



En Línea

#DesarrollaTuNivelTop



Macros

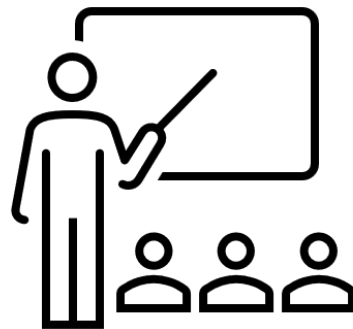


SEMANA 25



Objetivo particular de la lección

En esta sesión, el estudiante comprenderá las macros y su aplicación en Excel, aprendiendo a usar la Grabadora de Macros para automatizar tareas. Nos adentraremos en los fundamentos de VBA (Visual Basic for Applications), explorando variables y cómo manipular objetos de Excel desde el código. Aplicaremos buenas prácticas de programación para que, al finalizar, seas capaz de desarrollar una macro que copie datos automáticamente con un solo clic, optimizando así tu trabajo.



Grabadora de Macros

¿Qué es la grabadora de macros?

- Herramienta que registra tus acciones en Excel.
- Genera código VBA automáticamente.
- Ideal para principiantes que no conocen programación.

Cómo usarla:

1. Ir a la pestaña Vista o Desarrollador.
2. Clic en Grabar macro.
3. Realiza tus acciones (por ejemplo, dar formato, copiar datos).
4. Detén la grabación.
5. Ejecuta la macro para repetir esas acciones.

Consejo:

Puedes asignarla a un botón para facilitar su uso.

¿Qué es VBA?

VBA (Visual Basic for Applications)

Es un lenguaje de programación usado por Excel y otras apps de Office.

Permite escribir instrucciones más avanzadas que las grabadas por macros.

¿Por qué aprender VBA?

- Más control sobre tus macros.
- Crear lógica condicional, ciclos, validaciones
- Personalizar al 100% tus automatizaciones.



Variables en VBA

¿Qué es una variable?

Es un espacio en memoria donde se guarda información.

Puede ser texto, número, fecha, etc.

Ejemplos:

- Dim nombre As String
- Dim edad As Integer
- nombre = "Oscar"
- edad = 25

Importancia:

Las variables permiten hacer programas dinámicos y reutilizables.

Objetos en Excel

Todo en Excel es un objeto:

- Workbooks (libros)
- Worksheets (hojas)
- Range (rango de celdas)
- Cells (una celda)

Ventaja:

Puedes manipular todo lo que ves en Excel desde VBA.

Ejemplos de uso:

- Worksheets("Ventas").Activate
- Range("A1").Value = "Hola"

Celdas y Rangos

Cómo trabajar con celdas y rangos en VBA: Ejemplos de uso:

- `Range("A1").Value = "Dato"`
- `Cells(2, 3).Value = 100`
- `Range("B1:B10").Interior.Color = vbYellow`

Rangos dinámicos:

- Puedes usar variables para construir rangos:
 - `Range("A1:A" & ultimaFila).Copy`

Copiar datos con un solo clic

Ejemplo práctico: copiar datos de una hoja a otra

```
Sub CopiarDatos()
```

```
    Sheets("Origen").Range("A1:D100").Copy _
```

```
        Destination:=Sheets("Destino").Range("A1")
```

```
End Sub
```

¿Qué hace este código?

- Copia el rango A1:D100 de la hoja "Origen"
- Lo pega en A1 de la hoja "Destino"

Puedes vincular este código a un botón:

1. Inserta un botón
2. Asigna la macro CopiarDatos
3. ¡Listo! Copias todo con un clic.

Tipos de Variables y Buenas Prácticas

Tipo	Descripción	Ejemplo
String	Texto	"Hola mundo"
Integer	Números enteros	100, -5
Long	Números enteros grandes	1234567890
Double	Números decimales	3.14, -25.6
Boolean	Verdadero o falso	True, False
Date	Fechas y horas	#07/09/2025#
Variant	Tipo flexible (puede contener de todo)	Usar solo si es necesario

Ejemplo:

Dim nombre As String

Dim edad As Integer

Dim totalVentas As Double

Dim esClienteActivo As Boolean

Buenas Prácticas para Nombrar Variables y Objetos

Usa nombres claros y descriptivos:

- Malo: x, a1, temp1
- Bueno: totalVentas, nombreCliente, fechaInicio

Evita usar espacios, tildes o caracteres especiales.

- VBA no los acepta. Usa guiones bajos o cambia el formato.

Sé coherente en el estilo de nombrado.

Estilos comunes de nombrado

Estilo	Ejemplo	Uso típico
camelCase	nombreCliente, fechaInicio	Variables o funciones
PascalCase	NombreCliente, FechaInicio	Nombres de clases o módulos
snake_case	nombre_cliente, fecha_inicio	Poco común en VBA, más en Python

Recomendación en VBA:

- CamelCase para variables
- PascalCase para módulos, clases o funciones públicas

Ejemplo combinado:

```
Dim nombreCliente As String  
Dim totalComprasAnuales As Double
```

```
Function CalcularTotal()  
    ' Código aquí  
End Function
```



En Línea

#DesarrollaTuNivelTop



CONCLUIMOS LA PRIMER PARTE DE NUESTRA SESIÓN



SEMANA 25

CONTINUAMOS

Buenas Prácticas de Legibilidad en VBA

1. Usa indentación (tabulaciones)

- Cada vez que estés dentro de un bloque (If, For, Sub, etc.), usa Tab para que el código sea más claro.
- Mejora la lectura y el mantenimiento del código.

Ejemplo sin identar:

```
If total > 1000 Then  
MsgBox "Descuento aplicado"  
End If
```

Ejemplo con indentación:

```
If total > 1000 Then  
    MsgBox "Descuento aplicado"  
End If
```

Buenas Prácticas de Legibilidad en VBA

2. Divide líneas largas con _ (guión bajo)

- Si una línea es muy larga, usa _ para continuarla en la siguiente línea.
- Se usa solo si la línea no cabe cómodamente en pantalla.

Ejemplo:

```
Range("A1:D100").Copy _  
    Destination:=Range("E1")
```

Buenas Prácticas de Legibilidad en VBA

3. Usa comentarios (') para explicar el código

- Escribe con ' al inicio de la línea.
- Sirve para recordar qué hace cada parte y ayudar a otros (o a ti mismo en el futuro).

Ejemplo:

' Copia los datos de la hoja Origen a la hoja Destino

```
Sheets("Origen").Range("A1:D100").Copy  
Destination:=Sheets("Destino").Range("A1")
```

Buenas Prácticas de Legibilidad en VBA

4. Usa nombres descriptivos para variables y procedimientos

Evita:

```
Dim x As Integer  
Sub t()
```

Mejor:

```
Dim totalVentas As Integer  
Sub CalcularDescuento()
```

Buenas Prácticas de Legibilidad en VBA

5. Agrupa el código por bloques lógicos

Malo:

Todo el código junto y sin orden.

Bueno:

Agrupar por secciones: declaraciones, cálculos, salidas, etc.

Buenas Prácticas de Legibilidad en VBA

6. Deja líneas en blanco para separar ideas

Ejemplo:

Dim total As Double

Dim descuento As Double

total = 5000

If total > 1000 Then

 descuento = total * 0.1

End If

Buenas Prácticas de Legibilidad en VBA

7. Evita repetir código innecesariamente

Malo:

Copiar y pegar 5 veces la misma instrucción.

Mejor:

Usar bucles (For, Do While) o procedimientos (Sub, Function).

Activar la pestaña “Desarrollador” en Excel:

1. Ir a:

Archivo > Opciones > Personalizar cinta de opciones

1. En la columna derecha, marca la casilla "Desarrollador"

2. Clic en Aceptar

3. Ahora tendrás acceso a:

- Grabadora de macros
- Editor de VBA
- Controles de formulario y ActiveX

Habilitar Macros y Documentos Confiables

¿Cómo habilitar macros en Excel?

Cuando abres un archivo con macros, Excel puede mostrar una barra amarilla con un botón:

“Habilitar contenido”

Debes hacer clic en ese botón para que las macros se activen.

Otras formas de confiar en el archivo:

1. Ubicación de confianza

- Guarda el archivo en una carpeta marcada como confiable
- Ruta:

Archivo > Opciones > Centro de confianza > Configuración del Centro de confianza > Ubicaciones de confianza

2. Firmar el archivo digitalmente

- Si tienes un certificado, puedes firmar el archivo para que Excel lo reconozca como seguro

Otras formas de confiar en el archivo:

3. Desbloquear el archivo manualmente (si viene de internet)

- Haz clic derecho en el archivo > Propiedades > Marcar "Desbloquear" si aparece la opción
- Esto elimina la protección extra por haber sido descargado

4. Habilitar macros en la configuración

- Archivo > Opciones > Centro de confianza > Configuración de macros
- Puedes elegir:
 - Deshabilitar todas las macros
 - Habilitar solo las firmadas digitalmente
 - Habilitar todas (no recomendado sin precaución)

Seguridad: ¡Cuidado con macros externas!

¡Advertencia importante sobre seguridad!

Las macros son potentes... pero también pueden ser peligrosas.

¿Por qué Excel bloquea macros de fuentes externas?

- Las macros son código que puede realizar acciones automáticas (borrar, mover, modificar archivos)
- Hackers pueden usar macros para ejecutar virus, ransomware o espiar al usuario
- Por eso, Excel bloquea macros de archivos descargados o adjuntos de correo hasta que tú las habilites manualmente

Recomendaciones de seguridad:

- Solo habilita macros de archivos confiables
- No abras archivos con macros de fuentes desconocidas
- Si vas a compartir tu archivo, explica su contenido y por qué usa macros
- Considera usar una firma digital si compartes muchos archivos macro

Artículos o e-books con referencias en formato APA.

Alberto, S. (2020). *Beginning Power BI: A Practical Guide to Self-Service Business Intelligence with Power BI Desktop and Power BI Service*. Apress.

Gardner, M., & Excelerator BI. (2022). *Supercharge Power BI: Power BI & Power Query for Excel*.

Gerrard, D. (2021). *Microsoft Power BI Complete Reference: The Ultimate Guide to Design, Develop, and Deploy Powerful Business Intelligence Solutions*. Packt Publishing.

Microsoft. (n.d.). *Documentación de Power BI*. Microsoft Learn. Recuperado el 5 de julio de 2025, de <https://learn.microsoft.com/es-es/power-bi/>

Power BI Blog Oficial. (n.d.). *Microsoft Power BI Blog*. Recuperado el 5 de julio de 2025, de <https://powerbi.microsoft.com/es-es/blog/>

SEMANA 25 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



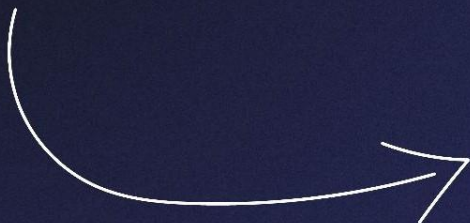
En Línea

#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 [...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