

Capítulo 8

Visual Basic para aplicaciones

Continuar

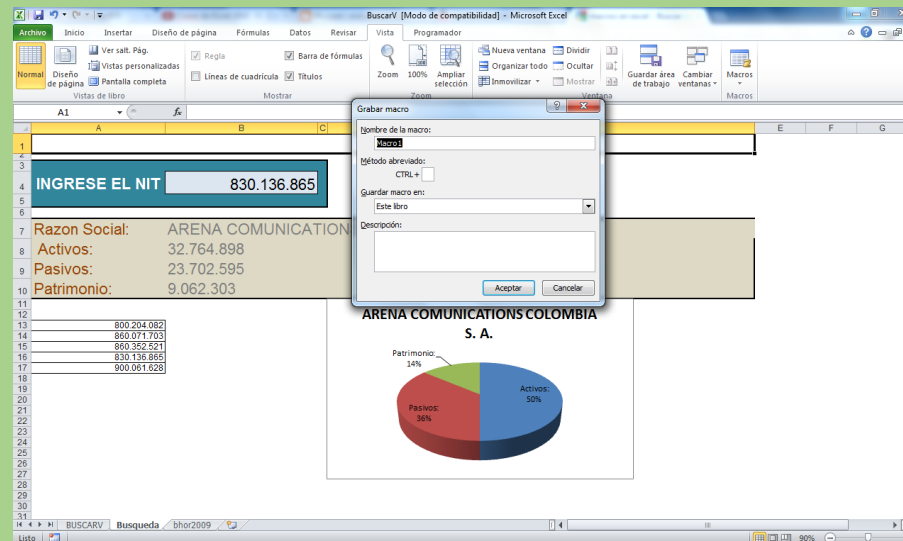
Visual Basic para aplicaciones

Visual Basic para Aplicaciones[®] (VBA) es el lenguaje de programación que Microsoft[®] incorpora en la mayoría de sus productos de Office[®] 2013, e incluso algunas otras marcas lo han incorporado a sus aplicaciones. Una gran ventaja de conocer y manejar VBA es que nos permite manejar otros programas con el mismo lenguaje gracias a su modelo de manejo de objetos.



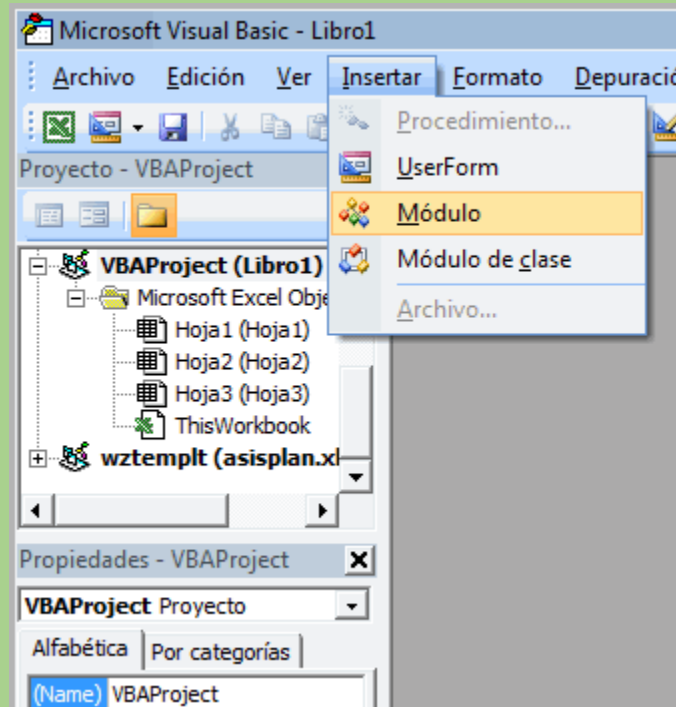
Macros

Las macros son programas que desarrollamos dentro del Editor de Visual Basic® para automatizar uno o varios procesos. Los programas son un conjunto de instrucciones ordenadas lógicamente para la realización de una o varias tareas.



Módulo

Los módulos de Visual Basic para Aplicaciones® son nuestra área de trabajo para el desarrollo de macros y se guardan dentro de los libros.



Códigos

Son secuencias de instrucciones que se desarrollan dentro de los módulos de Visual Basic para Aplicaciones®.

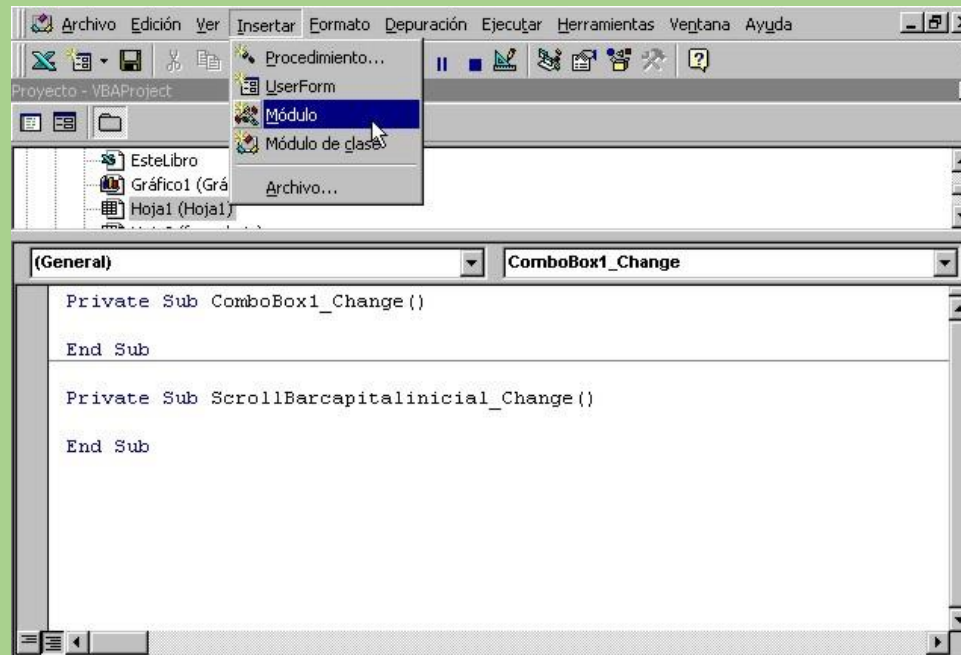
```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Me.BackgroundImage = WindowsApplication1.My.Resources.Resources.fondo1
    Me.BackgroundImageLayout = ImageLayout.Stretch
End Sub

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
    Me.BackgroundImage = WindowsApplication1.My.Resources.Resources.fondo2
    Me.BackgroundImageLayout = ImageLayout.Stretch
End Sub

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click
    Me.BackgroundImage = WindowsApplication1.My.Resources.Resources.fondo3
    Me.BackgroundImageLayout = ImageLayout.Stretch
End Sub
```

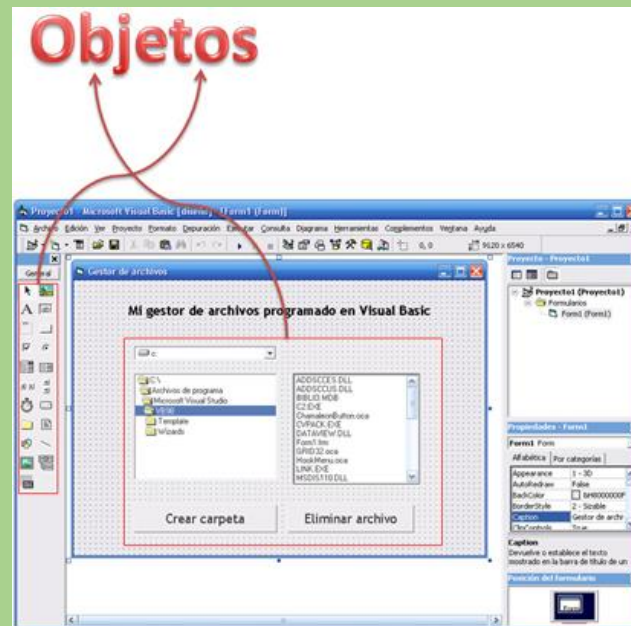
Procedimientos

Son rutinas que se elaboran a base de código en un módulo que realiza ciertas acciones. Hay dos tipos de procedimientos: Sub y Function.



Objetos

Cualquier elemento que exista dentro de Excel® es considerado un objeto. Por ejemplo, Excel® es un objeto llamado Application que contiene otros objetos, como hojas de cálculo, y las hojas de cálculo pueden tener un objeto gráfico o una forma, y así sucesivamente. A la forma de trabajo aquí descrita se le conoce como manejo de objetos.



Colecciones

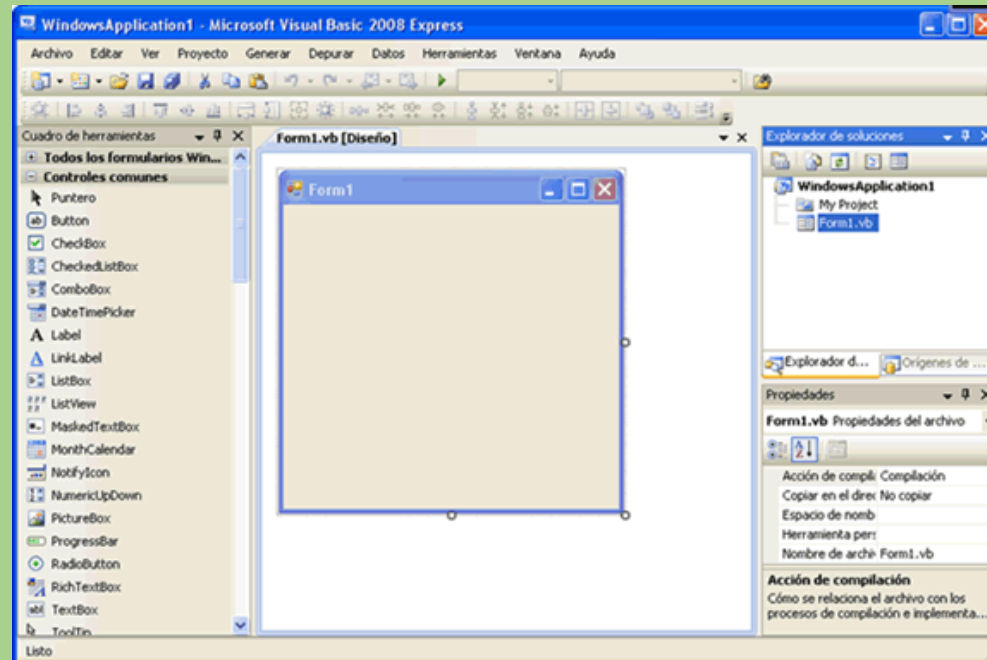
Son propiedades de los objetos que se pueden agrupar para que su manejo sea más fácil de programar.

```
'Anteriormente: Creando una lista de tipo List(Of
Integer)
Dim coleccionint1 As New List(Of Integer)
coleccionint1.Add(1)
coleccionint1.Add(2)
coleccionint1.Add(3)
coleccionint1.Add(4)
coleccionint1.Add(5)
coleccionint1.Add(6)
coleccionint1.Add(7)

'Anteriormente: Inicializamos una colección de tipo
Dictionary
Dim coleccionDic1 As New Dictionary(Of Integer, String)
coleccionDic1.Add(1, "Uno")
coleccionDic1.Add(2, "Dos")
coleccionDic1.Add(3, "Tres")
```

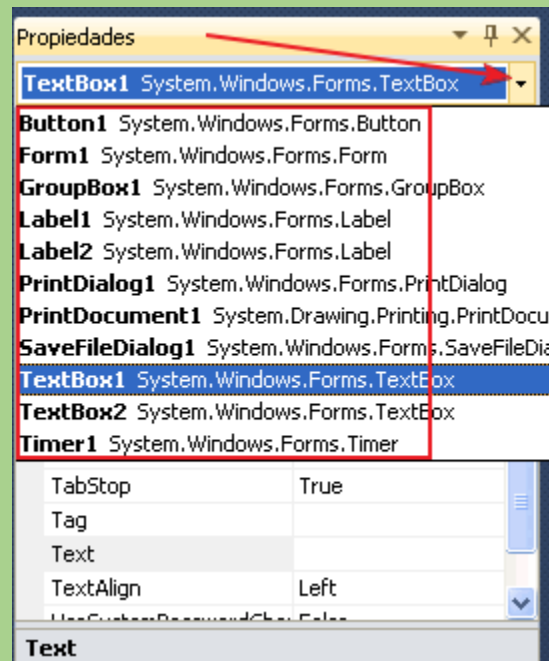

Objeto activo

Es el objeto en que se trabaja. Esto es muy útil, porque permite omitir líneas de código.



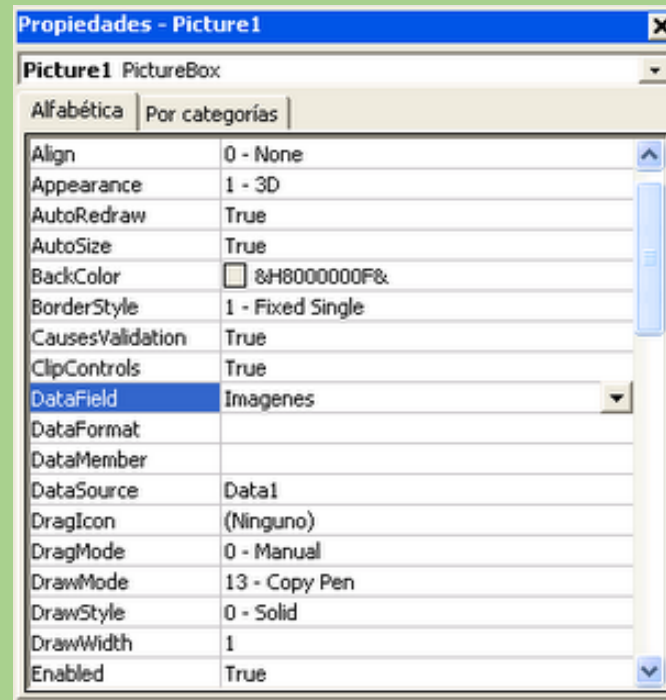
Propiedades de los objetos

Son atributos del objeto que definen sus características, como tamaño, color o ubicación en pantalla.



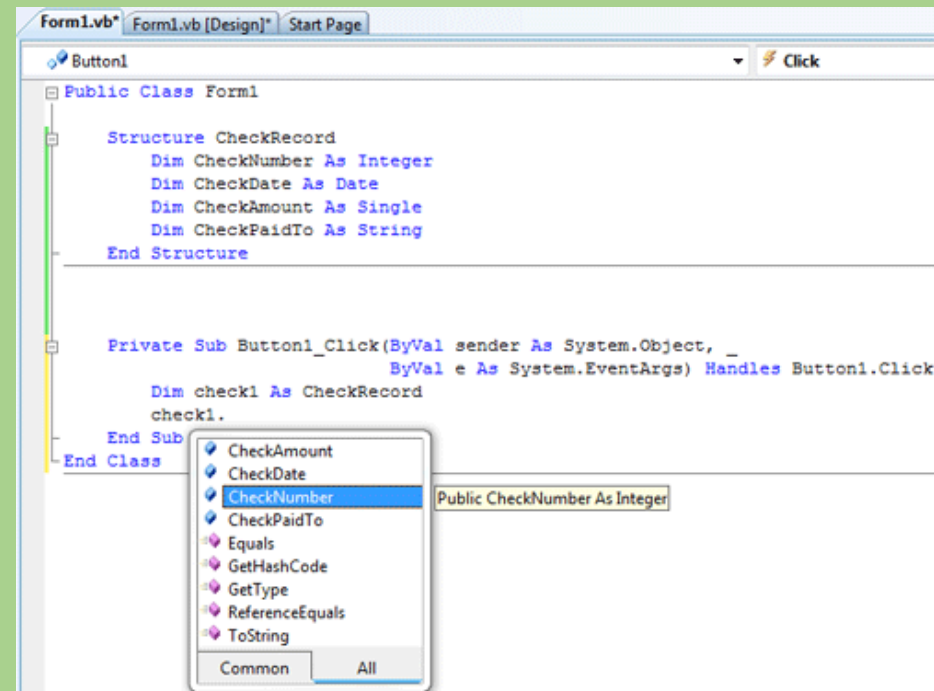
Métodos de los objetos

Detonan una acción en estos; por ejemplo, se puede eliminar el contenido de una celda.



Variables

Son almacenamientos temporales para guardar información y usarla en cualquier momento dentro de la macro.



```
Form1.vb* Form1.vb [Design]* Start Page
Button1 Click
Public Class Form1
    Structure CheckRecord
        Dim CheckNumber As Integer
        Dim CheckDate As Date
        Dim CheckAmount As Single
        Dim CheckPaidTo As String
    End Structure

    Private Sub Button1_Click(ByVal sender As System.Object, _
        ByVal e As System.EventArgs) Handles Button1.Click
        Dim check1 As CheckRecord
        check1.
    End Sub
End Class
```

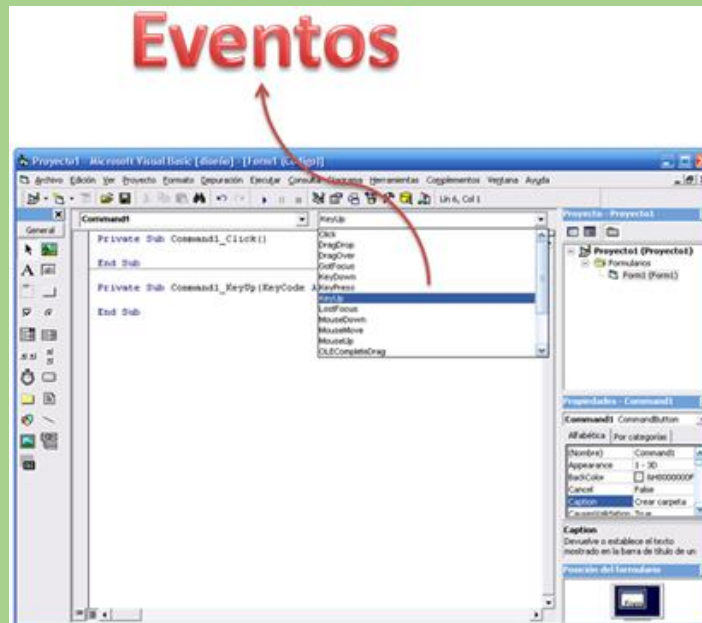
Public CheckNumber As Integer

- CheckAmount
- CheckDate
- CheckNumber
- CheckPaidTo
- Equals
- GetHashCode
- GetType
- ReferenceEquals
- ToString

Common All

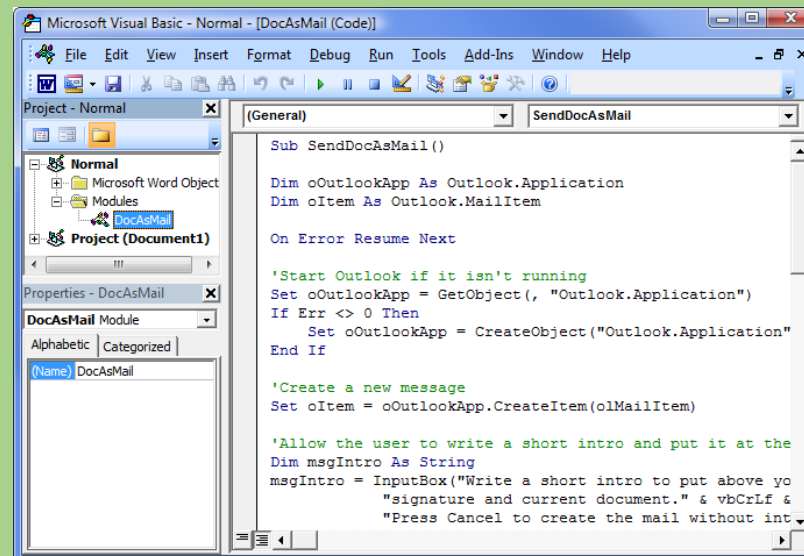
Eventos

Son disparadores automáticos de macros a través de algún tipo de cambio detectado en un objeto; por ejemplo, se puede ejecutar una macro cuando alguien seleccione una hoja en particular.



Editor de Visual Basic

El Editor de Visual Basic® es donde se desarrollan las macros, y es una aplicación que está pegada con Microsoft® Excel® 2013, lo que significa que no se puede ejecutar el Editor si no se está ejecutando Excel®. Cuando necesitamos el Editor, oprimimos las teclas [Alt] + [F11]. El Editor de Visual Basic® es nuestra área de trabajo para el desarrollo de las macros.



Ficha desarrollador

Otra forma de acceder al Editor de Visual Basic® es desde la ficha Desarrollador. Si la etiqueta Desarrollador no está activada, se puede activar en la etiqueta Archivo > Opciones > Personalizar cinta de opciones. En el cuadro derecho, donde dice Pestañas principales, también se encuentra Desarrollador.

