

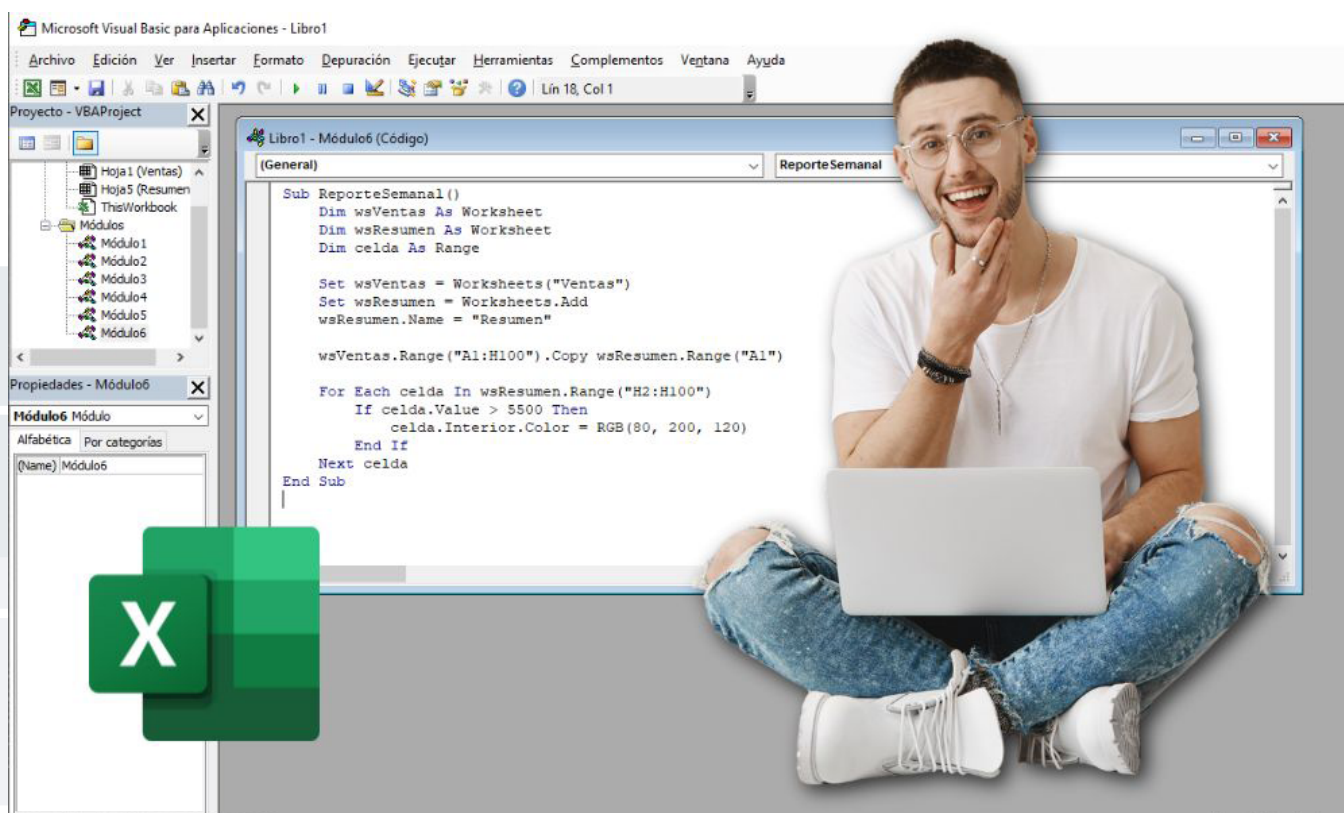


IMECAF®

INSTITUTO MEXICANO DE CONTABILIDAD,
ADMINISTRACIÓN Y FINANZAS

Macros en Excel: Domínalas y ahorra horas de trabajo

Octubre, 2025 - Blog IMECAF



Introducción

Si alguna vez te has encontrado haciendo las mismas tareas repetitivas en Excel, sabes lo tedioso que puede ser copiar fórmulas, formatear celdas o consolidar datos una y otra vez. Aquí es donde las Macros en Excel se convierten en tus mejores aliadas.

Una macro es, básicamente, una serie de instrucciones que Excel puede ejecutar automáticamente para realizar tareas repetitivas. Con ellas, puedes ahorrar tiempo, reducir errores y mejorar la eficiencia de tu trabajo. Lo mejor de todo es que no necesitas ser un experto en programación para empezar; con un poco de práctica, cualquiera puede grabar y usar macros.

En este artículo aprenderás qué son las macros, sus tipos, cómo crearlas y grabarlas, ejemplos prácticos paso a paso y consejos para aplicarlas en tu trabajo diario.

¿Qué son las Macros en Excel?

Definición de macro

Una macro en Excel es una serie de comandos o acciones que se registran para ejecutarse automáticamente cuando las necesites. Piensa en ella como un “botón mágico” que realiza varias tareas por ti con un solo clic.

Por ejemplo, si cada semana necesitas crear un reporte con varios pasos: copiar datos, dar formato, crear gráficos y enviar el archivo, una macro puede hacer todo eso automáticamente.

Beneficios de usar macros

- **Ahorro de tiempo:** Las tareas que solían tomar horas pueden completarse en segundos.
- **Reducción de errores:** Al automatizar procesos, disminuyes la probabilidad de equivocarte.
- **Estandarización:** Todas las tareas se realizan de la misma manera, manteniendo consistencia.

En resumen, las macros son una herramienta poderosa que transforma Excel de una hoja de cálculo común en una herramienta de automatización efectiva.

Tipos de Macros en Excel

Macros grabadas

Las macros grabadas son las más sencillas de crear. Excel registra cada acción que realizas (como escribir, copiar, pegar o formatear) y la guarda como un script que puede repetirse tantas veces como quieras.

Ventajas:

- No requiere conocimientos de programación.
- Útil para tareas repetitivas simples.

Limitaciones:

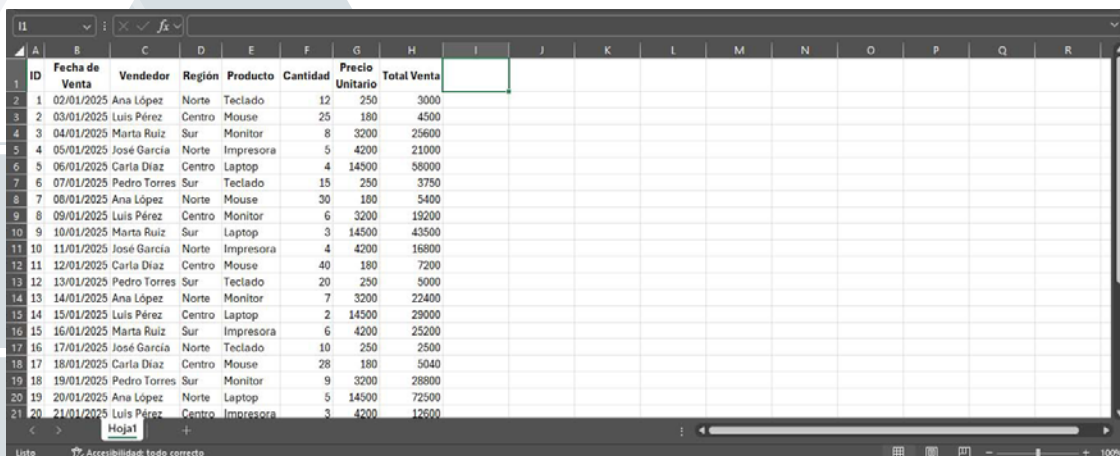
- Solo repite exactamente lo que grabaste; si algo cambia en la estructura de la hoja, puede fallar.
- No permite decisiones condicionales (por ejemplo, “si la celda A1 es mayor que 10, hacer X”).

Cómo grabar una macro paso a paso

Grabar una macro es la manera más fácil de empezar a automatizar tareas en Excel. Sigue estos pasos:

Paso 1: Preparar la hoja de cálculo

Antes de grabar, asegúrate de que tu hoja de Excel esté lista. Identifica exactamente qué acciones quieres automatizar: formato, cálculos, ordenamiento de datos, etc.

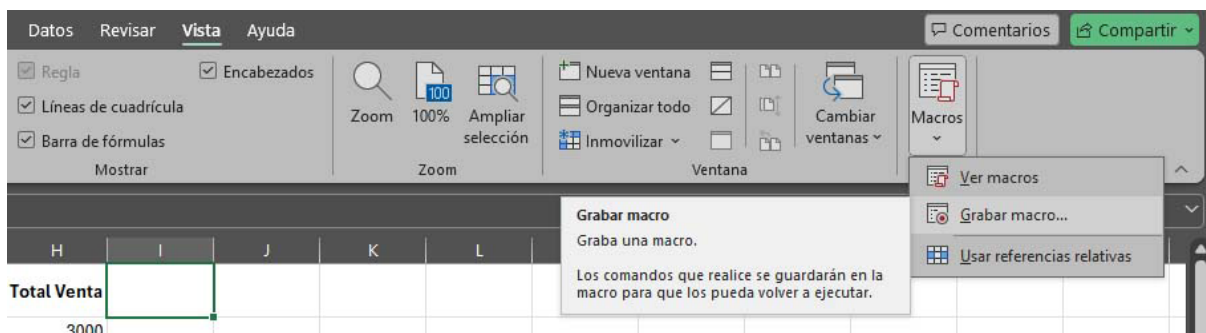


The screenshot shows an Excel spreadsheet with a table of sales data. The table has 8 columns: ID, Fecha de Venta, Vendedor, Región, Producto, Cantidad, Precio Unitario, and Total Venta. The data is organized into 21 rows, with the first row being the header. The 'Total Venta' column contains calculated values for each row.

ID	Fecha de Venta	Vendedor	Región	Producto	Cantidad	Precio Unitario	Total Venta
1	02/01/2025	Ana López	Norte	Teclado	12	250	3000
2	03/01/2025	Luis Pérez	Centro	Mouse	25	180	4500
3	04/01/2025	Marta Ruiz	Sur	Monitor	8	3200	25600
4	05/01/2025	José García	Norte	Impresora	5	4200	21000
5	06/01/2025	Carla Díaz	Centro	Laptop	4	14500	58000
6	07/01/2025	Pedro Torres	Sur	Teclado	15	250	3750
7	08/01/2025	Ana López	Norte	Mouse	30	180	5400
8	09/01/2025	Luis Pérez	Centro	Monitor	6	3200	19200
9	10/01/2025	Marta Ruiz	Sur	Laptop	3	14500	43500
10	11/01/2025	José García	Norte	Impresora	4	4200	16800
11	12/01/2025	Carla Díaz	Centro	Mouse	40	180	7200
12	13/01/2025	Pedro Torres	Sur	Teclado	20	250	5000
13	14/01/2025	Ana López	Norte	Monitor	7	3200	22400
14	15/01/2025	Luis Pérez	Centro	Laptop	2	14500	29000
15	16/01/2025	Marta Ruiz	Sur	Impresora	6	4200	25200
16	17/01/2025	José García	Norte	Teclado	10	250	2500
17	18/01/2025	Carla Díaz	Centro	Mouse	28	180	5040
18	19/01/2025	Pedro Torres	Sur	Monitor	9	3200	28800
19	20/01/2025	Ana López	Norte	Laptop	5	14500	72500
20	21/01/2025	Luis Pérez	Centro	Impresora	3	4200	12600

Paso 2: Iniciar la grabación

- Ve a Vista → Macros → Grabar Macro.
- Asigna un nombre descriptivo a la macro. Evita espacios; usa guiones bajos si es necesario.
- Selecciona dónde guardar la macro:
 - **Este libro:** la macro solo funcionará en el archivo actual.
 - **Nuevo libro:** se creará un libro nuevo con la macro.
 - **Libro de macros personal:** disponible en todos los libros de Excel en tu computadora.



Paso 3: Ejecutar las acciones

Ahora realiza todas las acciones que quieres que la macro repita. Por ejemplo:

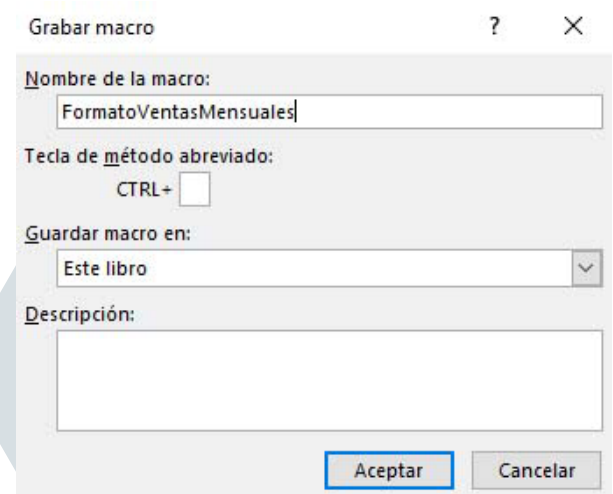
- Dar formato a celdas (color, fuente, bordes)
- Ordenar datos
- Aplicar filtros
- Copiar y pegar datos

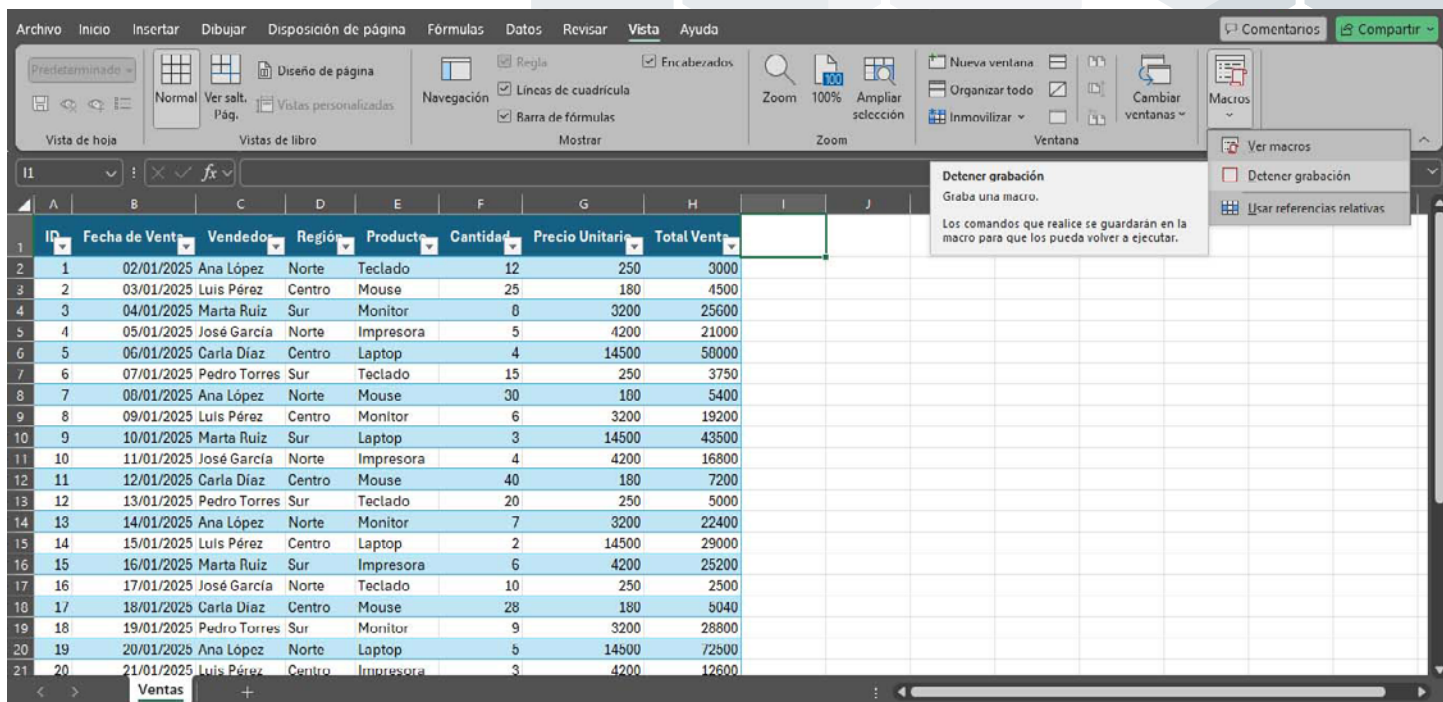
Paso 4: Detener la grabación

Cuando termines, ve a Vista → Macros → Detener grabación.

Tu macro ya está lista y puedes ejecutarla cada vez que la necesites.

Tip: Usa un nombre claro y descriptivo, como **FormatoVentasMensuales**, para que sea fácil de recordar.





Macros con VBA (Visual Basic for Applications)

Las macros con VBA son más avanzadas y te permiten crear scripts que incluyen lógica, condiciones y loops. A diferencia de las macros grabadas, estas no solo repiten pasos, sino que pueden “pensar” un poco: tomar decisiones, recorrer listas, calcular resultados y más.

Ventajas:

- Mayor flexibilidad y control.
- Permite automatizar procesos complejos.
- Puedes reutilizar partes de código en distintas hojas o libros de Excel.

Limitaciones:

- Requiere aprender un poco de programación básica en VBA.
- Un error en el código puede hacer que la macro no funcione o genere errores.

Cómo habilitar la pestaña Programador en Excel 365

Antes de empezar a grabar o crear macros, asegúrate de tener visible la pestaña Programador, ya que desde ahí podrás acceder a todas las herramientas de automatización y VBA.

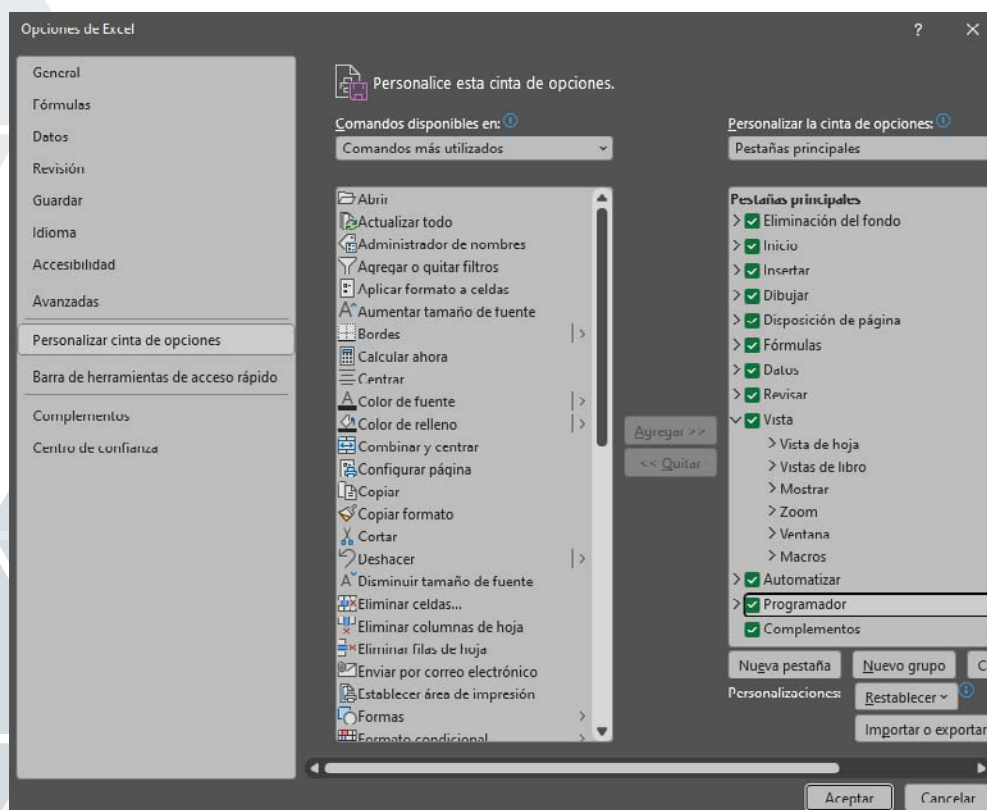
Para activarla:

1. Ve a Archivo → Opciones → Personalizar cinta de opciones.
2. En la lista de pestañas principales, marca la casilla Programador.
3. Haz clic en Aceptar.

Método alternativo:

También puedes hacerlo más rápido dando clic derecho en cualquier pestaña de la cinta de opciones, selecciona Personalizar la cinta de opciones, marca la casilla Programador y confirma con Aceptar.

Una vez habilitada, verás la pestaña Programador en la parte superior de Excel, desde donde podrás grabar macros, abrir el editor de VBA y administrar tus scripts.



Comandos y funciones más importantes de VBA

Visual Basic for Applications (VBA) tiene cientos de comandos, pero algunos son esenciales para automatizar tareas en Excel. Aquí tienes una tabla con los más utilizados y su explicación:

Comando / Función	Descripción	Ejemplo práctico aplicado al reporte de ventas
Range	Selecciona un rango de celdas	Range(«A1:D20»).Value = 0 - Limpia los valores antes de consolidar datos
Cells	Selecciona celdas por número de fila y columna	Cells(1,1).Value = «Ventas Semana 39» - Escribe encabezado
If...Then...Else	Permite condicionales	If Cells(i,3).Value > 1000 Then Cells(i,3).Interior.Color = RGB(255,0,0) - Resalta ventas mayores a 1000
For...Next	Bucle para repetir acciones varias veces	For i=2 To 20: Cells(i,4).Value = Cells(i,2).Value * 0.1: Next i - Calcula comisiones
Do While...Loop	Bucle que continúa mientras se cumpla una condición	i = 2: Do While Cells(i,1).Value<>»» : i = i + 1 : Loop - Recorre todas las filas con datos
Sub	Define una macro o procedimiento	Sub CrearReporteVentas() - Macro principal que ejecuta todas las tareas
Function	Define una función personalizada	Function CalcularIVA(x) : CalcularIVA = x*0.16 : End Function - Calcula IVA automáticamente
MsgBox	Muestra un mensaje al usuario	MsgBox «Reporte generado con éxito» - Notificación al finalizar macro
InputBox	Solicita un valor al usuario	MontoFiltro = InputBox(«Ingrese monto mínimo de ventas») - Filtra ventas según valor ingresado
Worksheets / Sheets	Hace referencia a hojas de cálculo	Worksheets(«Ventas»).Activate - Selecciona la hoja de origen de datos
Cells.Interior.Color	Cambia color de fondo de celdas	Cells(1,1).Interior.Color = RGB(0, 176, 240) - Resalta encabezados en azul

Copy / Paste	Copiar y pegar celdas	Range(«A2:D20»).Copy Worksheets(«Resumen»). Range(«A1») - Copia los datos al resumen
Sort	Ordena rangos	Range(«A1:D20»).Sort Key1:=Range(«C1»), Order1:=xlDescending - Ordena por ventas de mayor a menor
Filter	Aplica filtros automáticos	Range(«A1:D100»).AutoFilter Field:=3, Criteria1:>=>1000" - Filtrar ventas mayores a 1000
Add (hoja nueva)	Crea nueva hoja	Worksheets.Add.Name = «Resumen» - Crea hoja donde se consolidarán datos
Delete	Elimina elementos	Worksheets(«HojaTemp»).Delete - Elimina hoja temporal usada en cálculo
Charts.Add	Crea un gráfico	ActiveSheet.Shapes. AddChart2(251, xlColumnClustered, 300, 50, 400, 250).Chart.SetSourceData Source:=Range(«A1:D20») - Genera gráfico de ventas
SaveAs	Guardar archivo	ActiveWorkbook.SaveAs «C:\ Reportes\Ventas_Semana39. xlsx» - Guarda el reporte automáticamente
Offset	Se mueve un número de filas y columnas desde una celda	Cells(2,1).Offset(0,1).Value = «Nombre» - Inserta datos al lado de la celda seleccionada
Resize	Cambia tamaño de un rango seleccionado	Range(«A1»).Resize(10,4).Interior. Color = RGB(242, 242, 242) - Aplica formato a un rango de 10x4

Estos comandos permiten cubrir la mayoría de las tareas comunes: formateo, cálculos, filtros, consolidación de datos y generación de reportes. Con ellos, cualquier macro puede automatizar procesos complejos de manera eficiente.

Declaración de Variables en VBA y Tipos de Datos Más Utilizados

En VBA, Dim (abreviatura de Dimension) se utiliza para declarar variables. Una variable es un espacio en la memoria donde puedes guardar datos temporales que usarás dentro de tu macro, como números, textos o rangos. Declarar variables ayuda a que tu código sea más claro, eficiente y fácil de mantener.

Sintaxis básica:

Dim nombreVariable As Tipo

- nombreVariable: nombre que eliges para la variable.
- Tipo: tipo de datos que contendrá la variable, como Integer, Long, Double, String, Boolean, Range, Worksheet, etc.

Declaración / Comando	Tipo de dato	Descripción	Ejemplo práctico en macros de Excel
Integer	Número entero (-32,768 a 32,767)	Para contar o indexar filas y columnas	Dim i As Integer: For i = 1 To 10: Cells(i,1).Value = i: Next i
Long	Número entero grande (-2,147,483,648 a 2,147,483,647)	Para contar grandes cantidades de filas o cálculos grandes	Dim totalFilas As Long: totalFilas = 100000
Double	Número decimal	Para cálculos con decimales, totales y promedios	Dim totalVentas As Double: totalVentas = Cells(2,3).Value + Cells(3,3).Value
String	Texto / cadena	Para almacenar nombres, mensajes o cualquier texto	Dim cliente As String: cliente = «Juan Pérez»
Boolean	Verdadero/Falso	Para condiciones lógicas o banderas	Dim esMayor As Boolean: esMayor = Cells(2,3).Value > 1000
Variant	Cualquier tipo	Variable flexible que puede almacenar cualquier tipo de dato	Dim valor As Variant: valor = Cells(2,3).Value
Date	Fecha / hora	Para almacenar fechas y horas	Dim fechaVenta As Date: fechaVenta = Now
Object	Objeto	Para referirse a objetos como hojas, rangos, gráficos	Dim ws As Worksheet: Set ws = Worksheets(«Ventas»)

Range	Rango de celdas	Para referirse a un rango de celdas específico	Dim celda As Range: Set celda = Range(«A1»)
Worksheet	Hoja de cálculo	Para declarar una hoja específica	Dim wsResumen As Worksheet: Set wsResumen = Worksheets.Add

Consejos prácticos sobre variables en VBA

- Siempre declara tus variables: Esto evita errores y hace que el código sea más fácil de leer.
- Usa nombres claros y descriptivos: Por ejemplo, totalVentas en lugar de x.
- Elige el tipo de dato correcto: Usar Integer para grandes cantidades puede generar errores; para hojas con muchos datos, usa Long.
- Variables Object requieren Set: Cuando asignas un rango, hoja u objeto, usa Set para referenciarlo correctamente.
- Evita Variant salvo que sea necesario: Aunque flexible, ocupa más memoria y puede reducir el rendimiento si se abusa.

Ejemplo práctico sencillo en VBA:

Sub ResaltarMayor100()

Dim celda As Range

For Each celda In Selection

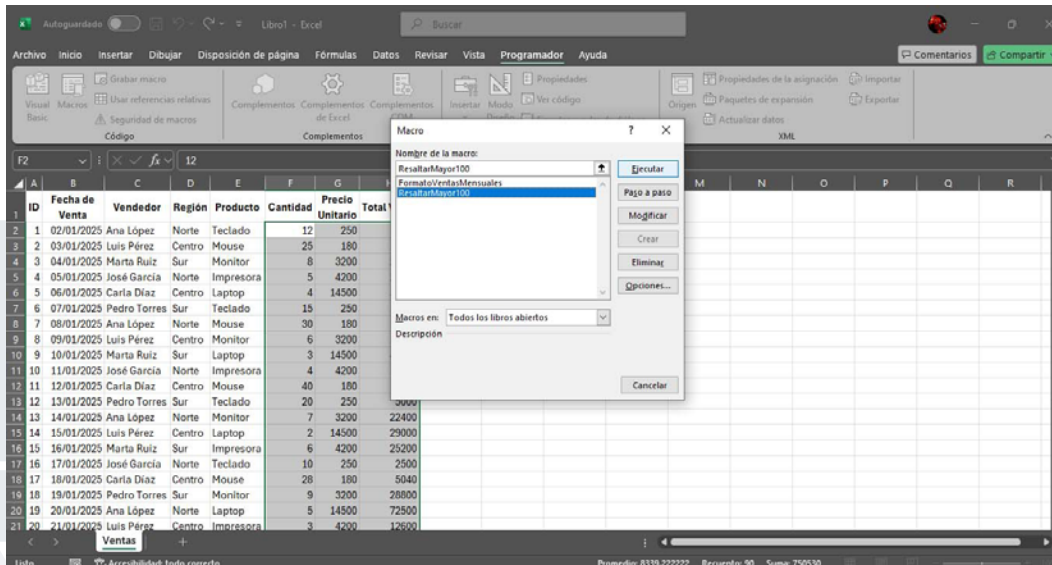
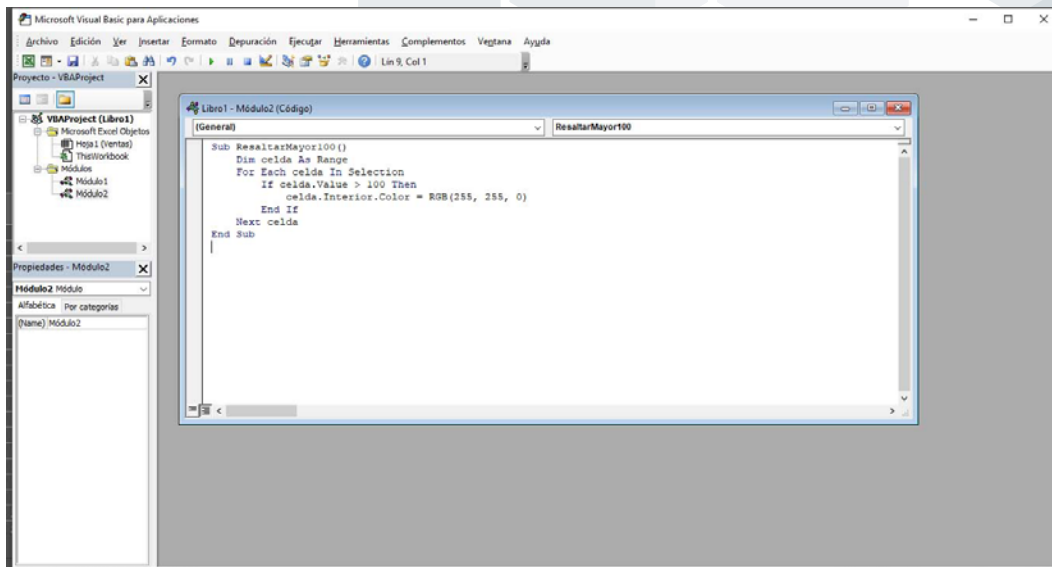
If celda.Value > 100 Then

celda.Interior.Color = RGB(255, 255, 0)

End If

Next celda

End Sub



ID	Fecha de Venta	Vendedor	Región	Producto	Cantidad	Precio Unitario	Total Venta
1	02/01/2025	Ana López	Norte	Teclado	12	250	3000
2	03/01/2025	Luis Pérez	Centro	Mouse	25	180	4500
3	04/01/2025	Marta Ruiz	Sur	Monitor	8	3200	25600
4	05/01/2025	José García	Norte	Impresora	5	4200	21000
5	06/01/2025	Carla Díaz	Centro	Laptop	4	14500	58000
6	07/01/2025	Pedro Torres	Sur	Teclado	15	250	3750
7	08/01/2025	Ana López	Norte	Mouse	30	180	5400
8	09/01/2025	Luis Pérez	Centro	Monitor	6	3200	19200
9	10/01/2025	Marta Ruiz	Sur	Laptop	3	14500	43500
10	11/01/2025	José García	Norte	Impresora	4	4200	16800
11	12/01/2025	Carla Díaz	Centro	Mouse	40	180	7200
12	13/01/2025	Pedro Torres	Sur	Teclado	20	250	5000
13	14/01/2025	Ana López	Norte	Monitor	7	3200	22400
14	15/01/2025	Luis Pérez	Centro	Laptop	2	14500	29000
15	16/01/2025	Marta Ruiz	Sur	Impresora	6	4200	25200
16	17/01/2025	José García	Norte	Teclado	10	250	2500
17	18/01/2025	Carla Díaz	Centro	Mouse	28	180	5040
18	19/01/2025	Pedro Torres	Sur	Monitor	9	3200	28800
19	20/01/2025	Ana López	Norte	Laptop	5	14500	72500
20	21/01/2025	Luis Pérez	Centro	Impresora	3	4200	12600

Explicación:

- Este código recorre todas las celdas seleccionadas y resalta en amarillo aquellas con valores mayores a 100.
- Para usarlo, presiona ALT + F11 → Insertar → Módulo → pega el código
- Regresa a la hoja, selecciona el rango que quieres resaltar y ve a Programador - Macros - Selecciona la Macro “ResaltarMayor100” y da clic en Ejecutar.

Ejemplo de declaración de variables:

Dim i As Integer ‘ Variable para contar filas

Dim totalVentas As Double ‘ Variable para sumar ventas

Dim cliente As String ‘ Variable para guardar nombre de cliente

Dim ws As Worksheet ‘ Variable para referirse a una hoja de cálculo

Dim celda As Range ‘ Variable para recorrer celdas

Uso en un bucle:

Sub SumarTotalVentas()

Dim i As Long

Dim totalVentas As Double

Dim ws As Worksheet

Dim ultimaFila As Long

‘ Referencia a la hoja correcta

Set ws = ThisWorkbook.Worksheets(«Ventas»)

totalVentas = 0

‘ Encontrar la última fila con datos en la columna H

ultimaFila = ws.Cells(ws.Rows.Count, 8).End(xlUp).Row

‘ Sumar los valores numéricos de la columna H

For i = 2 To ultimaFila

If IsNumeric(ws.Cells(i, 8).Value) Then

totalVentas = totalVentas + ws.Cells(i, 8).Value

End If

Next i

‘ Poner el total en la siguiente fila disponible de la columna H

ws.Cells(ultimaFila + 1, 8).Value = totalVentas

‘ Opcional: mostrar mensaje con el total

MsgBox «El total de ventas es: » & totalVentas

End Sub

```
Libro1 - Módulo1 (Código)
(SumarTotalVentas)

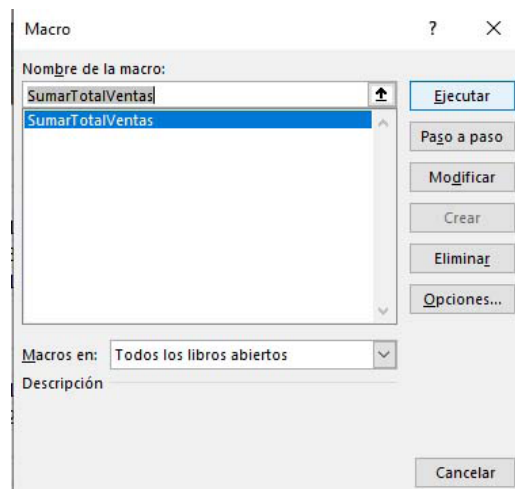
Sub SumarTotalVentas()
    Dim i As Long
    Dim totalVentas As Double
    Dim ws As Worksheet
    Dim ultimaFila As Long

    ' Referencia a la hoja correcta
    Set ws = ThisWorkbook.Worksheets("Ventas")

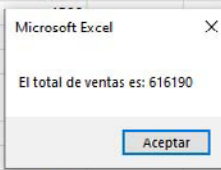
    totalVentas = 0

    ' Encontrar la última fila con datos en la columna H
    ultimaFila = ws.Cells(ws.Rows.Count, 8).End(xlUp).Row

    ' Sumar los valores numéricos de la columna H
    For i = 2 To ultimaFila
        If IsNumeric(ws.Cells(i, 8).Value) Then
            totalVentas = totalVentas + ws.Cells(i, 8).Value
        End If
    Next i
End Sub
```



H32	A	B	C	D	E	F	G	H	I	J
19	18	19/01/2025	Pedro Torres	Sur	Monitor	9	3200	28800		
20	19	20/01/2025	Ana López	Norte	Laptop	5	14500	72500		
21	20	21/01/2025	Luis Pérez	Centro	Impresora	3	4200	12600		
22	21	22/01/2025	Marta Ruiz	Sur	Teclado	18	250			
23	22	23/01/2025	José García	Norte	Mouse	35	180			
24	23	24/01/2025	Carla Díaz	Centro	Monitor	10	3200			
25	24	25/01/2025	Pedro Torres	Sur	Laptop	4	14500			
26	25	26/01/2025	Ana López	Norte	Impresora	2	4200			
27	26	27/01/2025	Luis Pérez	Centro	Teclado	14	250			
28	27	28/01/2025	Marta Ruiz	Sur	Mouse	20	180	3600		
29	28	29/01/2025	José García	Norte	Monitor	5	3200	16000		
30	29	30/01/2025	Carla Díaz	Centro	Laptop	3	14500	43500		
31	30	31/01/2025	Pedro Torres	Sur	Impresora	7	4200	29400		



Aquí i controla el bucle y totalVentas acumula la suma de las ventas de la columna H ('Total Venta'). Luego, el resultado se coloca en la siguiente fila vacía de esa misma columna.

Diferencia entre macros grabadas y macros con VBA

Característica	Macro Grabada	Macro con VBA
Conocimiento de programación	No requiere	Básico a intermedio
Flexibilidad	Limitada	Alta
Capacidad de decisión	No permite condiciones	Permite condiciones, loops y cálculos
Uso recomendado	Tareas simples	Tareas complejas o repetitivas avanzadas
Facilidad de edición	Baja, depende de la grabación	Alta, se puede modificar el código directamente

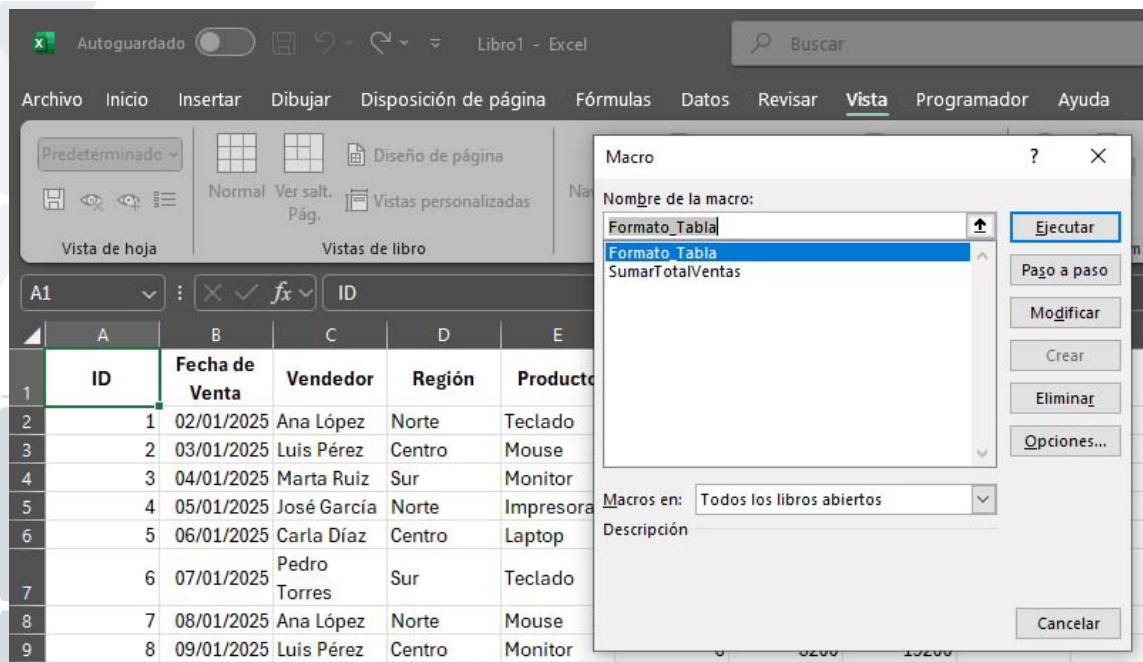
Esta tabla te ayuda a decidir cuál tipo de macro conviene según tus necesidades.

Cómo ejecutar una macro

Existen varias formas de ejecutar macros en Excel:

1. Desde la pestaña Vista

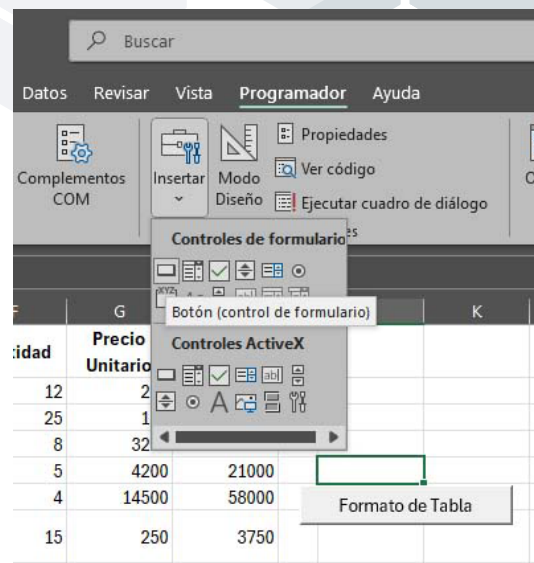
- Ve a Vista → Macros → Ver Macros
- Selecciona la macro que quieres ejecutar y haz clic en Ejecutar



2. Con un botón en la hoja

Puedes insertar un botón y asignarle la macro:

1. Ve a Programador → Insertar → Botón (Control de formulario)
2. Dibuja el botón en la hoja
3. Asigna la macro al botón
4. Cada vez que hagas clic, la macro se ejecutará



3. Con un atajo de teclado

Al grabar la macro, puedes asignarle un atajo de teclado (por ejemplo, Ctrl + Shift + M) para ejecutarla rápidamente sin usar el mouse.

Macros para automatizar tareas comunes

Existen tareas que se repiten constantemente en Excel y que son ideales para automatizar con macros. Entre las más utilizadas están:

1. Formateo de celdas y tablas

- Cambiar color de encabezados, filas alternas o celdas específicas según valor.
- Ajustar automáticamente el ancho de columnas y filas según el contenido.
- Aplicar estilos predefinidos o personalizados a tablas para uniformidad visual.



Ejemplo: Una macro puede aplicar formato a todas tus tablas de ventas: encabezados en azul, filas alternas en azul claro y números con formato de moneda.

2. Consolidación de datos

- Copiar rangos de varias hojas o archivos.
- Combinar datos en una sola hoja de resumen.
- Eliminar duplicados automáticamente.



Ejemplo: Si recibes reportes diarios de distintos vendedores, una macro puede combinar todos los datos en un solo archivo semanal, listo para análisis.

3. Reportes automáticos

- Generar tablas dinámicas automáticamente.
- Crear gráficos basados en datos consolidados.
- Insertar encabezados, pies de página y fechas de actualización.



Ejemplo: Una macro puede crear un reporte de ventas mensual que incluya totales por región, gráfico de barras de los productos más vendidos y guardar el archivo en PDF automáticamente.

4. Ordenamiento y filtrado de datos

- Ordenar listas por valores numéricos o alfabéticos.
- Aplicar filtros automáticos según criterios predefinidos.
- Resaltar datos que cumplan condiciones específicas (mayores que, menores que, duplicados, vacíos).



Ejemplo: Filtrar clientes con ventas mayores a \$10,000 y resaltar aquellos que no han realizado compras en los últimos 3 meses.

5. Cálculos automáticos y fórmulas

- Aplicar fórmulas a rangos completos.
- Copiar resultados a otras hojas o tablas.
- Actualizar cálculos automáticamente cuando se agregan nuevos datos.



Ejemplo: Calcular comisiones de ventas automáticamente y sumar el total por vendedor sin tener que escribir las fórmulas manualmente.

6. Exportación e importación de datos

- Guardar hojas o rangos como archivos CSV, TXT o PDF automáticamente.
- Importar datos desde otros archivos y formatearlos al mismo tiempo.
- Enviar reportes por correo electrónico directamente desde Excel.



Ejemplo: Una macro puede exportar el reporte de ventas a PDF, guardarlo en la carpeta de informes y enviar un correo al gerente con el archivo adjunto.

Buenas prácticas al trabajar con macros

1. Nombrar macros claramente: Usa nombres descriptivos y sin espacios.
2. Guardar antes de ejecutar: Algunas macros modifican muchos datos; guarda una copia por seguridad.
3. Documentar tus macros: Escribe comentarios dentro del código para recordar qué hace cada sección.
4. Probar en hojas de ejemplo: Antes de aplicar en tu archivo principal, prueba la macro en un archivo de prueba.
5. Evitar macros innecesarias: Solo automatiza tareas repetitivas que realmente ahorren tiempo.

Ejemplo completo de macro automatizada

Supongamos que quieres un reporte semanal que:

- Copie los datos de la hoja «Ventas»
- Pegue en una nueva hoja llamada «Resumen»
- Resalte los valores mayores a 5500

Código en VBA:

Sub ReporteSemanal()

Dim wsVentas As Worksheet

Dim wsResumen As Worksheet

Dim celda As Range

Set wsVentas = Worksheets(«Ventas»)

Set wsResumen = Worksheets.Add

wsResumen.Name = «Resumen»

wsVentas.Range(«A1:H100»).Copy wsResumen.Range(«A1»)

*For Each celda In wsResumen.
Range(«H2:H100»)*

If celda.Value > 5500 Then

*celda.Interior.Color = RGB(80,
200, 120)*

End If

Next celda

End Sub

ID	Fecha de Venta	Vendedor	Región	Producto	Cantidad	Precio Unitario	Total Venta
1	02/01/2025	Ana López	Norte	Teclado	12	\$ 250.00	\$ 3,000.00
2	03/01/2025	Luis Pérez	Centro	Mouse	25	\$ 180.00	\$ 4,500.00
3	04/01/2025	Marta Ruiz	Sur	Monitor	8	\$ 3,200.00	\$ 25,600.00
4	05/01/2025	José García	Norte	Impresora	5	\$ 4,200.00	\$ 21,000.00
5	06/01/2025	Carla Díaz	Centro	Laptop	4	\$ 14,500.00	\$ 58,000.00
6	07/01/2025	Pedro Torres	Sur	Teclado	15	\$ 250.00	\$ 3,750.00
7	08/01/2025	Ana López	Norte	Mouse	30	\$ 180.00	\$ 5,400.00
8	09/01/2025	Luis Pérez	Centro	Monitor	6	\$ 3,200.00	\$ 19,200.00
9	10/01/2025	Marta Ruiz	Sur	Laptop	3	\$ 14,500.00	\$ 43,500.00
10	11/01/2025	José García	Norte	Impresora	4	\$ 4,200.00	\$ 16,800.00
11	12/01/2025	Carla Díaz	Centro	Mouse	40	\$ 180.00	\$ 7,200.00
12	13/01/2025	Pedro Torres	Sur	Teclado	20	\$ 250.00	\$ 5,000.00
13	14/01/2025	Ana López	Norte	Monitor	7	\$ 3,200.00	\$ 22,400.00
14	15/01/2025	Luis Pérez	Centro	Laptop	2	\$ 14,500.00	\$ 29,000.00
15	16/01/2025	Marta Ruiz	Sur	Impresora	6	\$ 4,200.00	\$ 25,200.00
16	17/01/2025	José García	Norte	Teclado	10	\$ 250.00	\$ 2,500.00
17	18/01/2025	Carla Díaz	Centro	Mouse	28	\$ 180.00	\$ 5,040.00
18	19/01/2025	Pedro Torres	Sur	Monitor	9	\$ 3,200.00	\$ 28,800.00
19	20/01/2025	Ana López	Norte	Laptop	5	\$ 14,500.00	\$ 72,500.00
20	21/01/2025	Luis Pérez	Centro	Impresora	3	\$ 4,200.00	\$ 12,600.00
21	22/01/2025	Marta Ruiz	Sur	Teclado	18	\$ 250.00	\$ 4,500.00
22	23/01/2025	José García	Norte	Mouse	35	\$ 180.00	\$ 6,300.00

Explicación paso a paso:

Declaración de variables

Dim wsVentas As Worksheet

Dim wsResumen As Worksheet

Dim celda As Range

- wsVentas: variable que representa la hoja Ventas (origen de los datos).
- wsResumen: variable que representa la hoja Resumen (donde se va a copiar la información).
- celda: variable que se usará para recorrer las celdas de la columna H.

Asignación de hojas

Set wsVentas = Worksheets(«Ventas»)

Set wsResumen = Worksheets.Add

wsResumen.Name = «Resumen»

- Se asigna la hoja Ventas a wsVentas.
- Se crea una nueva hoja y se asigna a wsResumen.
- Se le cambia el nombre de la hoja nueva a Resumen.

Copiar los datos

wsVentas.Range(«A1:H100»).Copy wsResumen.Range(«A1»)

- Copia el rango A1:H100 de la hoja Ventas.

- Lo pega empezando en A1 de la hoja Resumen.
- Esto incluye columnas como ID, Fecha, Vendedor, Región y Total Venta (columna H).

Nota: Actualmente copia hasta la fila 100, aunque tu tabla tenga menos o más filas.

Resaltar valores mayores a 5500

For Each celda In wsResumen.Range(«H2:H100»)

If celda.Value > 5500 Then

celda.Interior.Color = RGB(80, 200, 120)

End If

Next celda

- Recorre cada celda de la columna H (Total Venta) desde la fila 2 hasta la 100.
- Si el valor de la celda es mayor a 5500, cambia el color de fondo a RGB(80,200,120) (verde).
- Esto permite resaltar automáticamente las ventas más altas en la hoja Resumen.

Resumen general

1. Crea un resumen de los datos de la hoja Ventas en una nueva hoja.
2. Copia todas las columnas importantes (A:H).
3. Resalta automáticamente los valores mayores a 5500 en la columna Total Venta (H).

Con esta macro, un proceso que tomaba varios minutos ahora se hace automáticamente en segundos.

Conclusión

Aprender a trabajar con Macros en Excel es una inversión que mejora tu productividad y reduce errores. Con este artículo aprendiste:



- Qué son las macros y por qué son útiles.
- Los tipos de macros: grabadas y con VBA.
- Cómo grabar, ejecutar y asignar macros a botones o atajos.
- Ejemplos prácticos de automatización de tareas comunes.
- Buenas prácticas para trabajar con macros y mantener tus datos seguros.
- Cómo empezar a aprender VBA para tareas más avanzadas.

Si quieres dominar la automatización de tu trabajo y llevar tus habilidades al siguiente nivel, te invitamos a inscribirte en nuestro [Curso de Macros en Excel](#), donde aprenderás paso a paso a automatizar tus tareas y ahorrar horas de trabajo cada semana.

FAQ: Domina las Macros en Excel

1. ¿Qué es una macro en Excel?

Una macro es una serie de instrucciones que Excel ejecuta automáticamente para realizar tareas repetitivas. Es como un “botón mágico” que puede copiar datos, aplicar formatos, crear gráficos y más, con un solo clic.

2. ¿Cuál es la diferencia entre una macro grabada y una macro con VBA?

Las macros grabadas registran acciones exactas que realizas en Excel y se repiten tal cual, sin lógica ni condiciones. Las macros con VBA permiten agregar decisiones, bucles y cálculos, ofreciendo más flexibilidad y control sobre tus procesos.

3. ¿Necesito saber programar para usar macros?

No. Puedes empezar usando macros grabadas, que no requieren conocimientos de programación. Para tareas más avanzadas o automatizaciones complejas, aprender VBA básico será muy útil.

4. ¿Cómo puedo grabar una macro en Excel paso a paso?

- Prepara tu hoja y define las acciones a automatizar.
- Ve a Vista → Macros → Grabar Macro, asigna un nombre y elige dónde guardarla.
- Realiza las acciones que quieres automatizar.
- Detén la grabación en Vista → Macros → Detener grabación. Tu macro estará lista para ejecutarse cuando quieras.

5. ¿Qué tipos de tareas puedo automatizar con macros?

Las macros son ideales para tareas repetitivas como:

- Formateo de celdas y tablas.
- Consolidación de datos de varias hojas.
- Creación automática de reportes y gráficos.
- Ordenamiento y filtrado de información.
- Aplicación de fórmulas y cálculos automáticos.
- Exportación o importación de datos.

6. ¿Cuál es la diferencia principal entre una macro grabada y una macro con VBA?

Una macro grabada registra exactamente las acciones que realizas en Excel y las repite tal cual, sin lógica ni condiciones.

Una macro con VBA permite agregar decisiones, bucles y cálculos, ofreciendo más flexibilidad y control sobre tus procesos.

7. ¿Cómo ejecuto una macro en Excel?

- Desde la pestaña Vista → Macros → Ver Macros, selecciona la macro y haz clic en Ejecutar.
- Asignando la macro a un botón en la hoja (Desarrollador → Insertar → Botón).
- Usando un atajo de teclado que hayas configurado al grabar la macro (por ejemplo, Ctrl + Shift + M).

8. ¿Cuáles son las buenas prácticas al usar macros?

- Nombrar las macros de forma clara y descriptiva.
- Guardar el archivo antes de ejecutar la macro para evitar pérdidas.
- Documentar el código con comentarios.
- Probar la macro en un archivo de ejemplo antes de aplicarla en archivos importantes.
- Evitar automatizar tareas innecesarias que no ahorren tiempo.