



aprenderaprogramar.com

# Declaración de variables en pseudocódigo. Ambito global o local programación. (CU00205A)

Sección: Cursos

Categoría: Curso Bases de la programación Nivel II

Fecha revisión: 2024

Autor: Mario R. Rancel

Resumen: Entrega nº4 del Curso Bases de la programación Nivel II

24

## DECLARACIÓN DE VARIABLES. VARIABLES GLOBALES Y VARIABLES LOCALES

Un desarrollo sobre la teoría de la programación incluiría, sin duda, un apartado específico relacionado con “estructuras de datos” que incluyera el estudio de vectores, matrices, listas, pilas, colas, etc. de forma detallada. Nosotros no vamos a ir más allá de, simplificada, analizar y poner en práctica conceptos básicos.



Entendemos que el lector no tendrá problema en encontrar manuales donde profundizar. Otra licencia que nos hemos permitido, en contra de lo “ortodoxo”, ha sido darle un tratamiento bastante libre a la aparición y gestión de variables en los algoritmos. El objetivo de ello ha sido meramente didáctico: hemos entendido como favorable para el aprendizaje partir de supuestos más libres y sencillos de lo que será habitual<sup>1</sup>. En un momento dado, como puede ser este, se hace necesario restringir esas libertades iniciales para acercarnos más a la práctica profesional de la programación.

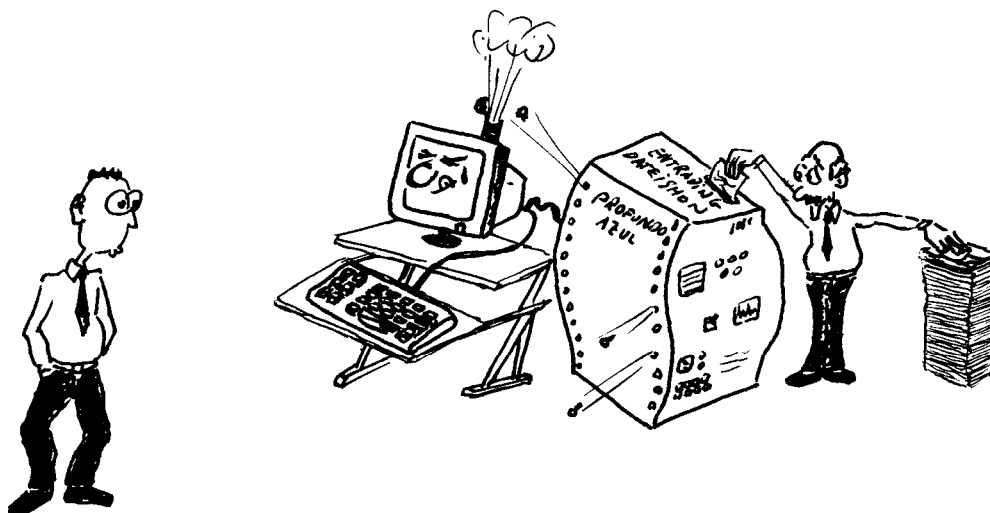
Al definir variable lo habíamos hecho como “un espacio de la memoria del ordenador a la que asignamos un contenido que puede ser un valor numérico (sólo números, con su valor de cálculo) o alfanumérico (sólo texto o texto con números).” El tratamiento de datos por parte del ordenador tiene bases bastante más complejas de lo que puede ser distinguir entre números y texto. Pensemos que las variables se guardan en la memoria y hagamos una analogía entre la memoria como un gran almacén y las variables como cajas en cuyo interior se guarda información. Supongamos que el almacén tiene una cabida de 1000 metros cúbicos ¿Cuántas cajas nos caben? Dependerá del volumen de cada caja. Y este volumen dependerá de la cantidad de información que queramos guardar en cada una de ellas.

Sabemos que ciertos números pueden tener infinitos decimales. Sabemos que los números pueden hacerse tan grandes como queramos. Pero, ¿dispondremos de cajas de infinita capacidad como para guardar en ellas entre nada e infinito? Obviamente no, y la lógica de las variables por ahí se mueve. Guardar o manipular un número entero “normalito” como 1, -32, 115 ó 2005 no nos va a requerir la misma capacidad que guardar o manipular números “pesados” como 1385223612.

Los ordenadores tienen mucha potencia y mucha capacidad, pero no infinita... Una forma de agilizar el trabajo es guardar los números en cajas apropiadas según su tamaño. Es por ello que los diferentes lenguajes trabajan con distintos tipos de datos. Cada tipo de dato tiene un rango de valores admisibles, un número de decimales si se trata de números y una cantidad de memoria ocupada por variable.

---

<sup>1</sup> En iguales circunstancias nos encontramos con el manejo de archivos, campo en el que no nos adentraremos hasta que comencemos con *Visual Basic*. La simplificada extracción de archivos que hemos ido utilizando tiene por objetivo facilitar el aprendizaje.



**Ilustración:** La capacidad de los ordenadores es muy elevada, pero no infinita. No podemos pensar que se puede procesar cualquier número de datos de cualquier maanitud.

En Turbo Pascal, por ejemplo, usar una variable como tipo entera supone requerir 2 bytes de memoria, mientras que una variable decimal de precisión 16 decimales requerirá 8 bytes. No vamos a entrar a analizar los distintos tipos de datos posibles, ya que éstos dependen del lenguaje que utilizemos. Será uno de los primeros aspectos a tener en cuenta cuando comencemos a programar con un lenguaje: los tipos de datos que utiliza. De su conocimiento y buen manejo dependerá en buena medida que seamos capaces de crear programas eficientes. Volveremos sobre ello más adelante.

Sí vamos a introducir aquí el concepto de declaración de variables. La declaración de variables no es otra cosa que trasladar lo que hemos llamado como organización previa de variables al código del programa en sí, de una forma normalizada. La declaración de variables se realizará bien en un espacio inicial previo a todo módulo o algoritmo principal, o bien a continuación de la declaración de módulo, dentro del mismo pero antes de la primera instrucción de ejecución. Una variable declarada en la cabecera del programa se podrá utilizar en cualquier parte del mismo (algoritmo principal o módulo), mientras que una variable declarada dentro de un módulo sólo podrá utilizarse en el módulo en que se ha declarado y en los módulos subordinados a éste si existen. En base a ello hablamos de variables globales y locales: serán globales las accesibles en cualquier parte del programa y locales las accesibles sólo por un módulo y sus subordinados.

```

PROGRAMA EJEMPLO 02

Variables
    Enteras: [nombre de variables]
    Reales: [nombre de variables]
    Booleanas: [nombre de variables]
    Alfanuméricas: [nombre de variables]

[Algoritmo principal]

[Módulos]
    
```

Las variables de un array se declaran agrupadas indicando el nombre del array y entre paréntesis el mayor índice existente. Así Dato(5) comprende a Dato(1), Dato(2), Dato(3), Dato(4) y Dato(5). Todas las variables del array han de ser del mismo tipo. Ampliaremos conceptos relacionados con arrays más adelante.

Por variables enteras entenderemos aquellas que pueden contener cualquier valor entero, positivo, negativo o cero. No admiten por tanto números con decimales.

Las variables reales podrán contener tanto números enteros como decimales.

Las variables booleanas únicamente admiten como contenido verdadero o falso.

Las variables alfanuméricas contienen texto o texto mezclado con números, pero en todo caso el tratamiento es como texto (no existe valor de cálculo).

Si no existe ninguna variable de un tipo la línea correspondiente puede ser eliminada. Una variable declarada de un tipo no puede cambiar de tipo ni contener valores de otro tipo.

Un ejemplo con nombre de variables concretas puede ser:

**PROGRAMA EJEMPLO 03**

**Variables**

Enteras: A, B, Dato(3,2), i, j  
Reales: Max, Min, Media, Longitud(10)  
Booleanas: Encontrado  
Alfanuméricas: Mensaje1, Mensaje2

**[Algoritmo principal]**

**[Módulos]**

La respuesta esperable en determinadas situaciones como uso de una variable no declarada o asignación de un dato no admisible a una variable dependerá del lenguaje que utilicemos, pero nosotros, a efectos de diseño del algoritmo lo catalogaremos simplemente como error.

Una variable declarada la consideraremos inicializada con los valores:

- Cero para variables enteras y reales.
- Falso para variables booleanas.
- Vacío para variables alfanuméricas.

Para el ejemplo anterior, analicemos el resultado que se obtendría con una serie de expresiones en el algoritmo principal.

Expresión	Valoración	Observaciones
A = B	Posible	
C = 18	Error	Variable C no declarada
A = Max	Error	Trata de asignar un contenido real a una variable entera *
Dato(2,2) = 5	Posible	
Dato(4,2) = 5	Error	La variable Dato(4,2) no se encuentra declarada. El máximo valor del primer localizador es 3
Min = 6,75	Posible	
Longitud(4) = 8	Posible	
Encontrado = Verdadero	Posible	
Mensaje1 = "Hola"	Posible	
Max = A	Posible	*
Expresión	Valoración	Observaciones
Media = B / A	Posible	
A = 3 * Max	Error	*
A = 6,75	Error	

\* Nótese que se acepta transferir valores desde una variable entera a una real, por englobar los reales a los números enteros, pero no al revés.

### Próxima entrega: CU00206A

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=36&Itemid=60](http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=36&Itemid=60)