



aprenderaprogramar.com

Tipos de variables en Visual Basic (integer, single, double, string, object, etc.). Ejemplos. (CU00308A)

Sección: Cursos

Categoría: Curso Visual Basic Nivel I

Fecha revisión: 2029

Autor: Mario R. Rancel

Resumen: Entrega nº7 del Curso Visual Basic Nivel I

VARIABLES CON VISUAL BASIC

Si has estudiado pseudocódigo y tipos de variables con pseudocódigo previamente los conocimientos adquiridos serán válidos y útiles con Visual Basic, pero con matices. Vamos a tratar los aspectos más básicos de la declaración de variables con Visual Basic y a dar algunas orientaciones a través de las que profundizar en el manejo de variables.

NOMBRES Y TIPOS DE VARIABLES. DECLARACIÓN. Option Explicit.

Para establecer nombres de variables puedes ver las indicaciones que se dan en el curso “Bases de la programación Nivel I” de aprenderaprogramar.com, apartado “Nombres de variables”. Visual Basic permite no declarar una variable antes de ser usada. Por tanto, en cualquier punto de un programa podríamos escribir algo similar a “Mostrar velocidad01”, sin necesidad de haber declarado la variable velocidad01. ¿Qué nos mostrará si no hemos asignado contenido a la variable velocidad01? Visual Basic es de los lenguajes que asignan automáticamente contenido cero, vacío o falso a una variable, dependiendo de qué tipo sea. Por tanto, se nos mostraría cero, “ ”, o falso. Conocido que no es estrictamente obligatorio, nosotros vamos a trabajar con Visual Basic realizando declaración de variables siempre. Los motivos para ello ya han sido expuestos, pero podemos resumirlos en:

- Generar buenos programas.
- Evitar errores y confusiones.

El propio Visual Basic dispone de una instrucción que obliga a tener que declarar todas las variables, denominada Option Explicit. La obligación de declaración de variables puede establecerse de dos maneras.

- a) En la ventana de código escribir Option Explicit (en las versiones más modernas debe escribirse Option Explicit On).
- b) En el menú Herramientas, selecciona Opciones... y en la pestaña Editor (en otras versiones Herramientas, Opciones, Proyectos y Soluciones, Valores predeterminados de VB) establece (o comprueba que está establecida) la opción Requerir declaración de variables como activa ó Option Explicit On.

Ten en cuenta que la opción a) es manual, y tendríamos que escribir Option Explicit cada vez que fuéramos a programar, mientras que la opción b) es automática y va a hacer que cada vez que arranquemos Visual Basic nos obligue a declarar variables sin tener que estar escribiendo código. Por tanto, activa la opción automática (o comprueba que esté activada) y así evitaremos olvidos. No vamos a volver a incidir en esto, pero siempre vamos a considerar que trabajamos con Option Explicit activado. ¿Qué ocurrirá si utilizamos en el código una variable que no ha sido declarada? En este caso Visual Basic impedirá la ejecución del programa mostrando el mensaje de error: “Error de compilación. Variable no definida / no declarada” así como la línea de programa donde ha aparecido la variable no declarada.

Siempre que exista al menos una variable no declarada ocurrirá esto, lo cual nos obliga a declarar todas y cada una de las variables.

En cuanto a los tipos de variables, Visual Basic las clasifica en función del tipo de dato que contienen. Veamos las más usuales.

TIPO DE DATOS	ABREVIATURA	MEMORIA REQUERIDA	RANGO DE VALORES	EQUIVALENCIA CON USADAS EN PSEUDOCÓDIGO	OBSERVACIONES
Integer (Entero)	%	2 bytes ó 4 bytes según versión	- 32768 a 32767 ó - 2.147.483.648 a 2.147.483.647 según versión	Entero	Uso en contadores, control de bucles etc.
Long (Entero largo)	&	4 bytes ó 8 bytes según versión	- 2147483648 a 2147483647 ó - 9,2E+18 a 9,2E+18 según versión	Entero	Igual que integer pero admite un rango más amplio
Single (Decimal simple)	!	4 bytes	- 3,4·10 ³⁸ a 3,4·10 ³⁸	Real	Hasta 6 decimales. También admite enteros
Double (Decimal doble)	#	8 bytes	- 1,79·10 ³⁰⁸ a 1,79·10 ³⁰⁸	Real	Hasta 14 decimales. También admite enteros
Boolean (Booleano)	No tiene	2 bytes	True o False	Booleano	False es el valor predeterminado y va asociado a cero
String	\$	10 bytes + 1 byte por cada carácter	0 a 2·10 ⁹ caracteres	Alfanumérica	Texto de longitud variable
Variant (Object en otras versiones)	No tiene	Variable	Los del tipo double para números o los del tipo string para texto	No tiene	Admite números enteros, decimales o texto



Existen otros tipos de variables. Algunos tipos son comunes a todas las versiones de Visual Basic mientras que otros tipos sólo están disponibles en algunas versiones. Otros tipos que pueden aparecer en Visual Basic son el tipo *Byte*, *Char*, *Date*, *Decimal*, *Object*, *Sbyte*, *Short*, *UInteger*, *Ulong*, *Ushort*, *Currency*, *String* de longitud fija o las variables definidas por el Usuario a través de *Type* ó *User-Defined*. Ahora no te preocupes por conocer todos los tipos de Visual Basic, te irás familiarizando con ellos poco a poco a lo largo del curso. Puedes acceder a información sobre los tipos de variables a través de la ayuda, buscando “Tipos de datos”, “Resumen de tipos de datos” y “Type (instrucción)” o “Tipo de datos definido por el usuario”.

Los tipos definidos por el usuario, también llamados variables *UDT* (User Defined Type) o variables *Structure* (estructuras) son variables especiales que tienen un nombre “general” y luego *n* variables internas o campos. Su interés e importancia son notables. Por ello te recomendamos que más adelante profundices en ellas por tu cuenta. Por el momento solo es necesario que atiendas a los contenidos del curso.

Conociendo ya el significado de *Option Explicit*, los tipos de variables y cómo nombrarlas, veamos ahora cómo declararlas. Hay distintas formas y sitios para declarar variables con Visual Basic. Veremos ahora una forma básica que nos permita empezar a trabajar y, de paso, crear nuestro primer programa.

Usaremos para ello la instrucción *Dim*. Su sintaxis será:

```
Dim [Nombre de variable] As [Tipo de variable]
```

El lugar donde realizaremos la declaración de variables será la cabecera del código asociado al formulario. Así pues, crea un nuevo proyecto (Menú archivo --> Nuevo Proyecto --> Exe estándar ó Menú Archivo --> Nuevo proyecto --> Aplicación de Windows Forms). Vete a la ventana de código (Ver --> Código) donde quizás te aparezca una línea que ponga *Option Explicit* ó *Public Class Form1* *End Class*. Escribe una nueva línea de código (debajo de *Option Explicit* o debajo de *Public Class Form1*) con este contenido:

```
Dim Numerodeplantas As Integer
```

Has declarado la variable *Numerodeplantas* como tipo entero. También puedes usar la abreviatura correspondiente al tipo *Integer* de esta forma:

```
Dim Numerodeplantas%
```

Usar el símbolo % unido al nombre de la variable declarada equivale a escribir “*As Integer*”, lo cual nos supone ahorrar unas cuantas pulsaciones de teclas.

En algunas versiones se permite declarar una variable para la que no se especifica tipo de forma extendida o abreviada. Si esto se hiciera y fuera permitido, la variable es automáticamente asignada por Visual Basic al tipo Variant. Así si escribimos:

```
Dim Numerodeplantas
```

Hay dos opciones: bien obtenemos un error de compilación (si la versión de Visual Basic que estamos usando no admite esta declaración), bien la variable Numerodeplantas es considerada por Visual Basic como tipo Variant. Todavía no está definido si es de tipo numérico (números) o alfanumérico (texto). En este caso la definición se producirá a posteriori cuando asignemos contenido a la variable. Otra forma de declarar la variable como tipo genérico (Variant u Object, según la versión) es usar la expresión:

```
Dim Numerodeplantas As Variant (ó Dim Numerodeplantas As Object)
```

Esta expresión no tiene abreviatura. Siempre será preferible especificar “As Variant” o “As Object” antes que declarar la variable sin especificación alguna. ¿Por qué? Simplemente porque al especificar As Variant ó As Object dejamos constancia de que nosotros, como programadores, hemos decidido crear esa variable como tipo Variant u Object. Si no se especifica puede dar la impresión de que se nos ha olvidado indicar el tipo de variable.

Las variables Variant u Object son un “caso especial”. Las podríamos calificar de potentes y peligrosas. Potentes porque a diferencia de otro tipo de variables, pueden contener distinto tipo de información como texto, números enteros o decimales, etc. y con pocas limitaciones. Peligrosas porque es fácil incurrir en un mal uso de ellas dando lugar a programas difíciles de leer o depurar y más ineficientes.

Supón que declaras: Dim edad As Variant ó Dim edad As Object, como variable destinada a contener la edad de una persona. Sabemos que la edad de una persona puede oscilar entre cero y 150 años (siendo groseros), y que sus valores no son decimales. Por tanto puede declararse como tipo Integer sin ningún problema. ¿Qué supone el declararla sin especificación de tipo (tipo Variant)?

- a) Vamos a ocupar mayor cantidad de memoria al sistema cuando podríamos haber ocupado una menor cantidad. Es una falta de eficiencia y economía.
- b) A la hora de liberar de errores al programa (depurarlo) no sabremos a ciencia cierta qué tipo de datos contiene la variable edad, ya que puede contener tanto el valor numérico 56 como el valor alfanumérico “cincuenta y seis”. Hacemos al programa más difícil de entender.

A la hora de declarar variables conviene seguir las pautas indicadas en el curso “Bases de la programación nivel I” y, resumidamente:

- Elegir siempre el tipo de variable más sencillo posible. Consideraremos que el grado de sencillez viene definido por la memoria requerida (a menor memoria requerida mayor es la sencillez). Esto redundará en un menor consumo de recursos del sistema y en una mayor rapidez de las operaciones que realiza el ordenador.
- Realizar declaraciones ordenadas y claras.

- Evitar el uso del tipo Variant u Object en la medida de lo posible, es decir, sólo usar estos tipos cuando realmente sean necesarios.

Hasta ahora estamos viendo las bases que nos van a permitir construir programas. A medida que vayamos avanzando, empezaremos a construir pequeños programas para aprender practicando.

Próxima entrega: CU00309A

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