

Unidad Didáctica 7

**Operaciones más  
cotidianas con Visual Basic**

---

# Contenido

1. Introducción
2. Instrucciones básicas para trabajar con libros
3. Instrucciones básicas para trabajar con hojas
4. Instrucciones básicas para trabajar con celdas y rangos

## 1. Introducción

A veces, en la necesidad de automatizar las tareas de trabajo, el manejo de *Excel* cotidiano puede resultar incompleto. Es por ello que surge la necesidad de elaborar macros propias que actúen de una manera determinada.

En este capítulo, se verán las instrucciones más frecuentes que se suelen utilizar para el manejo de libros y hojas de cálculo.

Los objetos a los que se hará referencia son: Libro de trabajo (**Workbooks**), Hoja de cálculo (**Worksheets**) y Rangos y celdas (**Range** y **cells**).

## 2. Instrucciones básicas para trabajar con libros

En este apartado, se van a ver las instrucciones básicas que se deben conocer para manejar un libro de trabajo.

El objeto principal para trabajar con un libro es **Workbooks**. Siempre que se quiera hacer referencia a un libro o realizar alguna operación con el libro, aparecerá este objeto. Cuando se quiera hacer referencia al libro que en ese momento se tiene activo, se escribirá la instrucción **ActiveWorkbook** seguida de la propiedad.

Lo mejor para entenderlo será ir viéndolo con ejemplos y poder ir practicándolos al mismo tiempo.

Para empezar, las instrucciones que se irán mostrando pueden ingresarse directamente en la ventana **Inmediato** o bien crear una macro para cada una de ellas. Para esto, se crea un nuevo módulo desde el menú **Insertar** y se escribe en él. Basta con escribir por ejemplo **Sub ejemplo()** y el código dentro de este procedimiento.

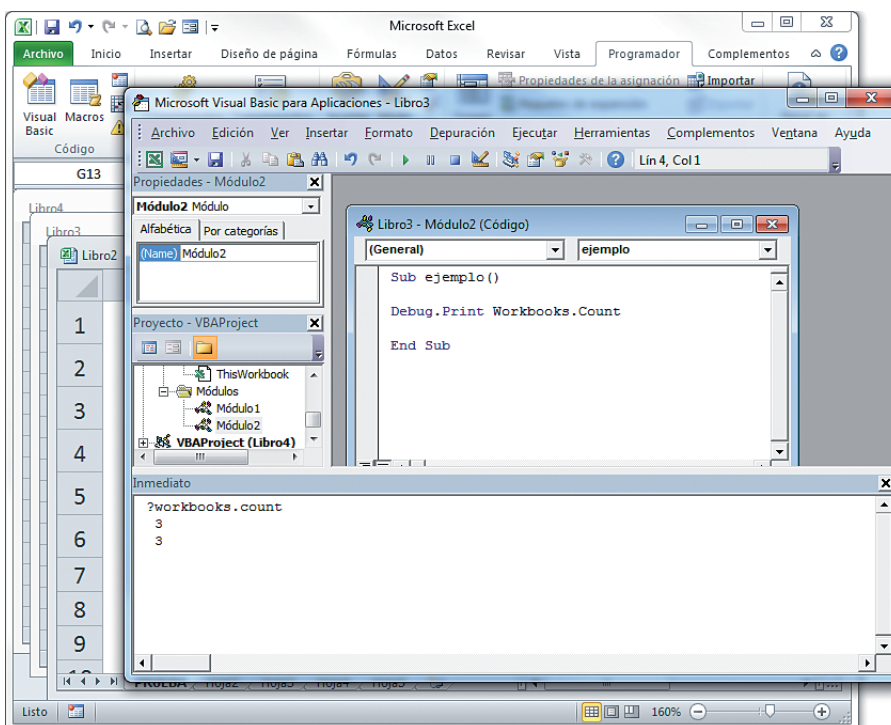


## Consejo

Se recomienda escribir el código en la ventana Inmediato para poder ver directamente lo que hace.

Se van a comenzar viendo las instrucciones referentes al libro de trabajo:

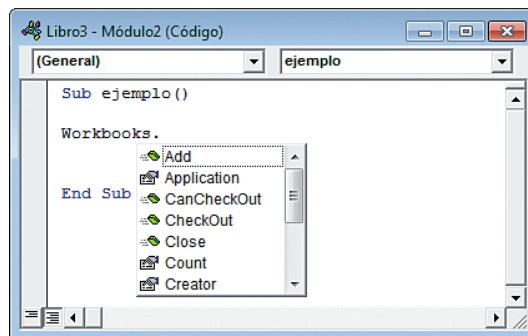
- **Añadir un libro de trabajo nuevo.** Esta instrucción creará un nuevo libro de trabajo en blanco: `Workbooks .Add`
- **Contar libros abiertos:** `Debug.Print Workbooks.Count`



Contar libros abiertos

- **Cerrar un libro de trabajo.** Si se tiene más de un libro abierto, se cambia la expresión **Nombre\_libro** por el nombre de uno de los libros abiertos: `Workbooks("Libro4").Close`
- **Abrir un libro de trabajo existente:** `Workbooks.Open "C:\Mis docu\Libro1.xlsx"`
- Cerrar el libro activo guardando los cambios:  
`ActiveWorkbook.Save`  
`ActiveWorkbook.Close`
- **Guardar un libro de trabajo con otro nombre:** `ActiveWorkbook.SaveAs Filename:="C:\Mis doc\Libro.xls",,,`  
**Nota:** Obsérvese que, al final, se ponen tres comas, ya que la instrucción tenía más parámetros que se han obviado.
- **Poner en primer plano un libro.** Para ver el funcionamiento de esta instrucción, se deberá tener abierto más de un libro y llamar a uno por su nombre: `Workbooks("Libro2.xls").Activate`

El objeto **Workbooks** tiene muchas propiedades, que se pueden ver en el editor de código. Cuando se introduce este objeto en un procedimiento y se pulsa la tecla punto, se despliega una lista de propiedades que se le pueden aplicar al objeto.



*Lista de propiedades aplicables al objeto*



## Importante

---

Cuando se introduzcan las instrucciones en la ventana de Datos, no hay que olvidar escribir primero el signo de interrogación ? o la palabra Print.

---

### 3. Instrucciones básicas para trabajar con hojas

El objeto principal para trabajar con las hojas de cálculo es **WorkSheets**. En este caso, para referirse a una determinada hoja, existen varios métodos:

- **ActiveSheet**: hace referencia a la hoja activa en ese momento.
- **Sheets("lunes")**: hace referencia a la hoja de cálculo llamada lunes.
- **Sheets(3)**: hace referencia a la tercera hoja, según el orden de las mismas.

Una vez vistas las formas de referirse a una hoja, se van a ver unos ejemplos de su funcionamiento:

#### Añadir una nueva hoja de cálculo

```
Worksheets .Add
```

#### Renombrar la primera hoja

```
Worksheets(1) .Name = "Ejemplo"
```

#### Activar una hoja por su nombre

```
Worksheets("Hoja2") .Activate
```

#### Borrar la hoja activa

Al ejecutar esta instrucción nos aparecerá una alerta advirtiéndonos de que puede contener datos.

```
ActiveSheet.Delete
```

### Copiar una hoja de cálculo

Cuando copiamos una hoja de cálculo sin especificar un destino dónde pegar, se crea un nuevo libro con la copia.

```
ActiveSheet.Copy
```

### Seleccionar la hoja anterior o siguiente

```
ActiveSheet.Previous.Select 'hoja anterior a la activa  
ActiveSheet.Next.Select 'hoja posterior a la activa
```

### Proteger o desproteger una hoja:

```
ActiveSheet.Protect "contraseña" 'proteger con contraseña  
ActiveSheet.Unprotect "contraseña" 'quitar la protección
```

### Eliminar la fila 10 de una hoja

```
ActiveSheet.Row(10).Select  
Selection.EntireRow.Delete
```

### Ver una vista preliminar de la hoja activa

```
ActiveWindow.ActiveSheet.PrintPreview
```



### Nota

---

Cuando se quiera hacer referencia a una fila o columna, se debe hacer referencia a la propiedad Row o Colum respectivamente.

---

## 4. Instrucciones básicas para trabajar con celdas y rangos

Llegado este último paso, tan solo queda ver cómo trabajar con celdas y rangos de celdas. Los objetos principales en este caso son dos: **Range** y **Cells**. Se puede hacer referencia a estos objetos de las siguientes maneras:

- **Activecell**: se hace referencia a la celda activa.
- **Range("A7")**: se hace referencia a la celda A7.
- **Cells(7,1)**: se hace referencia a la celda A7.

Cuando se utiliza **Range**, primero se pone la columna y después la fila. En **Cells** es al contrario, primero se pone la fila y después la columna. Algunos ejemplos de lo que se puede hacer con las celdas y rangos son:

### Seleccionar una celda concreta

```
Range("B2").Select
```

### Seleccionar un rango concreto

```
Range("B2:B7").Select
```

### Seleccionar todas las celdas de la hoja de cálculo

```
Cells.Select
```

### Seleccionar La columna B entera

```
Range("B:B").Select
```

### Seleccionar una celda cuya fila es una variable

```
Range("B" & variable).select
```

**Ver la dirección de la celda activa (absoluta)**

Esta instrucción, nos mostrará la dirección de la celda de una manera absoluta.

```
Debug.Print ActiveCell.Address
```

**Ver la dirección de la celda activa (relativa)**

Esta instrucción, nos mostrará la dirección de la celda de una manera relativa.

```
Debug.Print ActiveCell.Address(False, False)
```

**Combinar celdas seleccionadas**

En este caso se combinará el rango B1:E1.

```
Range("B1:E1").Select
Selection.Merge
```

**Obtener la primera celda vacía de una Columna**

```
Range("A1").Select
Do While Not IsEmpty(ActiveCell)
    ActiveCell.Offset(1, 0).Select
Loop
```

**Contar las celdas seleccionadas**

Para utilizar esta instrucción, primero debe seleccionar un rango de celdas. El número que nos muestre esta instrucción, se corresponde con el número de celdas que estén seleccionadas.

```
Debug.Print Selection.Count
```

### Borrar el contenido de una celda sin borrar el formato

```
Range (rango) .Select  
Selection.ClearContents
```

### Borrar el contenido y el formato de una celda

```
Range (rango) .Select  
Selection.Clear
```



#### Nota

---

Se han podido ver los métodos más importantes del manejo con libros, hojas, rangos y celdas. La aplicación de estas instrucciones u otras que se elaboren, se deberá en gran medida a la creatividad que uno tenga, ya que no hay una sola forma de hacer las cosas en Visual Basic.

---