



Microsoft Office Excel 2010

- **Macros**
- **Visual Basic per Aplicacions**



Aquesta obra estan subjectes –llevat que s'indiqui el contrari– a una llicència de Reconeixement--NoComercial - Compartir igual (BY-NC-SA) v.3.0 Espanya de Creative Commons. Podeu modificar l'obra, reproduir-la, distribuir-la o comunicar-la públicament, sempre que no sigui per a fins comercials sense permís exprés de l'autor i sempre que en citeu l'autor i la font (Julià Moreno Alonso), i sempre que l'obra derivada quedi subjecta a la mateixa llicència que el material original. La llicència completa es pot consultar a:

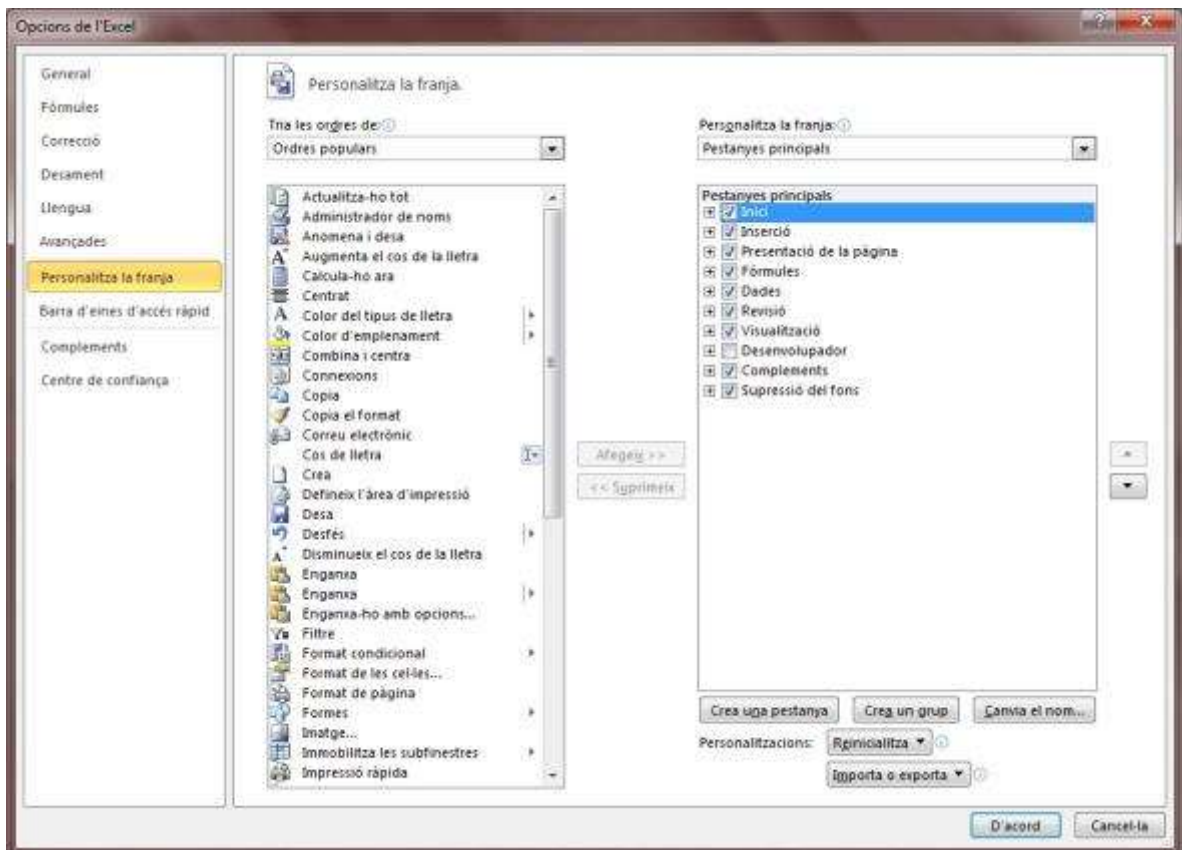
<http://creativecommons.org/licenses/by-nc-sa/3.0/es/>

Í N D E X

1. Macros.....	4
Gravar Macros.....	5
Gestió de les Macros.....	6
Assignació d'una Macro a un element.....	8
2. VBA – Visual Basic per Aplicacions	9
Mòduls en VBA.....	10
Els comentaris.....	11
Tipus de procediments	12
Procediments (Sub).....	12
Funcions (Function).....	12
Objectes i col·leccions	13
Propietats i mètodes.....	13
Les variables i les constants	14
Declaració de variables.....	15
Quadres de missatges	16
Sentències Condicionals.....	17
El condicional If.....	17
Sentències iteratives.....	18
For ... Next	18
Do While ... Loop.....	19
Webgrafia	20

1. MACROS

Les macros ens ajudaran a automatitzar accions que realitzem freqüentment amb Microsoft Excel. Tot allò referent a les Macros ho podrem trobar a dins de la pestanya *Desenvolupador*. Aquest pestanya per defecte està oculta i si necessitem mostrar-la hauréem d'anar al menú *Fitxer*, escolliríem *Opcions* i a dins de la categoria *Personalitza la franja* trobarem l'opció *Desenvolupador*, només caldrà marcar-la i ja tindrem l'opció a la nostra cinta d'opcions.



Pestanya desenvolupador, una vegada l'hem activat

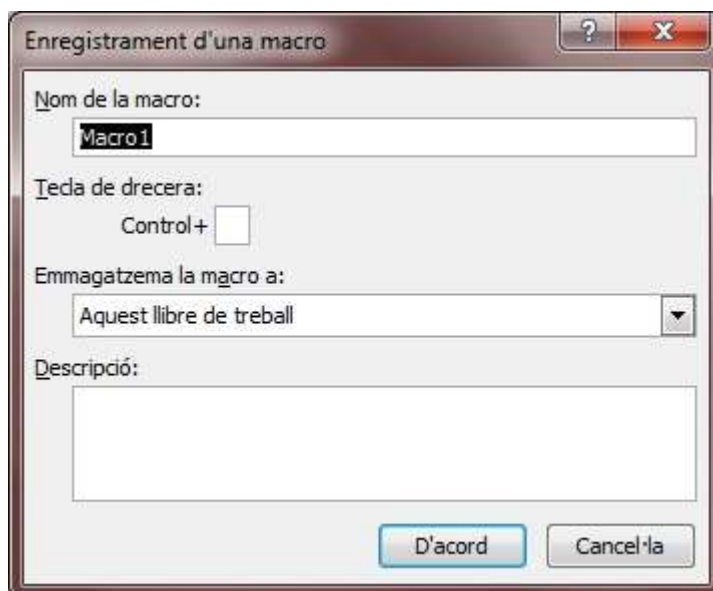


GRAVAR MACROS

Al gravar una macro, Excel emmagatzema informació sobre cada pas donat quan s'executa una sèrie de comandaments. A continuació, s'executa la macro per que repeteixi els comandaments. Si al gravar la macro realitzem algun pas que no volíem i utilitzem l'eina desfer, també guardarà les correccions que realitzem. Per aquest motiu haurem d'anar amb compte de tots aquells passos que fem mentre estem gravant la macro.

Tots aquells passos que fem, quedaran definits com a codi de Visual Basic en un nou mòdul adjunt al llibre de càlcul.

Des de l'opció *Enregistra una macro* de la pestanya Desenvolupador, podrem crear les macros que necessitem.



Per crear una macro, serà necessari posar-li un nom, que es obligatori fer-ho sense espais en blanc i començant per una lletra.

La tecla de drecera ens permetrà executar la macro utilitzant la combinació de tecles que nosaltres assignem. Per exemple si posem la lletra r, la nostra macro la podrem executar utilitzant la combinació Ctrl + r, també podríem posar la lletra en majúscules, però llavors hauríem de fer per executar-la Ctrl + Mays + r.

La macro la podrem emmagatzemar a tres llocs diferents i depèn de on ho fem, podrà executar-se des d'un document o no.

Llocs per emmagatzemar les macros:

- **Aquest llibre de treball** – Només serà executable des del llibre de treball que l'hem creat (es la més utilitzada).

- **Llibre de macros personal** – Podrem executar la macro des de qualsevol llibre de treball. Les guarda a un llibre que es diu PERSONAL.XLSB. Aquest arxiu es troba guardat a:

C:\Users\NomUsuari\AppData\Roaming\Microsoft\Excel\XLSTART

NomUsuari – Es refereix a l'usuari amb el que estem treballant en la sessió oberta per Windows.

Per esborrar una macro guardada al llibre de macros personals, el millor es fer-ho des de la finestra de Visual Basic, eliminant directament el mòdul. El llibre està ocult i si ho intentem fer des de la finestra de las macros en Excel, ens avisarà que està ocult i no podrem eliminar-la. Per fer-ho, des de les opcions de Windows hauríem de fer que ens mostrés tots els arxius ocults o del sistema.

- **Llibre nou** – Només serà executable des de els llibres que estiguin basats en la plantilla del document en blanc, en 2010 aquesta funció ens fa desar un llibre, per tant no funciona correctament.

Es importantíssim, si volem conservar les macros creades en un document, que desem el llibre com a llibre habilitat per a macros en el moment de guardar el document.

Els llibres d'Excel en la versió 2010 es guarden amb l'extensió .xlsx, mentre que els llibres amb macros s'han de desar amb l'extensió .xlsm

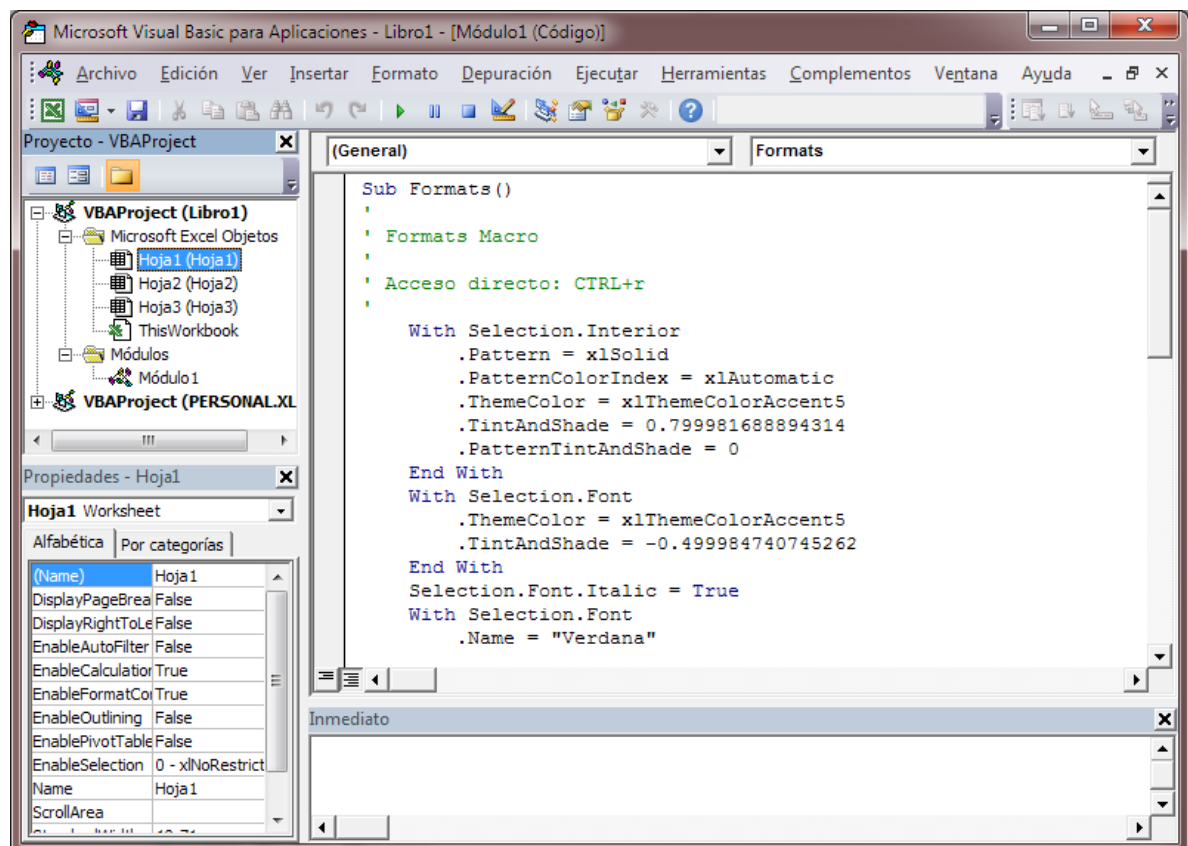
GESTIÓ DE LES MACROS

La modificació i eliminació d'una macro la podrem fer des de la pestanya *Desenvolupador*, amb l'opció *Macros*

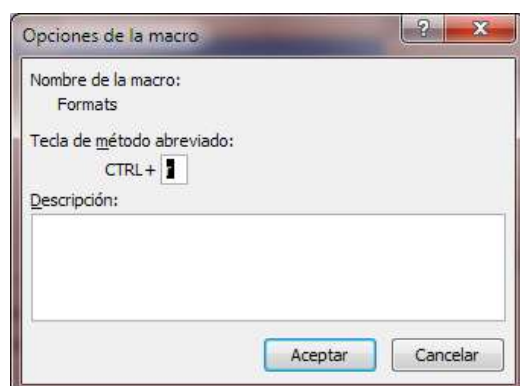


Una vegada estem dins del quadre de macros, des d'aquest podrem eliminar la macro o macros que hem generat i podrem modificar-les, però al fer Edita per fer-ho, veurem que ens porta directament a la finestra de codi amb Visual Basic per Aplicacions.

La forma de modificar una macro es només per codi, manualment. Es recomana, si no es tenen coneixements de programació o coneixements de codi en Visual Basic, i la macro es relativament senzilla, eliminar la macro i tornar-la a crear amb l'opció de gravació de macros.



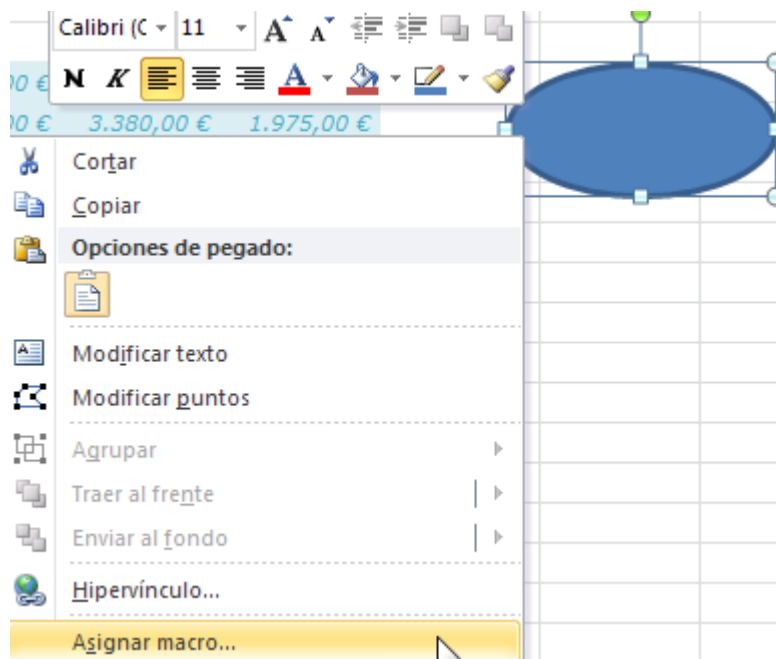
Amb l'opció: **Opcions**, podrem modificar la combinació de tecles que hem assignat a la macro i els comentaris que hem posat descrivint la funció de la macro.



ASSIGNACIÓ D'UNA MACRO A UN ELEMENT

Una vegada tenim creada la macro, podríem executar-la utilitzant una combinació de tecles o bé assignant aquesta macro a un element, ja sigui un botó, o qualsevol altre control dels que existeixen a la pestanya de desenvolupador. Però també ho podem fer a una figura qualsevol, per exemple una forma automàtica o una imatge.

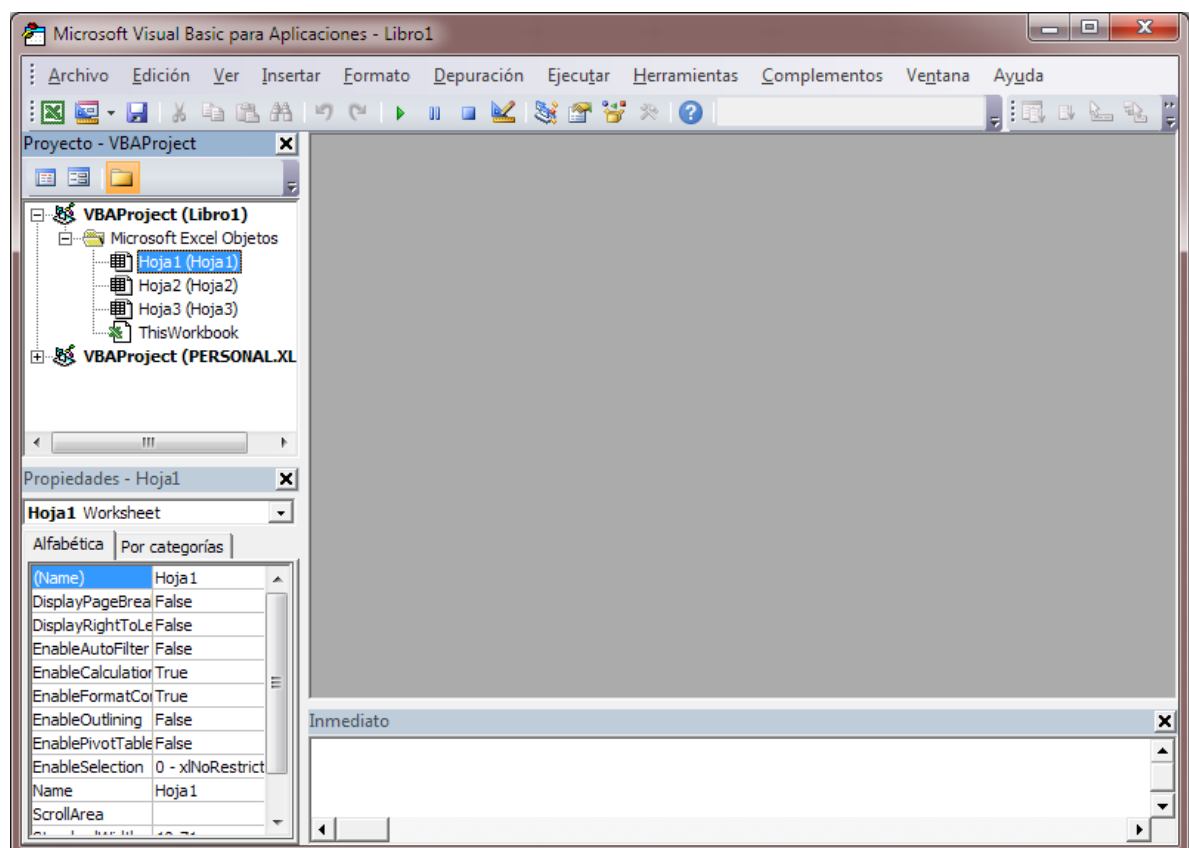
Per fer-ho, el primer que haurem de fer serà crear la forma i una vegada la tenim a dins del nostre full de càlcul, farem un clic amb el botó dret a sobre i farem clic a l'opció **assignar macro**. Això farà que se'ns obri un quadre des d'on podrem escollir la macro en qüestió.








2. VBA – VISUAL BASIC PER APLICACIONS






El llenguatge de programació Visual Basic es el que utilitzen les aplicacions d'office per programar les seves aplicacions. Ja sigui utilitzant macros o bé escrivint el codi de forma manual.

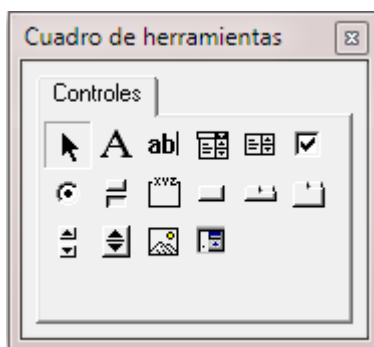
El editor de Visual Basic es una finestra independent de la pròpia aplicació, en aquest cas de Microsoft Excel, que conté totes les eines necessàries per programar en Visual Basic. Quan es tanqui l'aplicació també es tancarà la finestra de Visual Basic.



Detall del conjunt de botons específics de la finestra de VBA en la seva barra d'eines principal:

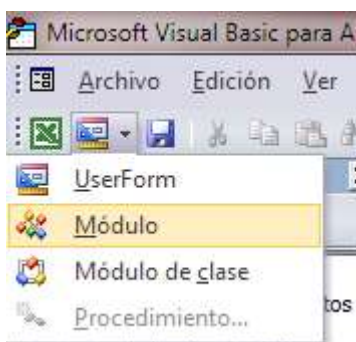
-  Tornar a l'Excel
-  Inserir elements: Formularis, mòduls, procediments
-  Executar procediment o formulari actiu.
-  Interrompre temporalment l'execució del procediment
-  Parar l'execució del procediment

- 
 Canvia entre la forma de codi i la vista disseny, només funciona quan està en execució el formulari
- 
 Mostra l'explorador de projectes si s'ha tancat
- 
 Mostra les propietats de l'objecte obert si s'ha tancat aquest panell. El objecte pot ser un formulari, un full, el llibre.
- 
 Mostra l'examinador d'objectes, una llista on es troben totes les propietats, mètodes i llibreries de visual basic.
- 
 Només està activat quan estem dissenyant un formulari, ens permet mostrar el quadre d'eines de disseny del formulari.



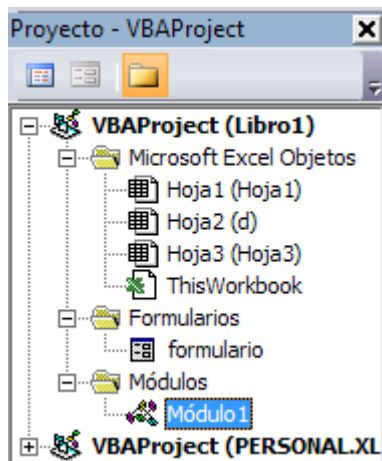
MÒDULS EN VBA

Son codis que es guarden en un llibre de treball d'Excel, i sempre podrem veure o editar el codi a través de l'editor de Visual Basic. Per afegir un mòdul, podrem anar a través de l'opció d'inserir elements a través de la barra d'eines i escollir l'opció Mòdul:



Això també ho podrem fer utilitzant el menú Insertar - Mòdul

A partir d'aquí es col·locarà a l'explorador de projectes el mòdul en qüestió.



En aquest mòdul, li podrem canviar el nom, bé amb el menú dret, utilitzant l'opció de canviar el nom o bé a través de les propietats del mòdul en qüestió.

Per eliminar un mòdul, podrem fer un clic amb el botó dret a sobre del mòdul i escollir l'opció **Quitar Mòdul...**

ELS COMENTARIS

Són línies de codi que Visual basic no té en compte quan executa el programa. Per indicar que es un comentari només haurem de posar un apòstrof al inici del comentari.

Els comentaris poden trobar-se en una línia sola o bé formar part de una línia de codi.

(General)
<pre> Option Explicit Function IVA(import, Perc_IVA) ' Funció que calcula un IVA segons els paràmetres passats ' Autor: Julià Moreno IVA = import * Perc_IVA / 100 ' Aquí escric la fórmula End Function </pre>

TIPUS DE PROCEDIMENTS

Son codis que executen una acció. Existeixen dos tipus de procediments, els procediments pròpiament dits (Sub) i les funcions (Function)

PROCEDIMENTS (SUB)

Son independents per si mateixos, es poden executar bé, a través d'una acció o bé a través de cridar-los des d'un altra procediment.

FUNCIONS (FUNCTION)

Sempre tornen un valor, després de la realització de una o varies operacions, aquestes funcions podran contenir paràmetres que seran allò que necessitarà la funció per executar-se correctament.

Les funcions les podrem cridar a través d'una altra procediment, i si ho fem així, sempre hauran de forma part d'una expressió. Es a dir, hauran d'estar igualats a una variable perquè aquesta prengui el valor resultant.

També les podrem cridar com si fossin una de les funcions que incorpora Microsoft Excel per defecte.

Per exemple, aquest funció calcularia l'IVA sobre un import:

```
Function IVA(import, Perc_IVA)
  IVA = import + (import * Perc_IVA / 100)
End Function
```

Des de l'Excel es cridaria, suposant que en la cel·la A1 hi ha la quantitat 1.000 € i en la cel·la B1 hi ha 21:

=IVA(A1;B1)

Des de codi, l'hauríem de cridar formant part d' una expressió, per exemple:

```
Msgbox IVA(1000,21)
```

O

```
calcul_iva = IVA(1000,21)
```

OBJECTES I COL·LECCIONS

Un objecte és una combinació de codi i dades que pot tractar-se com a una unitat, pot ser una porció d'una aplicació, com un botó, un element qualsevol o inclòs un programa es podria considerar un objecte.

Cada objecte en Visual Basic està definit per una classe, que descriu les propietats i mètodes d'un objecte. Un objecte es el que se'n diu una instància de classe.

Per posar un exemple de la vida real, imaginem que la nostra classe es diu vehicle i aquest té entre d'altres propietats, el color, els cavalls, el número de rodes, etc., i els seus mètodes podrien ser per exemple l'acció de frenar, la d'accelerar, si no té benzina que ens avisi, l'acció d'encendre les llums, etc.

Tot i que Excel ja portarà moltes classes creades, podrem crear les nostres pròpies classes utilitzant l'opció Insertar - Mòdul de Classe.

En el nostre cas, Excel podria considerar-se un objecte i aquest podria contenir un altra objecte que es diu llibre i el llibre conté un altra que es diu full...

Les col·leccions son conjunts de objectes.

Per exemple podríem referenciar la fulla Trimestre 1 del llibre Treball

```
Workbooks("Treball").Worksheets("Trimestre 1")
```

Però si una vegada ho tenim referenciat no li diem que ha de fer o que ha d'agafar, no aconseguirem res. Això ho haurem de fer a través de les seves propietats o mètodes.

PROPIETATS I MÈTODES

Propietats son les característiques de l'objecte, mentre que mètodes son les ordres que li podrem donar a l'objecte.

Imaginem que volem agafar el valor que hi ha en l'objecte referenciat anteriorment, podríem fer

```
Workbooks("Treball").Worksheets("Trimestre 1").Range("A1").value
```

D'aquesta forma estarem utilitzant la propietat value de l'objecte Rang A1 del full Trimestre 1 del nostre llibre Treball.

Si pel contrari volguéssim seleccionar la cel·la, podríem utilitzar l'ordre (mètode) SELECT.

```
Sheets("Trimestre 1").Select  
Range("A1").Select
```

Hem de tenir en compte que algunes propietats són només de lectura, per tant no es podran canviar, només consultar per saber el valor que tenen en aquell moment.

LES VARIABLES I LES CONSTANTS

Una variable és un espai de memòria reservat per guardar un valor que correspon a un tipus de dades suportat pel llenguatge de programació. Una variable es representa i usada a través d'una etiqueta (nom) que li assigna el programador o que si es una variable intrínseca ja ha estat assignada pel propi programa.

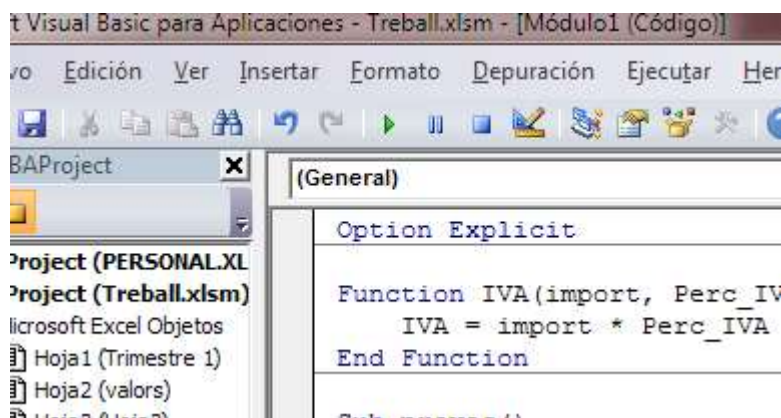
Una constant, al igual que les variables es un valor que s'assigna a un espai de memòria a través d'un nom. Però la diferència es que no pot ser alterat durant l'execució de l'aplicació. Un exemple molt clar de constant seria el valor PI, que sempre haurà de ser el mateix.

La seva utilització haurà de seguir les següents normes:

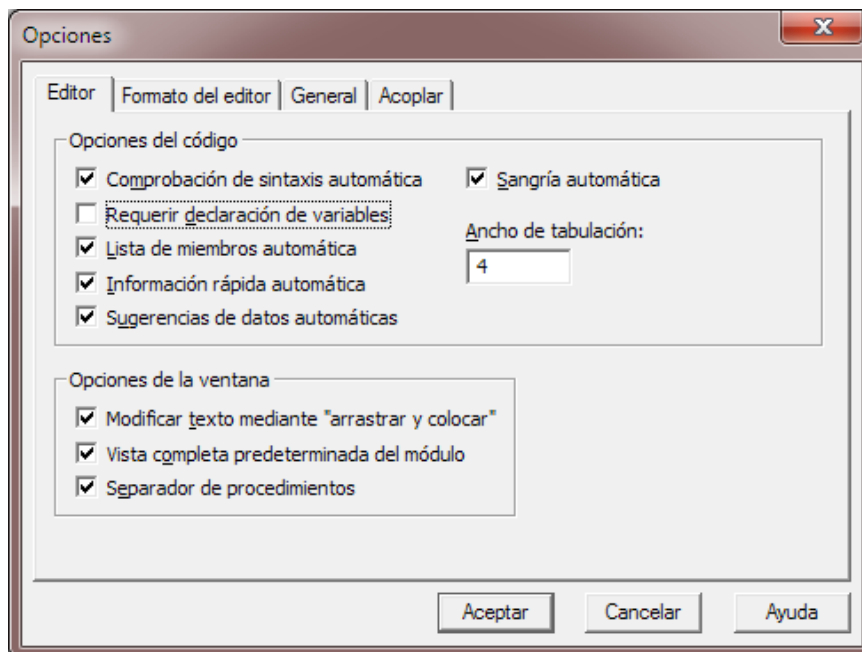
- El nom de la nostra variable ha de començar per una lletra.
- No pot tenir més de 250 caràcters
- No es pot utilitzar paraules reservades d'Excel com per exemple Range, select, etc.
- No pot contenir espais en blanc.

Les variables en Excel no es necessari definir-les a no ser que se li hagi dit expressament al programa que així ho farem. Per obligar a la definició de variables amb Excel, haurem d'activar l'opció: Option Explicit

Per activar-ho podrem fer-ho manualment col·locant directament aquesta ordre al inici de cada mòdul:



o activar a través *Herramientas – Opciones* i activar l'opció *Requerir declaración de variables* la declaració de variables, això només s'efectuarà en els mòduls creats a partir d'aquell moment:



DECLARACIÓ DE VARIABLES

Les variables poden ser:

- Numèriques (integer, long, single, double, etc)
- Text (String)
- Data / hora (Date)
- Lògiques (boolean)
- Objectes

I es declararan utilitzant la sentència DIM

Dim a As String	\ Estem declarant la variable a com a text
Dim fecha As Date	\ Estem declarant la variable fecha com a data
Dim numero as integer	\ Estem declarant la variable numero com a enter
Dim rang As Range	\ Estem declarant la variable com a objecte Range

Per assignar valor a una variable, únicament haurem d'igualar el valor a la variable:

Exemple:

a = "Hola Joan"	
fecha = Date	\ Date es una funció de VBA que torna el valor de la data del sistema
numero = 5	

Però en les variables del tipus objecte, s'ha d'assignar el valor utilitzant la sentència Set:

```
' Assignem a la variable rang, els valors d'un nom de rang d'Excel
Set rang = Range(ThisWorkbook.Names("totals"))
```

QUADRES DE MISSATGES

La instrucció MsgBox ens permet crear quadres de missatge:

Msgbox "Hola Mundo"

Mostrarà

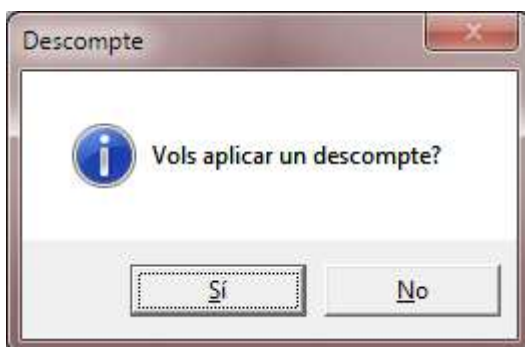


D'una altra banda, si el que volem es poder mostrar un missatge que ens demani alguna cosa i poder guardar el valor per utilitzar-lo més endavant, haurem de utilitzar MsgBox en una expressió:

```
Sub descomptes()
  Dim dte As Integer ' Declaro la variable Dte per poder guardar el valor del quadre msgbox
  dte = MsgBox("Vols aplicar un descompte?", vbYesNo + vbInformation, "Descompte")
End Sub
```

vbYesNo i **vbInformation** –Funcions internes de VB que representen el que volem mostrar en el quadre de missatge, per tant **vbYesNo**, farà que apareguin els dos botons Sí i No; i **vbInformation**, farà que aparegui el logo d'informació de la part de l'esquerra del quadre

Mostrarà:



En cas que premem el botó Sí, guardarem en la variable dte el valor 6, pel contrari si premem No, es guardarà el valor 7

SENTÈNCIES CONDICIONALS

Els condicionals permeten realitzar comparacions entre diferents valors, a la fi de poder executar un codi o un altre segons el resultat de la comparació.

EL CONDICIONAL IF

Sintaxi

```
If condicio [ Then ]  
  [ instruccions ]  
[ ElseIf elseif condicio [ Then ]  
  [ instruccions ] ]  
[ Else  
  [ instruccions ] ]  
End If
```

O

```
If condicio Then [ instruccions (una línia)] [ Else [ instruccions (una línia)] ]
```

Exemple:

```
If a >= 5 Then  
  MsgBox "El número introduït es més gran o igual que 5"  
Else  
  MsgBox "El número introduït es més petit que 5"  
End If
```

SELECT ... CASE

Quan tenim que comparar moltes condicions d'igualació utilitzarem aquesta sentència, que ens permetrà fer més comparacions de forma més còmode

Sintaxi:

```
Select [ Case ] expressió  
  [ Case valor  
    [ instruccions ] ]  
  [ Case Else  
    [ instruccions ] ]  
End Select
```

Exemple:

```
Sub numeros()  
  
    Dim numero As Integer  
  
    ` Inputbox, permet entra un valor a través d'un quadre de diàleg.  
    numero = InputBox("Entra un número: ")  
  
    Select Case numero  
        Case 1 To 5  
            MsgBox "Está entre 1 i 5, inclòsiu"  
        Case 6, 7, 8  
            MsgBox "Está entre 6 i 8, inclòsiu"  
        Case 9 To 10  
            MsgBox "És igual a 9 o a 10"  
        Case Else  
            MsgBox "El número no està entre 1 i 10, inclòsiu"  
    End Select  
  
End Sub
```

SENTÈNCIES ITERATIVES

FOR ... NEXT

Repeteix una sèrie d'instruccions fins a un límit donat

Sintaxi:

```
For valor_inicial To límit  
    Instruccions  
Next
```

Farem que conti fins a 1000 i que ho vagi col·locant a una cel·la successivament:

```
Range("A1").select  
For a=1 To 1000  
    Cells(a, 1).Value = a  
next
```

DO WHILE ... LOOP

Aquesta sentència repeteix les instruccions que hi posem en el seu interior mentre la condició sigui verdadera (TRUE)

Sintaxi

```
Do while condició
  Instruccions
Loop
```

Imaginem que volem buscar en un rang de cel·les, la primera cel·la que contingui un valor més gran que 5000, i que vagi col·locant al costat una N, en tots aquells que vagi trobant més petits. Podríem fer això:

```
Sub cincmil()

  i = 1
  Do While Cells(i, 1) < 5000
    Cells(i, 2).Value = "S"      ` Posa una S a la cel·la del costat
    Cells(i, 1).Offset(0, 0).Interior.Color = RGB(50, 100, 100)  ` color de fons de la cel·la
    i = i + 1
  Loop
  MsgBox "El valor se encontró en fila no. " & i

End Sub
```

WEBGRAFIA

http://www.youtube.com/watch?v=eX2WeA_8KnU

<http://www.youtube.com/watch?v=s1bpG7f49Z4>

<http://www.youtube.com/watch?v=EgllEo6ag7Y>

[http://es.wikipedia.org/wiki/Constante_\(inform%C3%A1tica\)](http://es.wikipedia.org/wiki/Constante_(inform%C3%A1tica))

http://es.wikipedia.org/wiki/Variable_en_programaci%C3%B3n

<http://cursovisualexcel.wordpress.com/2011/07/21/los-datos-en-vba-variables-y-constantes/>

[http://msdn.microsoft.com/es-es/library/752y8abs\(v=vs.90\).aspx](http://msdn.microsoft.com/es-es/library/752y8abs(v=vs.90).aspx)

[http://msdn.microsoft.com/es-es/library/cy37t14y\(v=vs.90\).aspx](http://msdn.microsoft.com/es-es/library/cy37t14y(v=vs.90).aspx)

http://www.xltoday.net/vba_bucles_dowhile-loop-asp