UNIVERSIDAD INSURGENTES

ELEMENTOS DE LENGUAJES DE PROGRAMACIÓN DE ALTO NIVEL

Programación

Bloque 2

PRA0420

El presente material recopila una serie de definiciones, explicaciones, ejemplos y ejercicios prácticos de autores especializados que te ayudarán a comprender los temas principales de este bloque.

Las marcas empleadas en la antología son única y exclusivamente de carácter educativo y de investigación, sin fines lucrativos ni comerciales.

3. Elementos de lenguajes de programación de alto nivel

Un lenguaje de programación es una herramienta que permite desarrollar un software. Se emplea para controlar todo dispositivo físico y lógico, ya sea de un ordenador, un celular, una tableta o una *Smart TV*. Todo esto lo logra, mediante la implementación de algoritmos que nos ayudan a dar instrucciones a una computadora, para resolver los problemas tecnológicos.

Un lenguaje de programación está compuesto por dos elementos relevantes: símbolos y sintaxis; mismos que definen la naturaleza del lenguaje y proporcionan significado a cada una de sus expresiones. Se pueden identificar dos tipos de lenguaje de programación: lenguaje de *bajo nivel* y lenguaje de *alto nivel*.

Figura 1. Lenguaje de bajo nivel: sistema binario

```
01110010
10001101
10101101
11101011
00010100
10010101
```

La característica principal de los lenguajes de programación de *alto nivel* es su estructura semántica, ya que es muy parecida a la forma en la que escribimos normalmente; lo que permite definir y escribir algoritmos de una manera más natural, es decir, sin necesidad de codificar en lenguaje binario (1 y 0), o en el nivel de lenguaje ensamblador.

Figura 2. Lenguaje de alto nivel: algoritmo, hombre-máquina

Actualmente, existen diversos lenguajes de programación como *C++, C, Pascal, Java, Visual Basic,* entre otros.

3.1. Tipos de datos

En informática un dato es la expresión general que describe los objetos con los que opera el algoritmo. Cada dato es un valor que puede ser representado por un número, letra o símbolo, con valores enteros, lógicos o boléanos, cadenas de texto, entre otros; por lo que su contenido puede ser prácticamente cualquiera.

Ahora bien, los tipos de datos son un conjunto de operaciones que se pueden realizar, esto incluye qué valores pueden tomar y qué operaciones pueden realizar. De esta forma, el tipo de dato determina la forma de almacenamiento en memoria y las operaciones que van a poder ser efectuadas con él (Joyanes, Fernández y Rodríguez, 2003, p.15).

Es importante mencionar que todos los lenguajes de programación utilizan los tipos de datos durante su desarrollo del código, pero cada uno con su estructura en particular. Sin embargo, señalan Joyanes, Fernández y Rodríguez (2003), que existen tipos de datos predefinidos, los cuales son:

- **Numéricos.** Se subdividen en enteros (subconjunto finito de los números enteros, cuyo rango o tamaño dependerá del lenguaje en el que posteriormente se codifique el algoritmo), y reales (subconjunto de los números reales limitado no sólo en cuanto al tamaño, sino también en cuanto a la precisión). Este tipo de datos es el más fácil de entender, debido a que las personas están familiarizadas con los números. Pero se pueden presentar situaciones de desbordamiento (*underflow* y *overflow*) positivo y negativo, al realizar operaciones aritméticas.
- **Lógico o booleano.** Conjunto formado por los valores verdadero (*true*) y falso (*false*).
- **Carácter.** Conjunto finito y ordenado de los caracteres que la computadora reconoce.
- **Cadena**. Los datos (objetos) de este tipo contendrán una serie finita de caracteres, que podrán ser directamente traídos o enviados a la consola o desde ésta. (pp. 90-93).

En los algoritmos, para indicar que un dato es numérico, lógico, de carácter, de cadena, etc., se declara utilizando directamente el identificador o nombre del tipo. Revisemos los siguientes ejemplos de tipos de datos:

Cuadro 1. Tipos de datos

Tipo de dato	Dato o valor	Ejemplo: (identificador + dato)
Numéricos o interger	Números reales o enteros (1,2,3,4, etc.).	***Int*** *5*
Caracteres o char	Números del 0 al 9 y letras de la a – z.	***Char*** *x*
Booleanos	Verdadero o falso.	***Booleans*** *True*
Bits	Código binario 0 y 1 (10 dígitos).	***Bits*** *5*
Byte	8 bits (128-127).	***Byte*** *130*
Short	16 bits (32768 a 32767).	***Short*** *5900*
Long	64 bits.	***Long*** (números muy largos)
Campos	Formularios	Nombre: María
Registros	Conjunto de campos.	Nombre: María Luisa Sánchez
Archivos	Conjunto ordenado de registros.	Nombre: María Luisa Sánchez Nombre: Francisco Cruz Pérez Nombre: Hilda Montaño Díaz

Float	Valor punto flotante 8 decimales después del punto.	***Float*** = 1.23456789
Double	Valor punto flotante 15 decimales después del punto.	***Double***= 1.234567890123456
String	Cadena de texto.	Hola buenos días

A partir de lo anterior, revisemos una situación de uso común en la cual identificaremos los datos y los tipos de datos que se utilizaron. Tenemos el siguiente formulario:

Cuadro 2. Ejemplo de tipo de datos

BASE DE DATOS		BASE DE DATOS	
Nombre:	Victoria Ruiz Pérez	Nombre	*string*
Edad:	25	Edad	entero
Sexo:	M	Sexo	carácter
Teléfono:	55 12 12 34 65	Teléfono	entero

Así entonces, identificamos tres tipos de datos:

- Nombre, corresponde a un tipo de dato string o cadena de texto.
- Edad y teléfono corresponden al tipo de dato numérico entero.
- Sexo, es un tipo de dato carácter.

Podemos concluir este primer subtema mencionando que los códigos de carácter más utilizados son EBCDIC (utilizado por las primeras máquinas de IBM) y ASCII (el código universal más extendido). En tanto, los string se utilizan para las cadenas de texto y pueden ser de tamaño fijo o variable.

Cuadro 3. Tipos de tamaño o string

Fijo	Buenos días
Variable	Introduce tu nombre:_____

Los datos pueden venir expresados como constantes, variables, expresiones o funciones.

3.2. Variables

Una variable es un objeto cuyo valor puede cambiar durante el desarrollo del algoritmo. Se identifica por su nombre y por su tipo, que podrá ser cualquiera, y es el que determina el conjunto de valores que podrá tomar la variable. Por ejemplo:

Interger A, O: significa dos variables de nombre A y O y de tipo entero.

Es importante que el nombre de una variable sea único y no ambiguo. Se recomienda no usar acrónimos y que los identificadores no coincidan con las palabras reservadas del lenguaje. Revisemos el siguiente ejemplo:

Tenemos los datos de la siguiente suma:

$$50 + 30 = 80$$

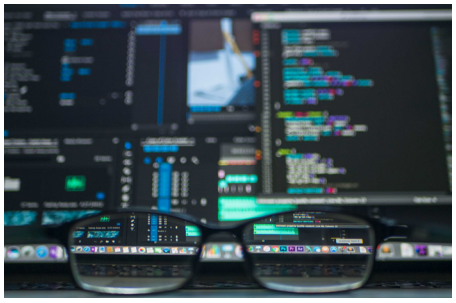
De lo anterior, nombramos las siguientes variables:

Asignamos la letra a para el valor de 50, y se expresa: a= 50.

Asignamos la letra b para el valor de 30, y se expresa: b= 30.

Asignamos la letra c para el resultado de 50, y se expresa: c= resultado.

En total, se ocupan tres variables que son a, b y c, con un tipo de dato numérico entero.

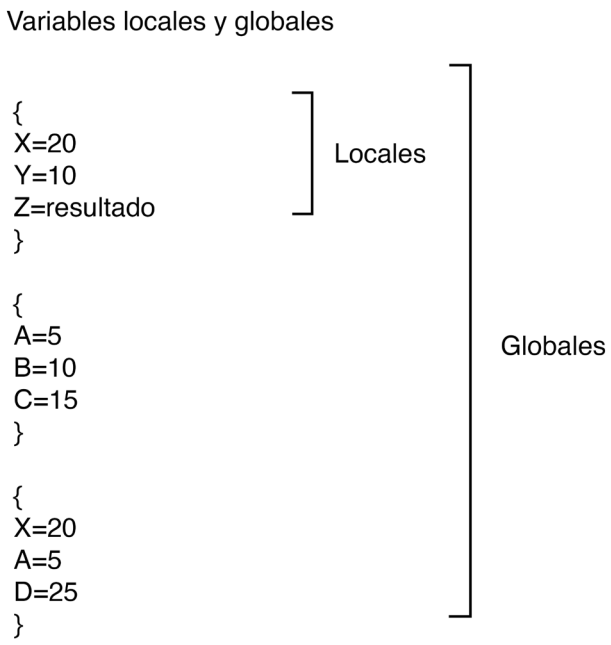


3.3. Variables locales y globales

Las variables locales son aquellas cuyo funcionamiento se restringe al método donde ha sido declarada y no podrá ser utilizada fuera de esa función, es decir, es accesible únicamente en una parte del programa. Por lo tanto, cualquier variable que se defina dentro de las llaves del método se interpreta como variable local.

Las variables globales son aquellas cuyo funcionamiento es implementado en cualquier método de la clase y su valor puede modificarse dentro de cada función donde sea implementada. Este tipo de variables se definen en un momento posterior a la declaración del nombre de la clase y son visibles desde cualquier lugar del programa (Juganaru, 2014, p. 13).

Figura 3. Ejemplo de variables locales y globales



3.4. Constantes

Son datos que mantienen un valor único no modificable a lo largo de toda la ejecución del programa. Por ejemplo, el valor de π= 3.14159265389…

De acuerdo con Cairo (2006), existen dos formas básicas para definir las constantes:

Const int nu1= 20; donde nu1 es una constante de tipo entero.

Otra alternativa es:

#define nu1 20; nu1 es una constante de tipo entero (p. 10).

3.5. Directivas

Las directivas son una guía para el ensamblador, se basan en palabras claves que se van interpretando cuando se lee el código del programa que realizamos. Algunos de los procesos son:

1. Iniciar el código.
2. Acomodar los segmentos del programa en orden alfabético.
3. Identificar el nombre de la variable y el tipo de datos.
4. Trabajar sobre el espacio en memoria.

Todo este proceso se realiza de forma interna al momento de ejecutar el código, por lo tanto, no es posible distinguirlo a simple vista. Por eso, hay que especificar bien las variables que usamos, así como el tipo de datos, para que todo lleve una lógica.

3.6. Bibliotecas de funciones

Los lenguajes de programación están estructurados en una biblioteca o librerías, como se conocen comúnmente, que son partes de programas ya desarrollados, que nos sirven para mostrar gráficos en pantalla, leer datos de entrada desde el teclado, realizar operaciones matemáticas, etc. Cada lenguaje cuenta con su librería en específico, para usarla necesitamos importarla a nuestro programa. Por ejemplo, en el lenguaje de *Java* se encuentra la librería *java.awt,* la cual se utiliza para el desarrollo de interfaz gráfica; en tanto en el lenguaje C++, se ubica la librería math, que se utiliza para la resolución de funciones matemáticas.

REFERENCIAS

Cairo, O. (2006). *Fundamentos de Programación Piensa en C.* [versión electrónica]. Recuperado de https://bit.ly/3adL7Cb

Joyanes, L., Rodríguez, L. y Fernández, M. (2003). *Fundamentos de programación: libro de problemas. Algoritmos, estructuras de datos y objetos.* España: McGraw-Hill.

Juganaru, M. (2014). *Introducción a la programación* [versión electrónica]. Recuperado de https://bit.ly/2X4ex2W