciones útiles y prácticas que contribuyan al dominio de sus campos de trabajo y a su mejor desempeño en un mundo global, cambiante y cada vez más competitivo.

Coordinación editorial: Adriana Gutiérrez M.

Carátula: Ediciones de la U

Impresión: DGP Editores SAS

Calle 63 #70D-34, Pbx (+57) 601 7217756

Impreso y hecho en Colombia

Printed and made in Colombia

No está permitida la reproducción total o parcial de este libro, ni su tratamiento informático, ni la transmisión de ninguna forma o por cualquier medio, ya sea electrónico, mecánico, por fotocopia, por registro y otros medios, sin el permiso previo y por escrito de los titulares del Copyright.

*Al principio Dios creó al hombre,
y viéndole tan débil, le dio el **perro***

ÍNDICE

INTRODUCCIÓN	13
QUÉ ES UNA MACRO DE EXCEL.....	13
LA SEGURIDAD Y LAS MACROS	14
VISUAL BASIC Y SU EDITOR DE EXCEL.....	16
OBJETOS, PROPIEDADES, MÉTODOS Y EVENTOS EN EXCEL.....	18
CÓMO HABILITAR/DESHABILITAR LAS MACROS EN EXCEL.....	20
CAPÍTULO 1. INTRODUCCIÓN A VISUAL BASIC FOR APPLICATIONS.....	23
1.1 TIPOS DE DATOS EN VBA.....	23
1.2 DECLARACIÓN DE VARIABLES.....	24
1.3 OPERADORES	24
1.3.1 Aritméticos	25
1.3.2 Concatenación	25
1.3.3 Comparación	26
1.3.4 Lógicos	26
1.3.5 Operador Like	26
1.4 BUCLES	30
1.4.1 For Each	30
1.4.2 For Next.....	30
1.4.3 Do While	30
1.4.4 Do Until.....	31
1.5 CONDICIONANTES	31
1.5.1 If	31
1.5.2 Case	32

1.6	ARRAYS.....	32
1.6.1	IsArray, LBound, UBound y otros comandos	34
1.7	CONVERSIONES A OTROS FORMATOS	36
1.7.1	Formatos numéricos	36
1.7.2	Formatos de tipo fecha	36
1.7.3	Formatos de tipo cadena.....	37
1.7.4	Formatos definidos por el usuario	38
1.7.5	Conversión y manipulación de cadenas	38
1.7.6	Conversiones de tipo	41
1.8	RANGOS Y CELDAS.....	44
1.9	CONTROL DE ERRORES	46
1.9.1	CVErr	46
1.9.2	On Error.....	46
1.10	MANIPULACIÓN DE COLORES	47
CAPÍTULO 2. INTRODUCCIÓN A LAS MACROS		49
2.1	INTRODUCCIÓN	49
2.1.1	Sub y Function	49
2.1.2	Eventos	50
2.2	GRABACIÓN DE MACROS.....	52
2.2.1	Macro 1: Hola mundo	52
2.3	CREACIÓN DE MACROS POR CÓDIGO.....	55
2.3.1	Macro 2: Hola mundo	55
2.3.2	Macro 3: Anidación y suma en celdas separadas	58
2.3.3	Macro 4: Rellenar un ComboBox	58
2.3.4	Macro 5: Detección de número primo.....	61
CAPÍTULO 3. TRABAJANDO CON EL OBJETO APPLICATION.....		63
3.1	INTRODUCCIÓN	63
3.2	PROPIEDADES MÁS COMUNES	64
3.2.1	ActiveWorkbook	64
3.2.2	ActiveCell.....	64
3.2.3	Caption	64
3.2.4	CalculateBeforeSave	65
3.2.5	Calculation	65
3.2.6	CheckSpelling	65
3.2.7	CommandBars.....	65
3.2.8	Cursor.....	66
3.2.9	DecimalSeparator	66
3.2.10	Dialogs	66
3.2.11	DisplayAlerts.....	67
3.2.12	EnableEvents	67
3.2.13	GenerateGetPivotData.....	67
3.2.14	ScreenUpdating	68
3.2.15	ThousandsSeparator	68

3.2.16	Visible.....	68
3.2.17	WorksheetFunction	69
3.3	MÉTODOS Y EVENTOS MÁS COMUNES	69
3.3.1	Calculate.....	69
3.3.2	InputBox.....	70
3.3.3	Goto.....	71
3.3.4	Evaluate.....	72
3.3.5	Intersect.....	73
3.3.6	OnKey.....	73
3.3.7	OnRepeat.....	75
3.3.8	OnUndo.....	75
3.3.9	OnTime.....	76
3.3.10	Repeat.....	76
3.3.11	Run.....	77
3.3.12	SendKeys.....	77
3.3.13	Quit.....	77
3.3.14	Union.....	78
3.3.15	Undo.....	78
3.4	EJEMPLOS PRÁCTICOS.....	78
3.4.1	Macro 6: Añadir el libro de Excel como correo adjunto.....	78
3.4.2	Macro 7: Crear una macro con aviso de final de jornada.....	79
3.4.3	Macro 8: Pedir la selección de un rango de celdas y mostrar un diálogo con la suma total.....	80
3.4.4	Macro 9: Cambiar separadores numéricos de la celda activa.....	81
3.4.5	Macro 10: Deshacer el último cambio al pulsar la tecla F9.....	81
3.4.6	Macro 11: Añadir la hora de última actualización de una celda.....	82
3.4.7	Macro 12: Deshabilitar todas las barras de comandos menos la correspondiente a celdas.....	83
3.4.8	Macro 13: Resaltar las celdas con palabras mal escritas.....	84
3.4.9	Macro 14: Ordenar alfabéticamente las hojas de calculo de un libro.....	85
3.4.10	Macro 15: Borrar todas las hojas de cálculo excepto la activa.....	86
	CAPÍTULO 4. TRABAJANDO CON EL OBJETO WORKBOOK	87
4.1	INTRODUCCIÓN.....	87
4.2	PROPIEDADES MÁS COMUNES.....	88
4.2.1	ActiveSheet.....	88
4.2.2	Name.....	88
4.2.3	Names.....	88
4.2.4	Path.....	88
4.2.5	Password.....	89
4.2.6	Permission.....	89
4.2.7	Save.....	89
4.2.8	Sheets y Worksheets.....	90
4.2.9	WritePassword.....	90
4.2.10	WriteReserved y WriteReservedBy.....	90

4.3	MÉTODOS Y EVENTOS MÁS COMUNES	91
4.3.1	Activate	91
4.3.2	BeforeClose	91
4.3.3	BeforePrint	92
4.3.4	BeforeSave	92
4.3.5	Close	93
4.3.6	Deactivate	93
4.3.7	NewChart	93
4.3.8	NewSheet	94
4.3.9	Open	94
4.3.10	Protect	95
4.3.11	PivotCaches	95
4.3.12	Save	97
4.3.13	SaveAs	97
4.3.14	SaveCopyAs	99
4.3.15	SheetActivate	99
4.3.16	SheetChange	100
4.3.17	SheetPivotTableUpdate	100
4.3.18	SheetSelectionChange	101
4.3.19	SendMail	101
4.3.20	Unprotect	102
4.3.21	WindowResize	102
4.4	EJEMPLOS PRÁCTICOS	103
4.4.1	Macro 16: Abrir un libro de Excel desde otro	103
4.4.2	Macro 17: Guardar el libro de trabajo de forma automática con nombre de usuario	104
4.4.3	Macro 18: Resaltar los rangos con nombre	104
4.4.4	Macro 19: Ocultar todas las hojas de cálculo menos la activa	105
4.4.5	Macro 20: Desocultar todas las hojas de cálculo	105
4.4.6	Macro 21: Convertir a mayúsculas o minúsculas un rango de celdas seleccionado en función de un parámetro	106
4.4.7	Macro 22: Proteger una hoja de trabajo con contraseña	107
4.4.8	Macro 23: Desproteger una hoja de trabajo	107
4.4.9	Macro 24: Crear una tabla de contenidos	108
4.4.10	Macro 25: Crear y actualizar una tabla dinámica en Excel	110
CAPÍTULO 5. TRABAJANDO CON EL OBJETO WORKSHEET		113
5.1	INTRODUCCIÓN	113
5.2	PROPIEDADES MÁS COMUNES	114
5.2.1	Application	114
5.2.2	Cells	114
5.2.3	CodeName	114
5.2.4	Columns	114
5.2.5	Comments	115
5.2.6	CommentsThreaded	115
5.2.7	DisplayPageBreak	116

5.2.8	DisplayRightToLeft.....	116
5.2.9	EnableCalculation	116
5.2.10	EnablePivotTable.....	116
5.2.11	EnableSelection	117
5.2.12	Index	117
5.2.13	Name	117
5.2.14	Next	117
5.2.15	Previous	118
5.2.16	PageSetup	118
5.2.17	Protection	118
5.2.18	Range.....	118
5.2.19	Rows.....	118
5.2.20	Type.....	119
5.2.21	UsedRange	119
5.2.22	Visible.....	119
5.3	MÉTODOS Y EVENTOS MÁS COMUNES	120
5.3.1	Activate	120
5.3.2	BeforeDelete.....	120
5.3.3	BeforeDoubleClick.....	120
5.3.4	BeforeRightClick	121
5.3.5	Calculate.....	122
5.3.6	Change.....	122
5.3.7	Copy	123
5.3.8	Deactivate.....	123
5.3.9	Delete	123
5.3.10	Evaluate.....	124
5.3.11	FollowHyperlink	124
5.3.12	FormatConditions.....	125
5.3.13	Move.....	127
5.3.14	Paste	128
5.3.15	PasteSpecial.....	129
5.3.16	PrintOut	130
5.3.17	PivotTables	131
5.3.18	PivotTableUpdate	132
5.3.19	Protect.....	133
5.3.20	Select	135
5.3.21	SelectionChange.....	136
5.4	EJEMPLOS PRÁCTICOS.....	136
5.4.1	Macro 26: Bloquear el botón derecho del ratón.....	136
5.4.2	Macro 27: Conocer el tipo de dato introducido.....	137
5.4.3	Macro 28: Ordenación automática al detectar un cambio.....	138
5.4.4	Macro 29: Actualizar tablas dinámicas al activar una hoja.....	139
5.4.5	Macro 30: Copia de seguridad al eliminar una hoja.....	140
5.4.6	Macro 31: Seleccionar las celdas de una tabla y formatearla.....	141
5.4.7	Macro 32: Buscar un valor en una tabla.....	143

5.4.8	Macro 33: Calcular el total de un rango seleccionado	145
5.4.9	Macro 34: Guardar el libro Excel al desactivar una hoja	146
5.4.10	Macro 35: Filtrado de registros en una tabla.....	146
5.4.11	Macro 36: Crear un calendario para un mes dado.....	149
5.4.12	Macro 37: Rellenar una hoja de Excel con datos extraídos de otros libros	155
CAPÍTULO 6. TRABAJANDO CON EL OBJETO USERFORM.....		159
6.1	INTRODUCCIÓN	159
6.2	CREACIÓN Y MANIPULACIÓN DE USERFORMS	160
6.3	ACEDIENDO A USERFORM Y SUS CONTROLES	161
6.4	PROPIEDADES DE LOS USERFORM.....	162
6.5	EVENTOS Y MÉTODOS MÁS COMUNES	163
6.5.1	Activate	163
6.5.2	Add.....	164
6.5.3	AddControl.....	166
6.5.4	Click	167
6.5.5	DbiClick	167
6.5.6	Deactivate.....	168
6.5.7	Error	168
6.5.8	Initialize.....	169
6.5.9	Hide	169
6.5.10	KeyDown y KeyUp.....	169
6.5.11	KeyPress.....	171
6.5.12	Load y Unload.....	171
6.5.13	MouseDown, MouseMove y MouseUp	171
6.5.14	PrintForm	172
6.5.15	QueryClose.....	173
6.5.16	Resize	174
6.5.17	RemoveControl	174
6.5.18	Scroll	174
6.5.19	Show.....	175
6.5.20	Terminate.....	176
6.6	EJEMPLOS PRÁCTICOS.....	176
6.6.1	Macro 38: Crear un diálogo emergente personalizado.....	176
6.6.2	Macro 39: Crear el formulario de una calculadora básica: parte 1	178
6.6.3	Macro 40: Crear un formulario simple de registro por código.....	179
6.6.4	Macro 41: Crear un formulario de acceso con usuario y contraseña ...	186
6.6.5	Macro 42: Crear un formulario para asignar formatos condicionales .	187
6.6.6	Macro 43: Descargar de la nube 1: URLDownloadToFile y	
6.6.7	Google Drive	190
6.6.8	Macro 44: Descargar de la nube 2: CreateObject y Pixabay.....	193
CAPÍTULO 7. REFERENCIAS.....		197
MATERIAL ADICIONAL.....		199



INTRODUCCIÓN

QUÉ ES UNA MACRO DE EXCEL

Una macro puede definirse como una serie de instrucciones o comandos que se utilizan para automatizar tareas repetitivas o recurrentes. Esta automatización se puede realizar a través la introducción manual usando un editor de texto o a través de la grabación de un conjunto de acciones que, más tarde, podrán ser o no reproducidas a petición del usuario.

Las macros, también nos ofrecen la posibilidad de realizar tareas complejas de una forma más clara, organizada y eficiente. Por ejemplo, podríamos crear una macro para formatear automáticamente una tabla, realizar unos cálculos específicos, generar unos informes personalizados o, incluso, realizar una serie de acciones específicas en función de ciertas situaciones o circunstancias.

Cuando un usuario crea una macro en Excel, en realidad, lo que se está haciendo es crear una secuencia de comandos, fórmulas, formatos u otras operaciones que normalmente se realizarían de manera manual dentro de un módulo de código denominado VBA, acrónimo de Visual Basic for Applications (Visual Basic para Aplicaciones) y que se almacena dentro del propio archivo de Excel que se está utilizando en ese momento.

Es por ello que, si tras la creación de una macro miramos su contenido, lo que veremos es un conjunto de instrucciones escritas mediante un lenguaje de programación, que denominamos de alto nivel, diseñado específicamente para la automatización de tareas en aplicaciones de la suite de Microsoft Office, lo que incluye Microsoft Excel.

Pero, ¿y cómo se activa esta macro que hemos creado? Pues, una vez que hemos creado nuestra macro, la podemos asignar a cualquier acción que pueda ser detectada o usada como disparador. Esto es, la pulsación de un botón, de una combinación de teclas, el cambio de valor de una celda o, incluso, la activación o enfocado de una hoja de cálculo concreta.

A continuación, se muestra el ejemplo de una macro sencilla que se ejecuta de manera automática cuando el valor de la celda A1 cambia:

```
Private Sub Worksheet_Change(ByVal Target As Range)
    ' Verifica si la celda modificada es A1
    If Target.Address = "$A$1" Then
        ' Instrucciones a ejecutar
        MsgBox "El valor de A1 ha cambiado a: " & Target.Value
    End If
End Sub
```

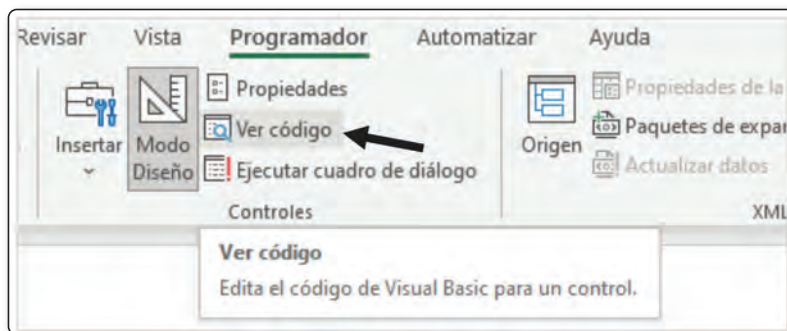
En este ejemplo, la macro se dispara o ejecuta con el evento de cambio **Worksheet_Change** que, aunque se verá más adelante, indica al Excel que se ejecute cuando se produzca un cambio en cualquier celda de la hoja de cálculo. Sin embargo, gracias al argumento **Target** que recibimos en la llamada del evento, podremos preguntar si es la celda deseada (en este caso A1) mediante el bloque **If** declarado a continuación. Este bloque viene a decir que, cuando la propiedad **Target.Address** sea la celda **A1** ejecute todas las instrucciones que están delimitadas por él.

LA SEGURIDAD Y LAS MACROS

La seguridad en las macros de Excel es un aspecto más que importante debido, fundamentalmente, a que contienen fragmentos o trozos de código ejecutable que pueden utilizarse con fines maliciosos. Por esta razón, las macros de Excel siempre vienen desactivadas, porque, su activación y uso puede representar un riesgo para la seguridad de los datos y/o del sistema.

A continuación, se comentan algunas consideraciones para garantizar la seguridad cuando se trabaja con macros en Excel:

- **Advertencia antes de la habilitación de macro:** Excel presenta una configuración de seguridad predeterminada que controla cómo se manejan las macros en los archivos. En general, se puede personalizar en función de nuestras necesidades, no obstante, lo recomendable y más frecuente es dejar que Excel muestre una advertencia antes de habilitar o deshabilitar las macros. De esta manera, tendremos la opción de revisar el código y asegurarnos de que proviene de una fuente confiable antes de ejecutarlo.
- **Fuentes confiables:** siempre se debe estar alerta cuando se vaya a abrir un archivo de Excel que contenga macros. Sin embargo, esta precaución debe incrementarse si procede de una fuente desconocida o no confiable. Esto es, nunca se deben abrir archivos que no permitan una verificación previa de su origen y contenido.
- **Habilitar las macros estrictamente cuando sea necesario:** dado que las macros pueden ser utilizadas como vectores de ataque, lo recomendable es que sólo se habiliten en momentos puntuales, y deshabilitándolas cuando ya no sean necesarias. Además, con ello, reduciremos el riesgo de que se ejecuten de forma accidental o malintencionada.
- **Prevenir antes que curar:** una buena práctica es que, antes de habilitar una macro, revisemos su código. Con ello, no sólo evitaremos la ejecución de instrucciones maliciosas, sino que también podremos evitar acciones inesperadas.



- **Firmas digitales:** aunque no es obligatorio, la recomendación es utilizar firmas digitales. Con ello, no sólo podremos verificar la autenticidad y su origen, sino que también podremos chequear la integridad de las macros ya que, al firmarlas digitalmente, podremos garantizar que no ha sido modificada por terceros y que proviene de una fuente confiable. Para ver las firmas digitales debemos ir a la opción **Archivo > Información**

- > **Ver Firmas.** Una vez que estemos ahí, deberemos hacer clic en la flecha que apunta hacia **abajo** y seleccionar **Detalles de la Firma**.
- ✔ **Actualizaciones y parches al día:** cuando se utiliza un software como Excel, es más que recomendable que esté actualizado con la última versión y con todos los parches de seguridad de Microsoft aplicados. Las actualizaciones periódicas ayudarán a abordar las posibles vulnerabilidades y mejorar la seguridad general del sistema.

VISUAL BASIC Y SU EDITOR DE EXCEL

Visual Basic (VB) es un lenguaje de programación de alto nivel desarrollado por Microsoft y que, habitualmente, forma parte del entorno de desarrollo integrado (IDE) llamado Visual Studio. Visual Basic es un lenguaje de programación que se utiliza comúnmente para desarrollar aplicaciones de software en entornos de Windows y para la automatización de tareas en aplicaciones de Microsoft Office, como Excel, Word o Access.

Entre sus principales características podemos encontrar que:

1. **Resulta fácil de aprender:** esto es debido a que Visual Basic utiliza una sintaxis sencilla y basada en lenguaje natural, lo que lo vuelve más "accesible" para todas aquellas personas que se quieren iniciar en la programación.
2. **Está orientado a eventos:** esto viene a decir que las acciones del usuario, como hacer clic en un botón, lanzan o disparan eventos que pueden ser controlados y respondidos mediante código, lo que hace que el desarrollo de interfaces de usuario interactivas sea más intuitivo.
3. **Integración con aplicaciones de Windows y Office:** esto es evidente, puesto que ya lo hemos comentado. Dado que Visual Basic se integra estrechamente con todas las aplicaciones de Windows y Office, podemos aprovechar todas sus capacidades y características para crear programas, macros de Excel, para desarrollar aplicaciones de Windows Forms, etcétera.
4. **Soporte de la comunidad:** como buen lenguaje vivo, Visual Basic cuenta con una amplia comunidad de desarrolladores y documentación disponible en línea, lo que ayuda y facilita el aprendizaje, la resolución de problemas y el acceso a recursos adicionales.

5. **Programación basada en componentes:** Visual Basic permite la programación basada en componentes utilizando el modelo de desarrollo conocido como Component Object Model (COM), lo que nos permite crear componentes reutilizables y utilizarlos en diferentes aplicaciones o sistemas.

En cuanto al Editor de Visual Basic de Excel, podemos decir que es una herramienta integrada que permite crear, editar y administrar código creado en Visual Basic for Applications (VBA).

Para acceder al Editor de Visual Basic desde Excel, bastará con presionar **ALT + F11** o pulsar en el icono **Visual Basic**, situado como primer elemento a la izquierda, dentro de la opción **Programador**.

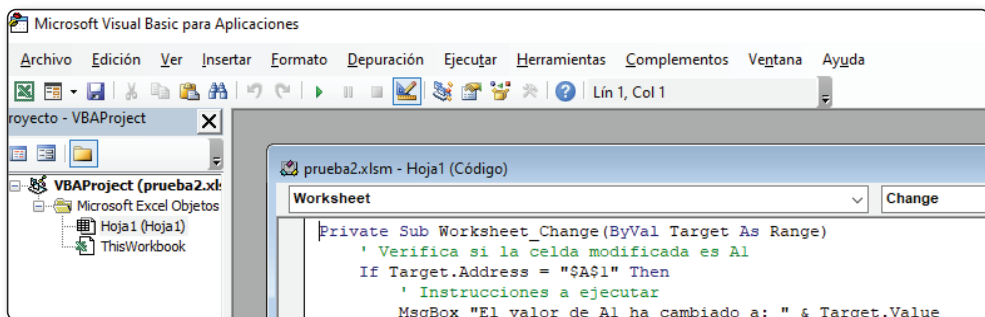
i NOTA

Para visualizar la pestaña reservada a Visual Basic y Macros, debemos hacer clic en la pestaña **Archivo** y después pulsar en el botón que indica **Opciones**. A continuación, deberemos seleccionar la categoría **Personalizar cinta de opciones** y, en la zona **Pestañas principales**, pulsar la opción de **Desarrollador**. Tras ello, pulsar en el botón de **Aceptar**.

Aunque en la imagen, la opción de "Programador" aparece en segundo lugar, lo normal es que se muestre a continuación de "Automatizar".

Si queremos ver las macros, podemos hacerlo pulsando en el icono que pone "Macros" de esta pestaña o también desde su icono identifioco en la pestaña "Vista".

Si pulsamos en el icono de "Visual Basic" o la combinación ALT + F11 deberíamos poder ver algo como:



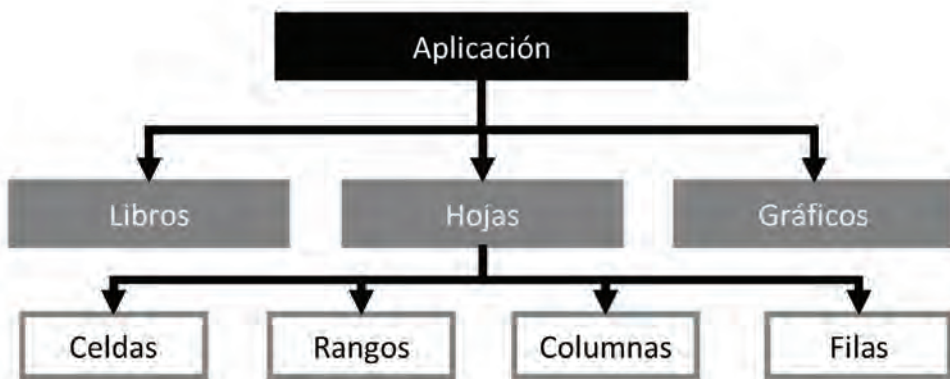
OBJETOS, PROPIEDADES, MÉTODOS Y EVENTOS EN EXCEL

Objetos

Un **objeto** es aquel elemento que representa a una entidad dentro de una aplicación. Esto es, por ejemplo, una hoja de cálculo, una celda, una imagen, un rango, un formulario, etcétera.

Todos los objetos podrán contener otros objetos dentro, no obstante, es frecuente diferenciar este tipo de casos. Mientras que un objeto es una entidad que posee unas determinadas propiedades, métodos y eventos, una colección es un conjunto de objetos que, a su vez, tiene propiedades, métodos y eventos.

Este es el caso del objeto Application, el cual contiene otros objetos como son Workbook, Worksheet, Chart, Range, Column, Row o Cells.



Como uno se podrá imaginar, en Excel existe una jerarquía y, esta jerarquía empieza por el objeto Application. De este objeto dependen, por decirlo así, una multitud de objetos, y cada uno de ellos tiene un propósito. Esto es, Excel posee objetos para manejar los estilos, los nombres definidos para celdillas y rangos, los gráficos, los libros, las hojas de cálculo, los rangos, las ventanas, los complementos y, cómo no, los proyectos de Visual Basic.

Propiedades

Una **propiedad** es una palabra clave que hace referencia a un valor dado y que, por lo general, define una característica determinada. Por ejemplo, si

estuviésemos hablando de una persona, sus propiedades podrían ser el color de sus ojos, el color del pelo, su altura, peso, sexo, número de pie, etcétera.

Cuando hablamos de Excel, sus propiedades pueden ser la dirección o el valor de una celda.

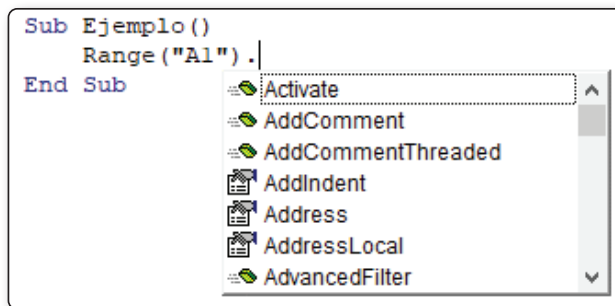
```
Range("A1").Value = "Hola"
```

Métodos

Un **método** es una acción que puede hacer o ejecutar un determinado objeto y que, normalmente, no está vinculado directamente a una petición de usuario. Si esta idea la contextualizamos a Excel, este tipo de acciones podrían ser seleccionar una celda o borrar su contenido:

```
Range("A1").Select  
Range("A1").Clear
```

Si se desean consultar o ver las propiedades, métodos y eventos de un objeto determinado bastará con escribir el nombre del objeto y presionar la tecla de punto:



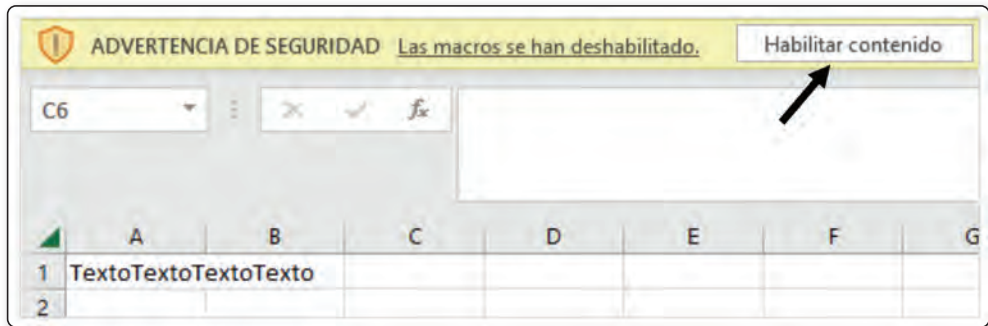
Eventos

Un **evento**, por su parte, es una acción asociada a un determinado suceso que produce, por lo general, el usuario. Los eventos de un objeto, al igual que las propiedades y métodos, podrán variar mucho en función de lo que representan, como ya se verá más adelante, pero, un ejemplo podría ser el evento **Activate**:

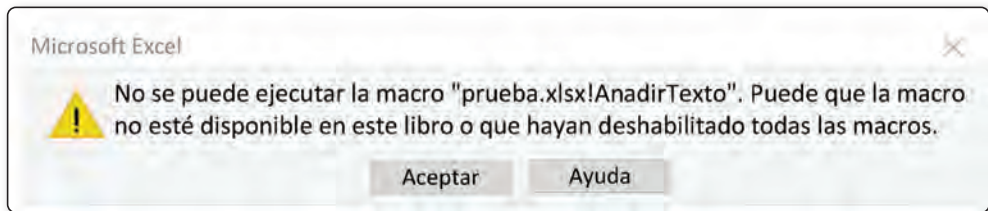
```
Worksheet_Activate()
```

CÓMO HABILITAR/DESHABILITAR LAS MACROS EN EXCEL

Cuando se abre un archivo de Excel que contiene macros, lo frecuente es que se nos pregunte si deseamos habilitar o no su contenido:



Mientras no pulsemos el botón de **Habilitar contenido**, todas las acciones, eventos y botones que estén asociados a una macro no se ejecutarán. En su lugar, se presentará, o debería presentarse, un mensaje similar al que se muestra a continuación:



Como se ha comentado anteriormente, antes de habilitar las macros será importante comprobar si el archivo es de una fuente confiable y si el contenido ejecutable es seguro.

Para ver si el contenido es seguro sólo podremos hacerlo activando las macros previamente, lo cual puede ser un riesgo puesto que podrían ejecutarse macros sin previo aviso.

No obstante, aunque no podamos ver su contenido y, aunque estén totalmente deshabilitadas, siempre podremos ver las macros definidas pulsando **Alt + F8** o pulsando en el icono **Macros**, situado al final de la pestaña **Vista**, y haciendo clic en **Ver Macros** posteriormente.