

# GUÍA BÁSICA DE MACROS EN EXCEL

## Objetivo

El participante será capaz de automatizar sus actividades mediante la creación de macros, mediante el análisis de los diferentes tipos de macros y la práctica de los métodos de programación que le permitan realizarlos en formas flexibles y acordes con sus necesidades de trabajo.

## Temario

### 1.- Automatización de Tareas

- a) Introducción
- b) Macros
- c) Cuando crear una macro en Excel
- d) Grabación de una macro
- e) Ejecución de una macro
- f) Uso de una macro

### 2.- Funciones y Procedimientos

- a) Funciones definidas por el usuario
- b) Procedimientos
- c) Objetos en Visual Basic. Código Visual Basic en procedimientos
- d) Uso de constantes para representar valores
- e) Creación de enunciados simples con argumentos con nombre
- f) Tipos de datos -Más información acerca de los procedimientos
- g) Uso de procedimientos propiedad

### 3.- Comandos para Pedir y Mostrar Información

- a) Objetivos
- b) Comandos para pedir información: Cuadros de diálogo, Método Cuadro Entr, Función Cuadro Msj

### 4.- Comandos para Toma de Decisiones

- a) Objetivos
- b) Comandos para toma de decisiones.
- c) Si...entonces, Si...entonces...Si Otro. Seleccionar Caso

### 5.- Comandos para Hacer Ciclos

- a) Comandos para hacer ciclos o bucles
- b) Hacer Bucle
- c) Para Siguiente
- d) Para Cada...Siguiente

# GUÍA BÁSICA DE MACROS EN EXCEL

## 6.- Editor de Menús

- a) Introducción
- b) Creación de elementos del menú
- c) Creación de una barra de menús nueva
- d) Creación de un menú nuevo
- e) Creación de un elemento de menú
- f) Creación de un submenú nuevo
- g) Eliminación, restauración y cambio del nombre

## 7.- Formularios Personalizados

- a) ¿Dónde están? ¿Cómo se crean? y ¿Cómo se ejecutan?
- b) Controles en un formulario
- c) Propiedades de los controles
- d) Macros automáticas en formularios

## 8.- Funciones Creadas por el usuario

- a) Funciones con cálculos propios
- b) Argumentos en funciones
- c) Llamar a una función personalizada
- d) Funciones de Excel vs Funciones de Visual Basic

# GUÍA BÁSICA DE MACROS EN EXCEL

## Macros en Excel

La Hoja de Cálculos Excel es una de las herramientas más poderosas para el tratamiento de datos y su análisis.

Una de las mayores ventajas que posee es la flexibilidad que ofrece y la rapidez con la que podemos construir plantillas y todo tipo de modelos, para diversos usos y aplicaciones.

**Macro:** Es un conjunto de instrucciones de código (programado) que permiten realizar una tarea determinada como así también expandir e incrementar las prestaciones de Excel. Los macros se escriben en lenguaje de programación VBA (Visual Basic for Applications). Excel ya trae incorporado un editor de VBA por lo que los macros se programan dentro del mismo programa Excel.

## Utilidad de una macro

Anteriormente definimos a los macros como algo que nos permite expandir e incrementar las prestaciones de la hoja Excel, pero a grandes rasgos podemos comentar que las 4 grandes "áreas" donde se aplican los macros son:

1. Automatización de tareas y procesos que involucran muchos pasos
2. Creación de nuevas funciones a medida (aparte de las que ya posee Excel).
3. Creación de nuevos comandos, complementos y menús.
4. Creación de completas aplicaciones a medida.

## Objetos, Propiedades y Métodos (OPM)

Cuando hablamos de macros hablamos de Objetos, Propiedades y Métodos (OPM). Estos son los 3 conceptos generales más importantes que debes conocer a la hora de programar los macros.

**Objetos:** prácticamente cada cosa que veas en Excel es un objeto, como, por ejemplo: un libro Excel, una hoja, un rango, una celda, un menú, un gráfico, una tabla dinámica, un cuadro de diálogo, las etiquetas de hojas, las columnas, las filas, etc. En fin, cada partecita de Excel es un objeto. Un objeto es algo que puedes ver e identificar con un nombre.

**Propiedades:** las propiedades son las características de los objetos. Por ejemplo, para el objeto "celda" algunas de sus propiedades serían: alto, ancho, color, bloqueada o desbloqueada, vacía, con un número o con una fórmula, etc. Por ejemplo, para el objeto "hoja" algunas de sus propiedades serían: visible u oculta, con o sin líneas de división, con o sin barras de desplazamiento vertical y horizontal, etc.

# GUÍA BÁSICA DE MACROS EN EXCEL

**Métodos:** un método es una acción que podemos realizar sobre el objeto o una de sus propiedades. Por ejemplo, sobre el objeto "hoja" podemos: activar, mover, copiar o borrar.

## Lenguaje VBA

Recuerda que la programación de macros se hace a través del lenguaje VBA (Visual Basic for Applications) desde el editor VBA que trae incorporado Excel. Este tipo de programación también se conoce como "Programación Orientada a Objetos" (OOP, Objects Oriented Programming). La Programación orientada a Objetos nos permite trabajar sobre los objetos propios de Excel como así también crear nuestros propios objetos.

Esto se relaciona con lo que te comentábamos en la sección anterior sobre Objetos, Propiedades y Métodos.

Ejemplos de Objetos VBA:

Cell (celda)  
Range (rango)  
Worksheet (hoja)  
Workbook (libro)

Ejemplos de Propiedades VBA:

Value (valor)  
Color (color)  
Format (formato)

Ejemplos de Métodos VBA:

Copy (copiar)  
Protect (proteger)  
Delete (borrar)

Sintaxis para escribir un macro

### **Sub Nombre\_Macro(Argumentos)**

'Comentarios  
Instrucción 1  
Instrucción 2  
Instrucción 3  
...  
...  
...  
Instrucción "n"

# GUÍA BÁSICA DE MACROS EN EXCEL

## End Sub

Ejemplo de un Macro:



```
(General) MiPrimeraMacro
Option Explicit
' Esta macro escribe el valor 1500 en la celda A1
Sub MiPrimeraMacro()
Range("A1").Value = 1500
End Sub
```

## Macros de Eventos

Como decíamos en la sección anterior, las macros también se pueden ejecutar automáticamente. En este caso se les denomina macros de evento, porque ante un determinado evento o acción realizada por el usuario, la macro se ejecuta automáticamente.

## Eventos de libros

Los eventos de libros permiten disparar automáticamente una macro cuando suceden determinadas acciones a nivel del libro Excel en cuestión, por ejemplo, al abrirlo, guardarlo, cerrarlo, imprimirlo, etc.

## Formularios (Userforms)

Un Formulario (o su denominación en inglés Userform) se utiliza para crear un Cuadro de Diálogo donde el usuario puede introducir información, o realizar otras operaciones.

Los Userforms se utilizan mucho para crear aplicaciones Excel que luzcan de forma profesional y permitan al usuario introducir datos o elegir opciones de una forma guiada y más intuitiva.

## Variables

Como prácticamente todas las macros utilizan variables, es importante comenzar a entenderlas. Una variable es sencillamente un dato o valor que VBA guarda en memoria.

Una variable es un lugar de almacenamiento con nombre que puede contener cierto tipo de datos que puede ser modificado durante la ejecución del programa. Cada variable tiene un nombre único que la identifica dentro de su nivel de ámbito. Puede especificar un tipo de datos



# GUÍA BÁSICA DE MACROS EN EXCEL

## Estructuras

Las estructuras son conjuntos de instrucciones VBA que deben respetar una sintaxis determinada y que nos permiten realizar diversas operaciones.

### Do Loop

La instrucción Do Loop es utilizada para repetir las mismas instrucciones una y otra vez hasta que una condición tenga lugar. Es probablemente la estructura más simple de cuantas hay en cuanto a repeticiones, pues no establece condiciones de entrada ni de salida como ocurre en Do While o Do Until.

### For Next

La instrucción For Next repite un grupo de instrucciones un número especificado de veces. La sintaxis es:

```
For contador = principio To fin [Step incremento] [instrucciones]
[instrucciones]
Next [contador]
```

La sintaxis de la instrucción For...Next consta de las siguientes partes:

- Contador (requerido): variable numérica que se utiliza como contador de bucle. La variable no puede ser Booleana ni un elemento de matriz.
- Principio (requerido): valor inicial del contador. -Fin (requerido): valor final del contador.
- Incremento (opcional): cantidad en la que cambia el contador cada vez que se ejecuta el bucle. Si no se especifica, el valor predeterminado de incremento es uno.
- Instrucciones (opcional): una o más instrucciones entre For y Next que se ejecutan un número especificado de veces.

### For Each Next

La instrucción For Each Next repite un grupo de instrucciones para cada elemento de una matriz o colección. Su sintaxis es:

```
For Each elemento In grupo [instrucciones]
[instrucciones]
Next [elemento]
```

Por lo general la instrucción For Each Next suele utilizarse para evaluar rangos y o hojas **While Wend**

# GUÍA BÁSICA DE MACROS EN EXCEL

La instrucción While Wend ejecuta una serie de instrucciones mientras una condición dada sea True, es decir verdadera, o tenga lugar. Su sintaxis es:

```
While condición  
    [instrucciones]  
Wend
```

Condición (requerido): expresión numérica o expresión de cadena cuyo valor es True o False. Si condición es Null, condición se considera False.

Instrucciones (opcional): una o más instrucciones que se ejecutan mientras la condición es True.

Si condición es True, todas las instrucciones se ejecutan hasta que se encuentra la instrucción Wend. Después, el control vuelve a la instrucción While y se comprueba de nuevo condición. Si condición es aún True, se repite el proceso. Si no es True, la ejecución se reanuda con la instrucción que sigue a la instrucción Wend.

## Go To

La instrucción GoTo es un método bastante rudo de variar el sentido que debería tomar una macro, ya que cuando una macro llega a una línea que contiene dicha instrucción "salta" al lugar especificado en GoTo (es decir, saltamos incondicionalmente a una línea especificada en el procedimiento). La Sintaxis de GoTo es:

### GoTo línea

El argumento línea puede ser cualquier etiqueta de línea o número de línea.

## If Then

La instrucción If Then permite escoger entre 2 o más posibilidades, para decidir sobre qué es lo que queremos, entre una serie de alternativas. La instrucción ejecuta condicionalmente un grupo de instrucciones, dependiendo del valor de una expresión. Su sintaxis es:

```
If condición Then  
    [instrucciones]  
End If
```

```
If condición Then  
    [instrucciones]
```

# GUÍA BÁSICA DE MACROS EN EXCEL

```
[Else  
    [instrucciones] End
```

```
If
```

También podemos utilizar la siguiente sintaxis en formato de bloque:

```
If condición Then  
    [instrucciones]  
[Elseif condición-n Then [instrucciones_elseif]
```

```
...
```

```
[Else  
    [instrucciones_elseif]]  
End If
```

## Select Case

La instrucción Select Case se utiliza cuando las alternativas en una estructura de If Then son muchas. En tal caso conviene evitar el If Then y hacer uso de Select Case, que permite ejecutar uno de varios grupos de instrucciones, dependiendo del valor de una expresión.

La sintaxis de Select Case es:

```
Select Case expresión_prueba [Case  
    lista_expresion-n  
    [instrucciones-n]] ...  
    [Case Else  
    [instrucciones_elseif]]  
End Select
```

## Ejemplos de Macros

Para cerrar un libro utilizamos la instrucción Close:

# GUÍA BÁSICA DE MACROS EN EXCEL

```
Sub Cerrar()  
ActiveWorkbook.Close  
End Sub
```

Si queremos cerrarlo sin que nos pregunte si queremos guardar los posibles cambios realizados, debemos indicar que no queremos que se muestren los mensajes de alerta:

```
Sub Cerrar()  
Application.DisplayAlerts = False  
ActiveWorkbook.Close  
End Sub
```

Para cerrar todos los libros abiertos sin que nos pregunte si queremos guardar los posibles cambios registrados en cada libro si los hubiese:

```
Sub Cerrar()  
Application.DisplayAlerts = False  
Application.Workbooks.Close  
End Sub
```

Y dejando que nos pregunte:

```
Sub Cerrar()  
Application.Workbooks.Close  
End Sub
```

```
Sub Macro()  
On Error Resume Next  
' Ponemos On Error Resume Next, para evitar el error derivado de una celda vacía y/o con un  
carácter con el que  
' excel no permite guardar un libro  
ActiveWorkbook.SaveAs [A1].Value  
' También podríamos poner  
' ActiveWorkbook.SaveAs Filename:=Range("A1").Value  
End Sub
```

Para borrar un libro utilizamos la instrucción Delete:

```
Sub BorraLibro ()  
Workbooks("C:\TuLibro.xls").Delete
```

# GUÍA BÁSICA DE MACROS EN EXCEL

End Sub

Para eliminar el libro activo, en caso de que estuviera abierto, primero deberíamos cerrarlo, tal que:

```
Sub Eliminar() Dim  
strFile As String  
strFile = ActiveWorkbook.FullName ActiveWorkbook.Close  
SaveChanges:=False  
Kill strFile  
End Sub
```

Para crear 1 nuevo libro utilizamos la instrucción Add, por ejemplo:

```
Sub NuevoLibro()  
Workbooks.Add  
End Sub
```

Si quisiéramos crear 2 nuevos libros:

```
Sub NuevoLibro()  
Workbooks.Add  
Workbooks.Add  
End Sub
```

Abrir un libro en blanco o crear un libro es lo mismo, pero también podemos crear libros basados en plantillas o en otras hojas de cálculo, es decir en otros libros tal que:

```
Sub Macro()  
Dim Libro1 As Workbook  
Dim Libro2 As Workbook  
Set Libro1 = Workbooks.Add  
Debug.Print Libro1.Name 'Utilizamos la ventana Inmediata del editor de VBA  
Set Libro2 = Workbooks.Add  
Libro1.Activate  
End Sub
```

## Rangos

Mediante macros VBA podemos manipular de distintas formas los rangos y celdas de las hojas Excel, veremos cómo hacerlo.

Selección de celdas

# GUÍA BÁSICA DE MACROS EN EXCEL

Existen diversas formas de seleccionar celdas, y en todas ellas utilizamos el método Select.

Veamos 3 métodos (equivalentes) para seleccionar una celda, por ejemplo la celda C4:

' La forma clásica de seleccionar

```
Sub Seleccionar ()  
Range("C4").Select  
End Sub
```

' Esta es otra terminología válida para seleccionar

```
Sub Seleccionar ()  
[C4].Select  
End Sub
```

' Finalmente podemos seleccionar de esta otra forma '

El primer N° equivale a fila y el segundo a columna.

```
Sub Seleccionar ()  
Cells(4, 3).Select  
End Sub
```