



En Línea

#DesarrollaTuNivelTop



Macros



SEMANA 26



Índice

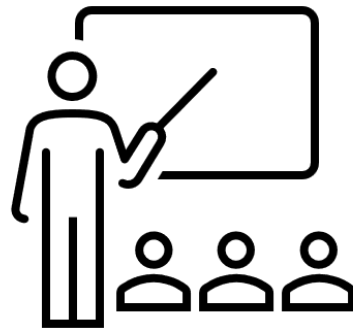
- ¿Qué es una macro?
- Grabadora de Macros
- Crear tablas desde VBA
- ¿Qué es un formulario en VBA?
- Ejemplo de formulario
- Ciclo Do While
- Ciclo Do Until
- Ciclo For...Next
- Ciclo While...Wend en VBA
- Comparación de ciclos en VBA



Objetivo particular de la lección

Dominar la automatización de tareas en Excel a través del uso de macros y Visual Basic for Applications (VBA). El aprendizaje abarca la capacidad de generar y manipular tablas mediante código, así como la creación de formularios simples para una eficiente captura de datos.

Así mismo el participante tendrá comprensión y aplicación práctica de estructuras de control de flujo, tales como los ciclos Do While, Do Until, For, entre otros, para desarrollar soluciones dinámicas y personalizadas en Excel.



¿Qué es una macro?

Una macro es un conjunto de instrucciones que automatiza tareas repetitivas en Excel. Las macros pueden realizar procesos como copiar datos, aplicar formato, crear reportes y más, con solo un clic.

Beneficios de usarla:

1. Se escribe en VBA o se graba
2. Automatiza acciones manuales
3. Mejora productividad y reduce errores

Sub Saludo()

**MsgBox "Bienvenido a la
automatización con Excel"**

End Sub

Grabadora de Macros

La grabadora de macros permite crear una macro sin necesidad de escribir código. Excel graba cada acción del usuario y la convierte en código VBA automáticamente.

- Ideal para tareas sencillas
- Accesible desde la pestaña Desarrollador
- El código grabado se puede modificar desde el Editor de VBA

Ventaja:

útil para aprender la sintaxis de VBA a partir de acciones reales.

Crear tablas desde VBA

Las tablas (ListObjects) permiten manejar datos de forma estructurada y ofrecen filtros, ordenamiento y formato automático. Se pueden crear dinámicamente desde código VBA.

- Ayudan en el análisis de datos
- Se insertan sobre un rango de celdas
- Se les puede asignar nombre y aplicar formato

Sub CrearTabla()

```
ActiveSheet.ListObjects.Add(xlSrcRange, Range("A1:D10"), , xlYes).Name = "Ventas"  
End Sub
```

¿Qué es un formulario en VBA?

Los formularios (UserForms) son interfaces gráficas que permiten recopilar datos del usuario. Son ideales cuando se quiere controlar cómo se ingresa la información, validar campos o aplicar lógica personalizada.

- Se insertan desde el Editor de VBA
- Contienen controles: TextBox, Button, Label, etc.
- Permiten interacción sencilla con el usuario

Sub MostrarFormulario()

UserForm1.Show

End Sub

The image shows a screenshot of a Windows-style dialog box titled "UserForm1". The dialog box has a standard title bar with a close button (X) in the top right corner. The main area of the dialog box contains three text input fields, each preceded by a label: "Nombre:", "Teléfono:", and "E-mail:". Below these input fields are two buttons: "Aceptar" (Accept) on the left and "Cancelar" (Cancel) on the right. The dialog box is set against a background with a dotted grid pattern.

Ejemplo de formulario

El siguiente formulario permite ingresar nombre y edad, y guarda los datos en la hoja “Datos”.

- Se pueden validar los campos
- Se insertan dinámicamente en la siguiente fila vacía
- Ideal para bases de datos internas

Private Sub btnGuardar_Click()

Dim fila As Long

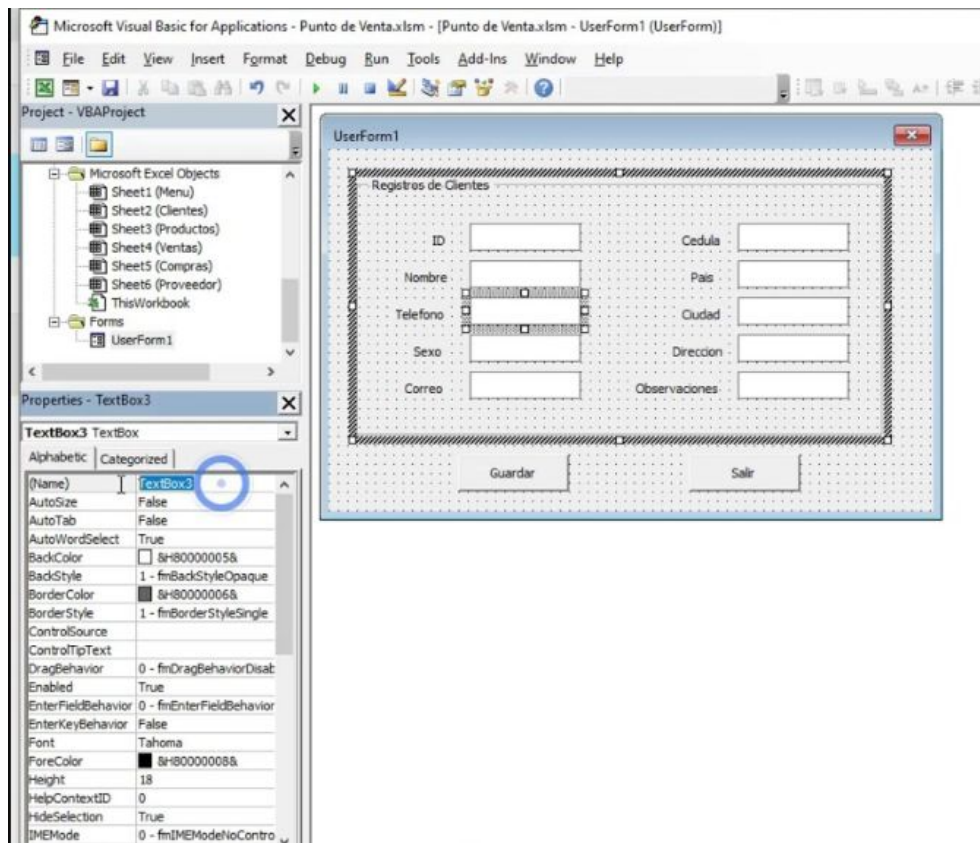
fila = Sheets("Datos").Cells(Rows.Count, 1).End(xlUp).Row + 1

Sheets("Datos").Cells(fila, 1).Value = txtNombre.Text

Sheets("Datos").Cells(fila, 2).Value = txtEdad.Text

Unload Me

End Sub





En Línea

#DesarrollaTuNivelTop



CONCLUIMOS LA PRIMER PARTE DE NUESTRA SESIÓN



SEMANA 26

CONTINUAMOS

Ciclo Do While

Definición: El ciclo Do While se ejecuta mientras la condición sea verdadera. Es útil cuando **no sabemos cuántas veces se repetirá la acción.**

- Se evalúa la condición antes de iniciar
- Útil cuando el número de repeticiones depende del contenido o estado de una celda

Dim i As Integer

i = 1

Do While Cells(i, 1).Value <> ""

Cells(i, 2).Value = Cells(i, 1).Value * 2

i = i + 1

Loop

Ciclo Do While

Cuándo usarlo: Cuando necesitas revisar una lista hasta encontrar una celda vacía.

Estructura
Iterativa

Expresión Lógica

Do While (**stock >= 10**)

cantP=cantP+1

Loop

Instrucciones
que se repetirán

```
(General)
Sub prueba_do()
Dim sum As Integer
Do While sum < 1000
    sum = sum + 50
Loop
MsgBox "alcanzó " & sum
End Sub
```

Ciclo Do Until

Definición: Este ciclo se ejecuta hasta que la condición sea verdadera, es decir, termina cuando se cumple la condición.

- Repite mientras la condición sea falsa
- Ideal para procesos con una condición de salida clara

Dim i As Integer

i = 1

Do Until Cells(i, 1).Value = ""

Cells(i, 2).Value = "Valor encontrado"

i = i + 1

Loop

Ciclo Do Until

Cuándo usarlo: Para recorrer hasta una fila vacía, o hasta encontrar cierto valor.

A
1
Nombre
2
Juan
3
Maria
4
Jesús
5
José
6
Ana
7
8
9
10



```
Sub Tutorial5c_Loops()  
i = 2  
Do Until Cells(i, 1) = ""  
    'Actividad repetitiva  
    i = i + 1  
Loop  
Ultima_fila = i - 1  
MsgBox ("Ultima fila: " & Ultima_fila)  
End Sub
```

- Estructura Do
- Actividad repetitiva
- Control del número de repetición

Ciclo For...Next

Definición: El ciclo For se usa cuando sabes cuántas veces se debe ejecutar la acción. Recorre un rango de valores definidos.

- Tiene un contador (i, j, etc.)
- Se puede controlar el incremento (Step)
- Muy útil para recorrer rangos fijos

Dim i As Integer

For i = 1 To 10

Cells(i, 1).Value = i ^ 2

Next i

Ciclo For...Next

Cuándo usarlo: Para recorrer exactamente 10 filas, 5 columnas, o repetir algo 3 veces.



(General)

```
Sub EjemploForNext()

For i = 1 To 5
    MsgBox ("i = " & i)
Next i

End Sub
```

	A	B	C	D	E	F	G
1	Valor	Grupo	Sub BucleIf()				
2	50	Positivo	Dim Celda As Range				
3	-10	Negativo	For Each Celda In Range("A2:A6")				
4	0	Cero	If Celda.Value > 0 Then				
5	1	Positivo	Celda.Offset(0, 1).Value = "Positivo"				
6	-13	Negativo	ElseIf Celda.Value < 0 Then				
7			Celda.Offset(0, 1).Value = "Negativo"				
8			Else				
9			Celda.Offset(0, 1).Value = "Cero"				
10			End If				
11			Next Celda				
			End Sub				

Ciclo While...Wend en VBA

Definición: El ciclo While...Wend ejecuta un bloque de código **mientras se cumpla una condición**. Es una estructura heredada y funcional, pero menos flexible que otros ciclos modernos como Do While.

- **Sintaxis básica:**

While condición

 ' Instrucciones a repetir

Wend

- **Ejemplo práctico:**

Dim i As Integer

i = 1

While i <= 5

 Cells(i, 1).Value = "Fila " & i

 i = i + 1

Wend

- **Ejemplo práctico:**

Dim i As Integer

i = 1

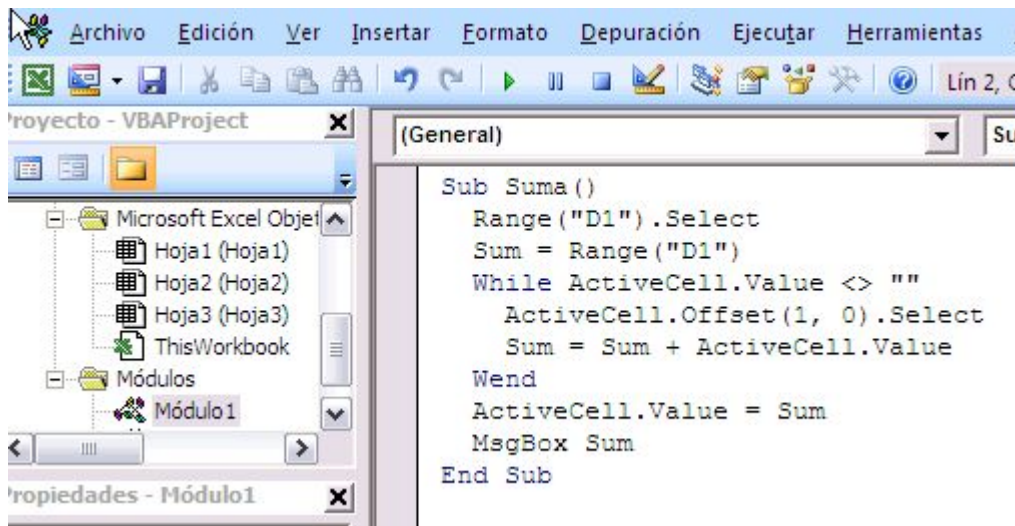
While i <= 5

Cells(i, 1).Value = "Fila " & i

i = i + 1

Wend

Este ciclo escribe "Fila 1" hasta "Fila 5"
en la columna A.



(General)

```
Sub WhileEx1()  
    Number = 1  
    Sum = 0  
    While Number <= 10  
        Sum = Sum + Number  
        Number = Number + 1  
        Debug.Print Sum  
    Wend  
End Sub
```

Consideraciones:

- No permite comandos como Exit While
- No es posible usar instrucciones como DoEvents dentro del ciclo
- Solo se recomienda para tareas simples y controladas

¿Usarlo o evitarlo?

- ✓ Bueno para ejemplos educativos
- ⚠ Evítalo en código complejo o profesional
- 👍 Si necesitas control total, mejor usar Do While...Loop

Comparación de ciclos en VBA

Ciclo	¿Cuándo usarlo?	Evalúa condición	Flexibilidad
Do While	Mientras se cumpla una condición dinámica	Antes del ciclo	Alta
Do Until	Hasta que se cumpla una condición	Antes del ciclo	Alta
For	Repeticiones con número fijo conocido	Antes del ciclo	Alta
While...Wend	Repeticiones simples con condición básica	Antes del ciclo	Baja

Recomendación final:

- Usa For para recorridos controlados
- Usa Do While o Do Until para condiciones variables
- Usa While...Wend solo en estructuras sencillas o didácticas

Artículos o e-books con referencias en formato APA.

Alexander, M., & Jelen, B. (2019). Excel 2019 Power Programming with VBA. Pearson Education.

Burg, S. (2020). Excel VBA in Easy Steps. In Easy Steps.

Coursera. (N.d.). *Business Analytics Specialization*. Recuperado el 17 de julio de 2025 de

<https://www.coursera.org/specializations/business-analytics>

Dalglish, D. (N.d.). *Contextures Excel Tips*. Contextures. Recuperado el 17 de julio de 2025 de

<https://www.contextures.com/>

Microsoft. (N.d.). *Office VBA reference for the Office client applications*. Microsoft Learn.

Recuperado el 17 de julio de 2025 de <https://learn.microsoft.com/en-us/office/vba/api/overview/>

MrExcel.com. (N.d.). *MrExcel.com: Excel Help & Forum*. Recuperado el 17 de julio de 2025 de

<https://www.mrexcel.com/>

Stack Overflow. (N.d.). *Stack Overflow*. Recuperado el 17 de julio de 2025 de

<https://stackoverflow.com/>

VBA Express. (N.d.). *VBA Express: Excel VBA Help Forum & Tutorials*. Recuperado el 17 de julio de

2025 de <https://www.vbaexpress.com/>

Walkenbach, J. (2023). Excel VBA Programming For Dummies. For Dummies.

Walkenbach, J. (2024). Excel Bible. John Wiley & Sons.



SEMANA 26 COMPLETADA
¡Felicidades!
Éxito en tu evaluación



© Top Learning Online, S.C., 2025.

Todos los derechos reservados. Prohibida su reproducción, difusión o uso sin autorización escrita.



En Línea

#DesarrollaTuNivelTop



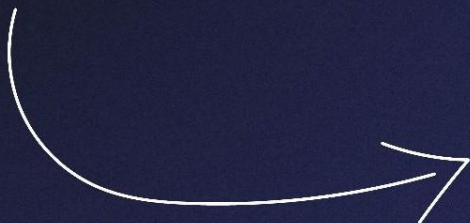
En Línea

#DesarrollaTuNivelTop

¿Quieres aprender más?

Susíbete a nuestro canal
de  YouTube

y descubre más contenido
educativo



Top Learning

@Toplearningoficial • 1.33 K suscriptores • 185 videos

¡Queremos a profesionistas a #desarrollarsunivelTop! +50k estudiantes certificados [ver más](#)

linktr.ee/toplearningoficial y 4 vínculos más

Suscribirse

Principal Videos Shorts Playlists 

Más recientes Popular Más antiguos



¿Cómo funciona una Tarjeta de Credito?
3 vistas • hace 17 horas



La Importancia de Tomar Cursos
11 vistas • hace 4 días



¿Quién es Marcos Galperin? -
Emprendedores Top
8 vistas • hace 7 días



Transformación Digital
23 vistas • hace 11 días



@toplearningoficial