

Editions ENI

# **VBA Excel 2007**

**Programar en Excel: Macros y Lenguaje VBA**

colección Recursos Informáticos

*Extracto del Libro*

## A. El objeto **Application**

El objeto **Application** representa la aplicación Microsoft Excel activa. Es el objeto por defecto y, por lo tanto, es generalmente opcional (ejemplo: *Version* equivale a *Application.Version*).

Este objeto contiene:

- las **propiedades relativas al entorno Excel** (opciones de Excel, etc.) y a la **presentación de la interfaz** (puntero del ratón, texto de la barra de estado, tamaño y estado de la ventana de la aplicación, etc.),
- hay distintos **métodos** para realizar acciones en el entorno Excel,
- **propiedades** que devuelven objetos y colecciones de primer nivel (objetos y colecciones del modelo de objetos Excel: *Workbooks*, *Charts*, etc.),
- **propiedades específicas** que hacen referencia directa a objetos: *ActiveCell*, *ActiveSheet*, *ActiveWindow*, etc.

---


➤ Las propiedades que hacen referencia a objetos se explican en el capítulo anterior.

---

### 1. Propiedades que representan las opciones de Excel

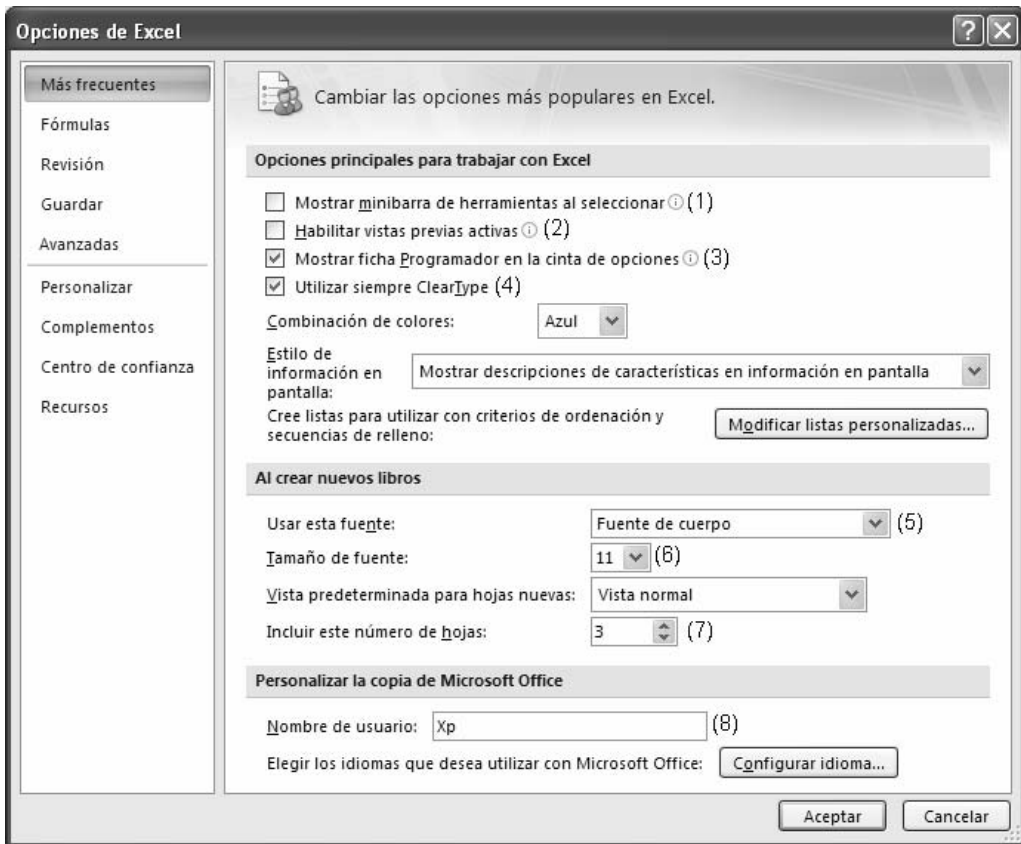
Las principales opciones de Excel pueden ser devueltas o definidas a partir de propiedades del objeto **Application**. La mayoría de estas propiedades son de lectura y escritura.

---

➤ Para acceder a las opciones Excel en la versión 2007, haga clic en el botón **Microsoft Office**  y luego en el botón **Opciones de Excel** (situado en la parte inferior del cuadro de diálogo).

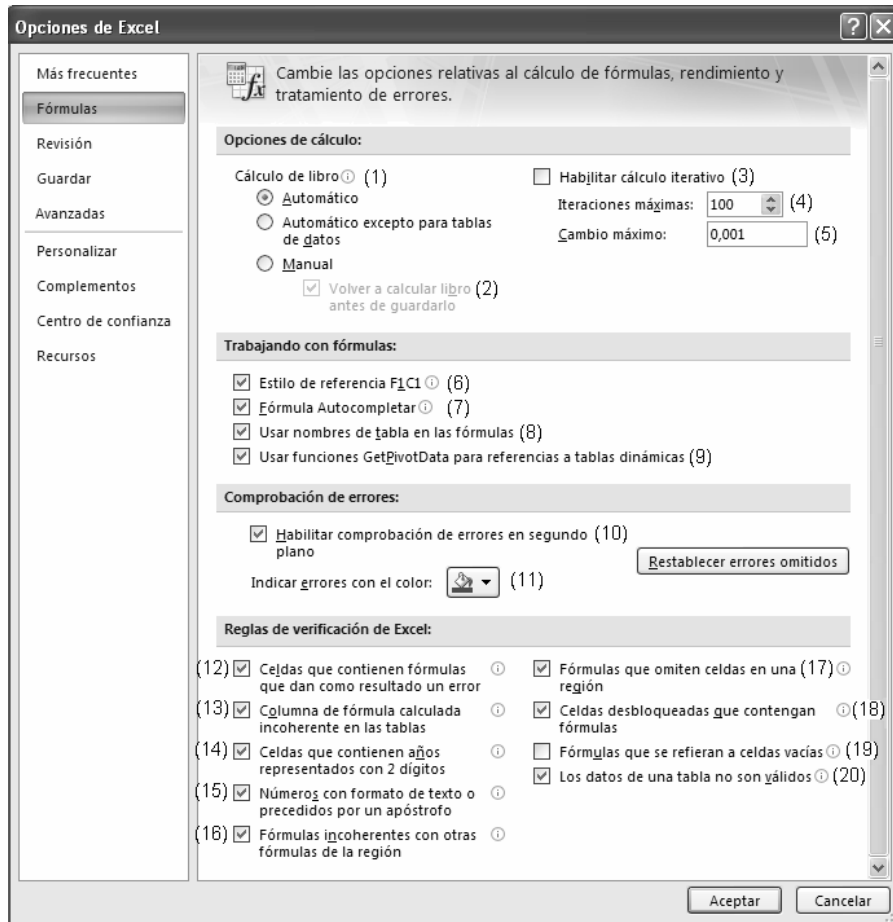
---

a. Opciones de la categoría Más frecuentes



N°	Propiedades	Valores devueltos
1	ShowSelectionFloaties	Boolean
2	EnableLivePreview	Boolean
3	ShowDevTools	Boolean
4	AlwaysUseClearType	Boolean
5	StandardFont.	Entero largo
6	StandardFontSize	Entero largo
7	SheetsInNewWorkbook	Entero largo
8	UserName	Cadena de caracteres

## b. Opciones de la categoría Fórmulas



N°	Propiedades	Valores devueltos
1	Calculation Constantes:	Constante xlCalculationAutomatic xlCalculationManual xlCalculationSemiautomatic
2	CalculateBeforeSave	Boolean
3	Iteration	Boolean
4	MaxIterations	Entero largo
5	MaxChange	Doble

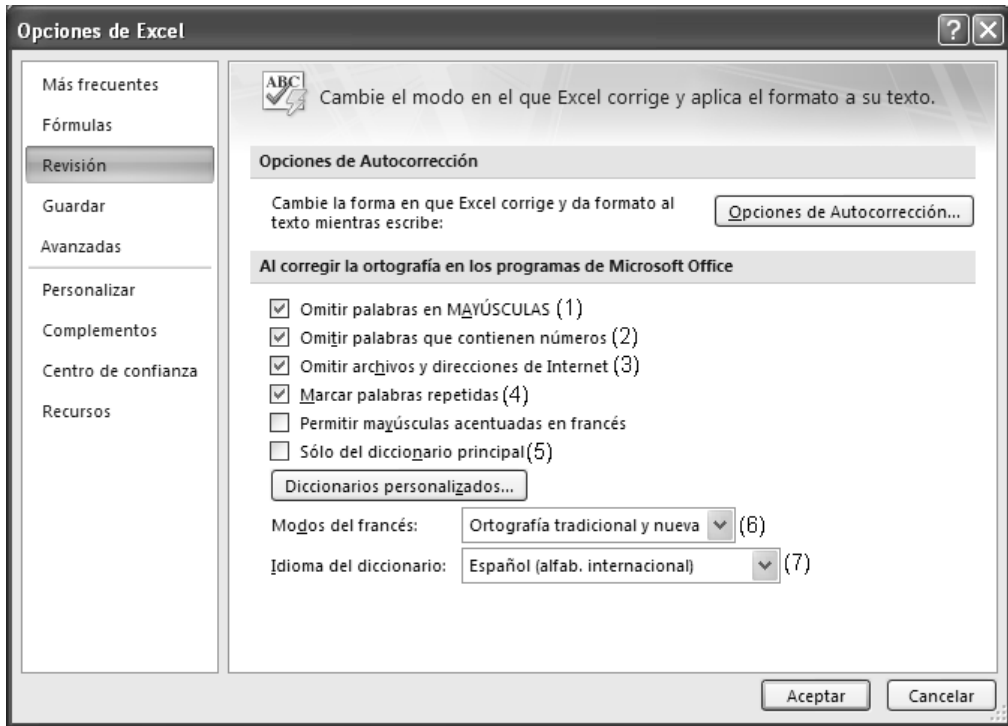
<b>N°</b>	<b>Propiedades</b>	<b>Valores devueltos</b>
6	ReferenceStyle	Constantes: <b>xIA1</b> <b>xIR1C1</b>
7	DisplayFormulaAutoComplete	Boolean
8	GenerateTableRefs	Constantes: <b>xIGenerateTableRefStruct</b> <b>xIGenerateTableRefA1</b>
9	GenerateGetPivotData	Boolean

### Opciones de comprobación de errores

Las siguientes propiedades dependen de la propiedad **ErrorCheckingOptions** del objeto Application. Esta propiedad devuelve un objeto **ErrorCheckingOptions**, que representa las opciones de comprobación de errores para una aplicación.

<b>N°</b>	<b>Propiedades</b>	<b>Valores devueltos</b>
10	BackgroundChecking	Boolean
11	IndicatorColorIndex	Constante XIColorIndex
12	EvaluateToError	Boolean
13	InconsistentTableFormula	Boolean
14	TextDate	Boolean
15	NumberAsText	Boolean
16	InconsistentFormula	Boolean
17	OmittedCells	Boolean
18	UnlockedFormulaCells	Boolean
19	EmptyCellReferences	Boolean
20	ListDataValidation	Boolean

### c. Opciones de la categoría Revisión

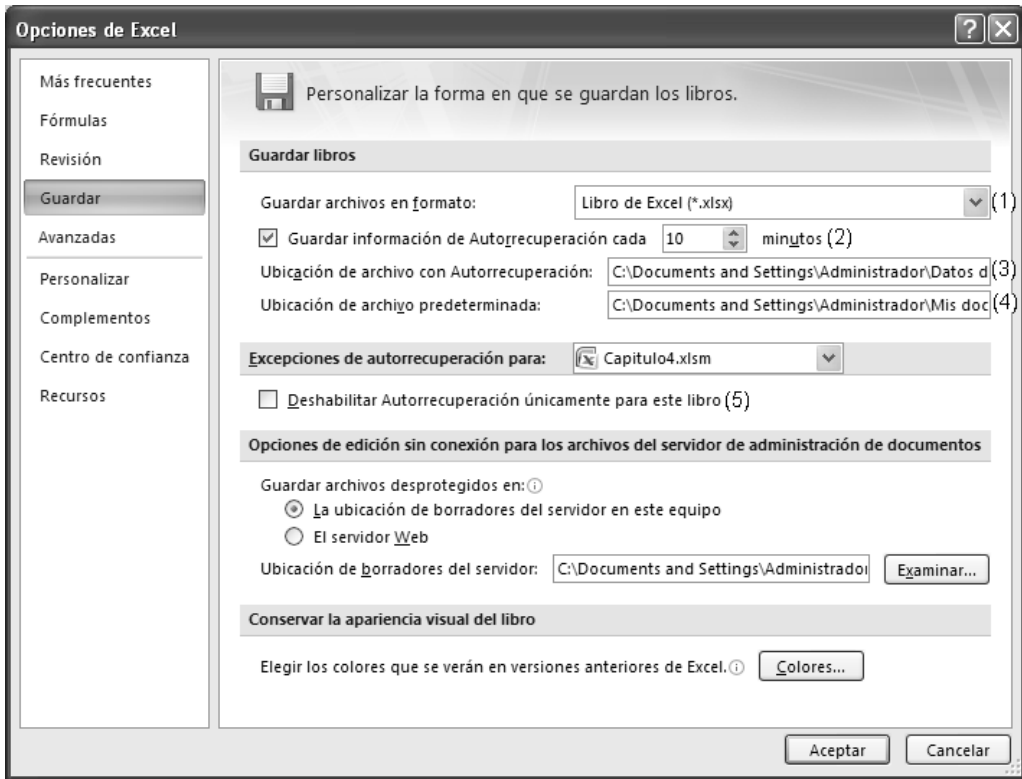


### Opciones de revisión ortográfica

Las siguientes propiedades dependen de la propiedad **SpellingOptions** del objeto **Application**. Esta propiedad devuelve un objeto **SpellingOptions**, que representa las opciones de revisión ortográfica para una aplicación.

Nº	Propiedades	Valores devueltos
1	IgnoreCaps	Boolean
2	IgnoreMixedDigits	Boolean
3	IgnoreFileNames	Boolean
4	GermanPostReform	Boolean
5	SuggestMainOnly	Boolean
6	ArabicModes	Constante XIArabicModes
7	DictLang	Entero largo

d. Opciones de la categoría Guardar



Nº	Propiedades	Valores devueltos
1	DefaultSaveFormat	Constante xlFileFormat
2	AutoRecover.Time	Entero largo
3	AutoRecover.Path	Cadena de caracteres
4	DefaultFilePath	Cadena de caracteres
5	EnableAutoRecover	Boolean

➤ La opción 5 se aplica al libro activo (Application.ActiveWorkbook).

Editions ENI

# **VBA Excel 2007**

**Aprenda a crear aplicaciones profesionales**

colección Prácticas Técnicas

*Extracto del Libro*

## Capítulo 4: Tablas

- **Duración:** 1 hora 30
- **Palabras clave:** declaración, alcance, índice, estático, dinámico, límite, inicialización, redimensionar, recorrido
- **Objetivos**

Uso de tablas para generar conjuntos de muchos elementos, generalmente del mismo tipo.

### Prerrequisitos

*Para comprobar los prerrequisitos necesarios, antes de encarar el TP, responda a las siguientes preguntas (ciertas preguntas admiten más de una respuesta):*

1. Todos los elementos de una tabla:
  - a. tienen el mismo tipo de datos que el tipo declarado para la tabla.
  - b. pueden ser de diferente tipo si la tabla se declara de tipo Variant.
  - c. pueden ser de tipo String o Integer si la tabla se declara de tipo String.
2. Las tablas estáticas:
  - a. tienen límite superior e inferior.
  - b. no tienen ningún límite.
  - c. tienen solamente límite inferior.
3. Los elementos de una tabla se indexan:
  - a. de manera relativa.
  - b. secuencialmente.
4. El número de índice por defecto del primer elemento de una tabla VBA es:
  - a. - 1
  - b. 1
  - c. 0
5. Un número de índice se refiere a:
  - a. un único elemento de una tabla.
  - b. muchos elementos de una tabla.
6. Una tabla de tipo Variant puede contener otras tablas si es:
  - a. estática
  - b. dinámica

7. Una tabla declarada con la palabra clave Dim en un procedimiento es accesible:
  - a. por todos los procedimientos del módulo.
  - b. por todos los procedimientos del proyecto.
  - c. sólo en el interior de su procedimiento.
8. La dimensión máxima de una tabla VBA es:
  - a. 10
  - b. 100
  - c. 60
9. Se puede precisar los índices inicial y final con la cláusula:
  - a. To
  - b. For
  - c. Next
10. El siguiente procedimiento es incorrecto porque:
 

```
Sub ProcTest()
    Private TabTest(19) As String
    TabTest(19) = "paraíso"
    Debug.Print "Vigésimo elemento :"; TabTest(19)
End Sub
```

  - a. No se indican los límites.
  - b. Se usa Private dentro del procedimiento.
  - c. El signo de asignación para tablas es :=.

### Atención

Todos los ejercicios se deben hacer con la opción de base por defecto: Option base 0.

### **Enunciado 4.1: Declarar una tabla accesible por todos los procedimientos del proyecto**

**Duración estimada:** 5 minutos

Declare una tabla pública de 100 elementos enteros. Déle el nombre **TabPub**.

## Enunciado 4.2: Declarar una tabla accesible solamente por los procedimientos del módulo

**Duración estimada:** 5 minutos

Declare una tabla de fechas con 100 elementos a nivel de módulo. Déle el nombre **TabFecha**.

## Enunciado 4.3: Usar una tabla declarada en un procedimiento

**Duración estimada:** 10 minutos

1. Declare una tabla de palabras con 100 elementos. Esta tabla es local a un procedimiento llamado **EX3**. Déle el nombre **TabPal**.
2. Complete el procedimiento **EX3** para que muestre los límites inferior e superior de la tabla TabPal (enunciado 4.3). Asígnele la palabra "océano" al primer elemento. Muestre también este elemento con **Debug.Print**.

```
Inmediato
Límite inferior : 0
Límite superior : 99
Primer elemento : océano
```

### Sugerencia

Las funciones *LBound* y *UBound* devuelven respectivamente los límites inferior y superior de la tabla.

### Atención

Si es necesario, haga clic en **Ver - Ventana inmediato** para ver el resultado.

## Enunciado 4.4: Usa una tabla a nivel de módulo

**Duración estimada:** 5 minutos

Use la tabla **TabPub** (enunciado 4.1) en un procedimiento llamado **EX4**. Asígnele el valor 123 al último elemento. Muestre este elemento con `Debug.Print`.

```
Inmediato
Último elemento: 123
```

## Enunciado 4.5: Inicializar una tabla con un bucle

**Duración estimada:** 5 minutos

Procedimiento **EX5**. Llene una tabla local llamada **TabInit** con 10 valores enteros aleatorios de 1 a 100. Muestre todos los valores junto con el total de la tabla.

Inmediato	
Número 1:	43
Número 2:	32
Número 3:	55
Número 4:	56
Número 5:	6
Número 6:	96
Número 7:	27
Número 8:	68
Número 9:	14
Número 10:	68
Total	465

## Sugerencia

Use la instrucción *Randomize* que inicializa el generador de números aleatorios y la función *Rnd*. Ejemplo:

```
Dim MiValor
Randomize ' Inicializa el generador de números aleatorios.
' Devuelve un valor aleatorio comprendido entre 1 y 6.
MiValor = Int((6 * Rnd) + 1)
```

Use el bucle *For...Next* para llenar la tabla.

## Enunciado 4.6: Declarar y usar una tabla de dos dimensiones

**Duración estimada:** 5 minutos

Declare y use una tabla multidimensional de tipo entero con 5 filas y 5 columnas (option base por defecto). Esta tabla es local a un procedimiento llamado **EX6**. Déle el nombre **Tab2D**. Inicialice la tabla con números aleatorios de 1 a 100. Muestre todos los elementos recorriendo la tabla por filas y columnas.

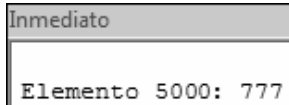
Ejemplo:

Inmediato					
Fila	1	Columna	1	-->>	4
Fila	1	Columna	2	-->>	26
Fila	1	Columna	3	-->>	3
Fila	1	Columna	4	-->>	36
Fila	1	Columna	5	-->>	61
Fila	2	Columna	1	-->>	58
Fila	2	Columna	2	-->>	20
Fila	2	Columna	3	-->>	63
Fila	2	Columna	4	-->>	68
Fila	2	Columna	5	-->>	99
Fila	3	Columna	1	-->>	34
Fila	3	Columna	2	-->>	38
Fila	3	Columna	3	-->>	17
Fila	3	Columna	4	-->>	57
Fila	3	Columna	5	-->>	72
Fila	4	Columna	1	-->>	11

### Enunciado 4.7: Declarar y usar una tabla de más de dos dimensiones

**Duración estimada:** 10 minutos

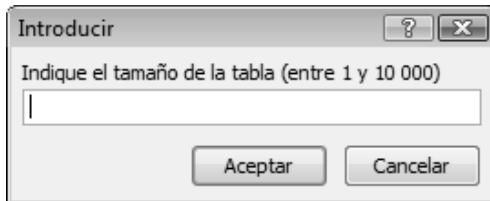
1. Declare una tabla tridimensional de tipo entero a nivel de módulo de 10 por 10 por 50. Déle el nombre **Tab3D**.
2. Cree un procedimiento llamado **EX7**. Asigne el valor 777 al 5000º elemento de la tabla dimensional **Tab3D** (enunciado 4.7). Muestre este elemento.



### Enunciado 4.8: Declarar y usar una tabla dinámica

**Duración estimada:** 10 minutos

Procedimiento **EX8**: cree una tabla dinámica de tipo módulo de una columna que contenga números. Déle a la tabla el nombre **TabDin**. Muestre un cuadro de entrada para asignar al último elemento según tamaño de la tabla. Muestre este elemento.



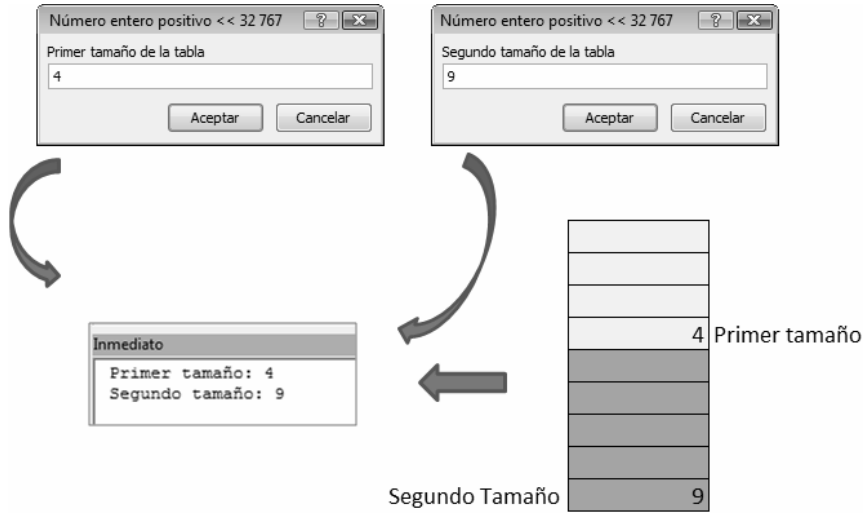
#### **Atención**

Evite indicar un tamaño demasiado grande para no saturar la memoria.

### Enunciado 4.9: Agrandar una tabla dinámica, manteniendo los valores iniciales

**Duración estimada:** 15 minutos

Procedimiento **EX9**: use la tabla dinámica **TabDin** (enunciado 4.8). Pregunte un primer tamaño y luego otro para agrandar la tabla. Muestre los dos tamaños, asignados a los elementos correspondientes de la tabla.



Ejemplo con  $\text{TabDin}(\text{Tamaño1}) = 4$  y  $\text{TabDin}(\text{Tamaño2}) = 9$

### Sugerencia

Use la palabra clave **Preserve** para conservar los primeros elementos de la tabla.

## Enunciado 4.10: Mostrar los elementos de una tabla con ayuda de un bucle

**Duración estimada:** 10 minutos

Escriba el procedimiento **EX10** que muestra una tabla de multiplicación según la elección del operador. Ejemplo:

Introducir

Introduzca el número para la tabla de multiplicar.

7

Aceptar      Cancelar

7					7
7	x	1	=		14
7	x	2	=		21
7	x	3	=		28
7	x	4	=		35
7	x	5	=		42
7	x	6	=		49
7	x	7	=		56
7	x	8	=		63
7	x	9	=		70
7	x	10	=		