



aprenderaprogramar.com

Ejemplo de programa básico en Visual Basic. Option Explicit, Form, Dim, String, etc. (CU00310A)

Sección: Cursos

Categoría: Curso Visual Basic Nivel I

Fecha revisión: 2029

Autor: Mario R. Rancel

Resumen: Entrega nº9 del Curso Visual Basic Nivel I

29

CREAR UN PROGRAMA BÁSICO

Vamos a crear un programa que declare una variable tipo String llamada mensaje y nos muestre un texto en pantalla.

Para ello como primer paso abrimos un proyecto de Visual Basic. Recordar que esto se hace de la siguiente manera. En el menú Archivo seleccionamos la opción “Nuevo proyecto”. A continuación elegimos:

- a) Para las versiones que nos dan la opción “Aplicación de Windows Forms” elegimos esta opción.
- b) Para las versiones que nos dan la opción “Exe estándar” elegimos esta opción.

Una vez abierto el proyecto debe aparecernos el formulario vacío sobre la pantalla. Hacemos doble click sobre el formulario y se nos debe abrir la ventana de código conteniendo algunas líneas. En caso de que no se abra la ventana de código, vete al menú “Ver” y elige la opción “Código”.

Escribiremos el siguiente código que corresponde al programa que vamos a ejecutar:

Para las versiones menos recientes:

```
Rem Curso Visual Basic aprenderaprogramar.com
Option Explicit
Dim mensaje As String

Private Sub Form_Load()
mensaje = "Bienvenido a este programa"
MsgBox (mensaje)
End Sub
```

Para las versiones más recientes:

```
REM Curso Visual Basic aprenderaprogramar.com
Option Explicit On
Public Class Form1
    Dim mensaje As String
    Private Sub Form1_Load(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles MyBase.Load
        mensaje = "Bienvenido a este programa"
        MsgBox(mensaje)
    End Sub
End Class
```

Ten en cuenta que según la versión de Visual Basic que estés utilizando pueden existir pequeñas diferencias en el código que debe usarse. Nosotros pretendemos centrarnos en la lógica de la programación más que en los detalles de la sintaxis de una versión concreta. Si es necesario, consulta la ayuda para conocer la sintaxis exacta que debes emplear.

Analicemos las líneas que lo componen.

1. Option Explicit (Option Explicit On). Ya hemos definido su significado.
2. Dim mensaje as String, declara una variable denominada mensaje como tipo alfanumérico de longitud variable.
3. Private Sub Form Load() ó Private Sub Form1_Load (...). Esta parte de código corresponde a la apertura de un módulo de código de Visual Basic, que en la terminología del lenguaje se denomina

Procedimiento. La cuestión se puede hacer un poco complicada en lo que se refiere a terminología, pero no vamos a preocuparnos por esta cuestión ahora.

En el curso “Bases de la programación I” de aprenderaprogramar.com se trabaja con programas que constan de un algoritmo principal y distintos módulos o subprogramas. La estructura y terminología de un programa en Visual Basic es algo más compleja, al trabajarse con distintos elementos dentro de un programa. Estos elementos varían según la versión, pero podrían ser por ejemplo módulos de formulario (.frm), módulos estándar (.bas), módulos de clases (.cls), etc.. Nosotros vamos a centrarnos en lo que es un módulo de formulario: un fichero en el que almacenamos información relacionada con el formulario donde hemos colocado objetos y para el que hemos escrito un código. Así pues, de momento para nosotros el programa va a constar de:

- Un formulario donde hemos colocado objetos (trabajo de diseño de tipo gráfico). Excepcionalmente un formulario puede carecer de objetos y encontrarse vacío.
- Un código que consta normalmente de: Option Explicit (Option Explicit On), Declaración de variables y Procedimientos. Un procedimiento es un conjunto de código que se ejecutará en un momento dado, algo equivalente a lo que habíamos definido con el pseudocódigo como módulo o subprograma.

Un procedimiento puede ejecutarse por varios motivos como:

- a) Ser llamado desde alguna parte del programa.
- b) Se produce un evento que da lugar a que se ejecute el código.

En este ejemplo que estamos viendo, el procedimiento se ejecuta cuando tiene lugar un evento que es la carga del formulario (form load), que vamos a considerar la ejecución del programa. Así pues, el código comprendido entre el fragmento de código de tipo “Private Sub Form_Load()” y “End Sub” se va a ejecutar cada vez que mandemos correr el programa, es decir, cada vez que ordenemos su ejecución.

4. mensaje = "Bienvenido a este programa", asigna a la variable mensaje el contenido de tipo alfanumérico indicado.
5. MsgBox (mensaje), indica al ordenador que proceda a mostrar el contenido de la variable mensaje dentro de una caja de mensajes.
6. End Sub, define el final del módulo de código (procedimiento).

Ejecuta el programa pulsando la tecla F5. También puedes hacerlo a través del menú <<Ejecutar / Iniciar>> ó <<Depurar / Iniciar la depuración>> según la versión que estés utilizando. En algunas versiones existe la opción <<Ejecutar / Iniciar con compilación completa>>, que permite que la ejecución del programa se haga realizando una revisión de errores más exhaustiva que si se ejecuta el programa pulsando simplemente F5.

Tras ejecutar el programa te aparecerá en pantalla el mensaje Bienvenido a este programa. Para cerrar el programa pulsa aceptar y haz click sobre el aspa de cierre de la ventana. Vamos a hacer una pequeña mejora estética. Vete a Ver-Diseñador (Ver-Objeto) y cambia el valor de la propiedad Caption ó Text (según la versión que estés usando) del form a "Bienvenida" y modifica el código de la siguiente manera.

Para las versiones menos recientes:

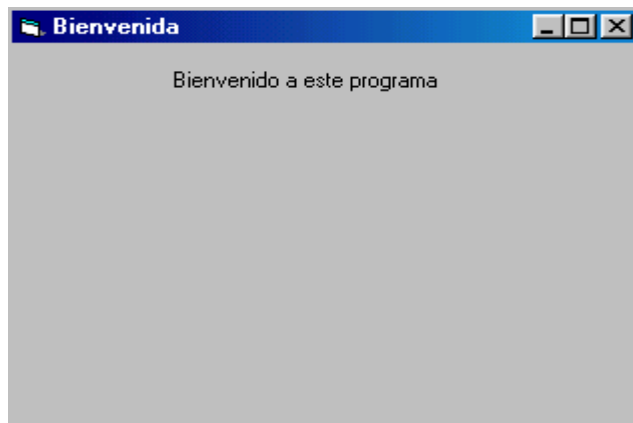
```
Rem Curso Visual Basic aprenderaprogramar.com
Option Explicit
Dim mensaje As String
Dim espacios As String
Private Sub Form_Load()
    espacios = vbTab
    mensaje = espacios & "Bienvenido a
este programa" & espacios
    MsgBox (mensaje)
End Sub
```

Para las versiones más recientes:

```
REM Curso Visual Basic aprenderaprogramar.com
Option Explicit On
Public Class Form1
    Dim mensaje As String
    Dim espacios As String

    Private Sub Form1_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
        espacios = vbTab
        mensaje = espacios & "Bienvenido a este
programa" & espacios
        MsgBox(mensaje)
    End Sub
End Class
```

Fíjate que cuando aparezca la ventana del formulario, debido a que hemos modificado sus propiedades, ahora nos aparecerá el texto "Bienvenida" en la parte superior.



También hemos usado otros elementos que iremos explicando poco a poco a lo largo del curso. El objetivo ahora es simplemente irnos familiarizando con Visual Basic.

La constante vbTab nos ha servido para introducir un tabulador. Ello da lugar a que el texto aparezca con un pequeño margen por el lado izquierdo y derecho sobre la ventanita del mensaje. Prueba a introducir variables de tipo numérico y a dar lugar a que se muestren sus contenidos sobre la ventana del mensaje. Juega con los nombres de las variables, sus contenidos y la asignación de contenidos.

También puede resultar de interés que compruebes qué ocurre si declaras una variable de un tipo y le asignas contenido de otro tipo. Por ejemplo declara *Dim salario As Integer*, asígnale el contenido *salario = "Son 3000 euros"* y prueba a mostrarlo en pantalla.



¿Qué ocurre cuando incumplimos las previsiones de *Visual Basic* para asignar contenidos a variables (por ejemplo, para una variable A tipo Integer definir $A = 5320000000000000000000$, que está fuera del rango previsto)? No vamos a analizar los distintos casos que se pueden presentar, sino a tratar de dar una respuesta genérica. Cuando hacemos algo no esperado, como asignar un valor fuera de rango, asignar un valor que no concuerda con el tipo de la variable, sumar variables numéricas con alfanuméricas, asignar decimales a un número entero..., etc. pueden suceder varias cosas:

- Salta un error. Un mensaje impide que el programa comience a ejecutarse, o bien el programa se detiene mientras se está ejecutando.
- El programa se ejecuta pero los resultados no son los deseados.
- El programa se ejecuta y los resultados son los deseados.

En resumen, es difícil prever lo que va a suceder, aunque se pueden estudiar y manejar las circunstancias. Por ejemplo una variable que se declare como tipo Long pero a la que se asigna un contenido numérico real con dos decimales no da lugar a un error, pero sí a una pérdida de información derivada de que el número decimal se va a redondear a un entero. Las consecuencias de esta circunstancia habría que valorarlas para cada programa y circunstancias concretas.

Como programadores hemos de buscar programas 100% predecibles, en los que no se pueda producir que "sea difícil prever lo que va a suceder". Por tanto intentaremos que la declaración y asignación de contenidos a variables se ajuste a las normas de *Visual Basic*. En última instancia, podremos prever una rutina de gestión de errores para casos imprevistos.

Próxima entrega: CU00311A

Acceso al curso completo en [aprenderaprogramar.com](http://www.aprenderaprogramar.com) -- > Cursos, o en la dirección siguiente:
http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=37&Itemid=61