

## Contenido

2.1. Tipos de Datos

2.2. Identificadores

2.3. Variables y constantes

2.4. Declaración de variables

2.5. Expresiones y operadores aritméticos

2.6. Asignación.

2.7. Entrada y Salida

A continuación se describen los elementos básicos de programación que se encuentran en casi todos los lenguajes, como los son: variables, constantes, expresiones, sentencias de entrada y salida.

---

## 2.1 Tipos de Datos

En términos generales, un dato es la representación simbólica de un atributo de una entidad; en programación, los datos expresan características de las entidades sobre las que opera un algoritmo. Los datos representan hechos, observaciones, cantidades o sucesos y pueden tomar la forma de números, letras o caracteres especiales. Por ejemplo, la edad y el domicilio (datos) de una persona (entidad). Son también elementos de información que pueden ser procesados por el computador. Se consideran la materia prima de la información.

Un tipo de dato define el conjunto de valores que es posible crear y manipular, y las operaciones que con ellos pueden llevarse a cabo. Cada variable, constante o expresión tiene asociado un tipo de dato que determina el conjunto de valores que puede tomar. La figura 2.1 muestra claramente los tipos de datos principales clasificados en dos grandes grupos: simples o primitivos y compuestos o estructurados.

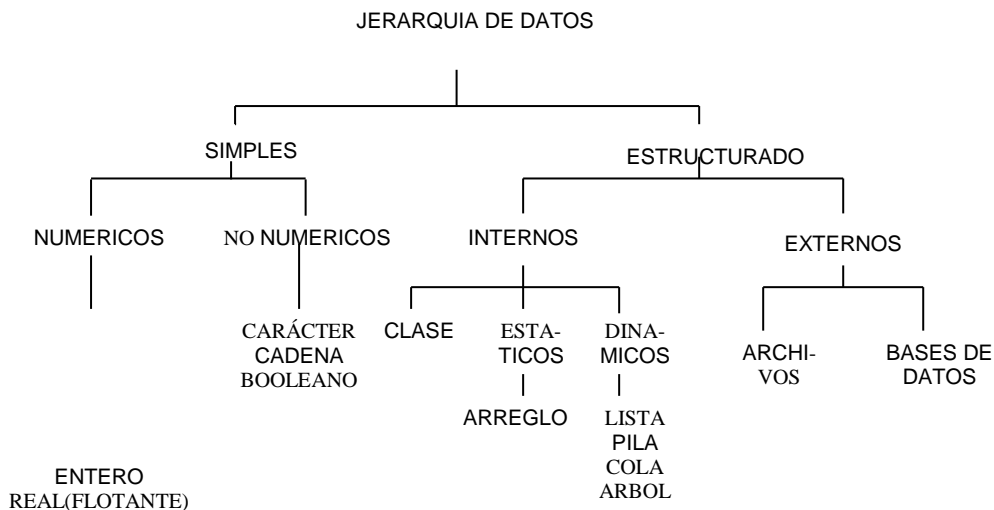


Figura 2.1. Clasificación de los tipos de datos principales

1. *Entero*: Es un valor numérico sin punto decimal. Está formado por un conjunto de dígitos del 0 al 9, que pueden ir precedido del signo + ó -, lo cual le permite representar tanto valores negativos como positivos.

Ejemplos:        2003  
                      -54  
                      +3

2. *Real (flotante)*: Es un valor numérico que tiene punto decimal. Está formado por un conjunto de dígitos del 0 al 9, que pueden ir precedidos del signo + ó -, lo cual le permite representar tanto valores negativos como positivos. La cantidad de valores que se puede almacenar dependerá del microprocesador que se utilice. Se expresa de dos maneras distintas denominadas: notación de punto fijo y notación de punto flotante (exponencial).

a) *Notación de Punto fijo*: es aquella que sólo maneja números del 0 al 9 y el punto decimal.

Ejemplos:        89.57  
                      +235.86  
                      -0.009

b) *Notación de punto flotante (Exponencial)*: Un valor en punto flotante tiene la forma "mantisa exponente E", donde la mantisa es un número real y el exponente un número entero el cual representa la cantidad mantisa multiplicada por 10 elevado al exponente.

Ejemplos:

0.567E2 se puede representar como  $0.567 \times 10^2 = 56.7$

-12.000E-4 se puede representar como  $-12.000 \times 10^{-4} = -.0012$

3. *Character*: Este tipo de dato acepta un solo caracter, alfanumérico (un dígito o una letra mayúscula o minúscula), símbolo o caracter especial, que se encuentre en el conjunto ASCII o UNICODE. El valor debe ir entre apóstrofes.

Ejemplos:

caracteres alfabéticos mayúsculas: 'A', 'B', ... , 'Z'  
caracteres alfabéticos minúsculas: 'a', 'b', ... , 'z'  
caracteres numéricos o dígitos: '1', '2', ... , '9'  
caracteres especiales: ' ', '\*', ... , '/'

4. *Cadena*: Este tipo de dato aloja un valor formado por una sucesión de caracteres (letras, dígitos y caracteres especiales). Se expresan utilizando las comillas dobles.

Ejemplos: “ Universidad Tecnológica”  
 “\$123.67”  
 “17/03/2003”

5. *Booleano o lógico*: Este tipo de dato puede alojar uno de dos valores que representan falso o verdadero.

Problemas propuestos:

	Valores	Tipo de Datos	Correcto S/N	Si está incorrecto ¿POR QUE ?
1.	5280	Entero	S	
2.	2.0e-8			
3.	1000,0			
4.	'A'			
5.	'verde'			
6.	"8:15 p.m."			
7.	"R"			
8.	\$77.7			
9.	+12.34			

## 2.2 Identificadores

Los identificadores son palabras creadas por el programador para denominar los elementos que necesita declarar en un programa, tales como: variables, clases, métodos, objetos y estructuras de datos.

Los identificadores deben crearse de acuerdo con las siguientes reglas:

1. Deben comenzar con una letra.
2. Puede combinar letras, números y el carácter especial subrayado (\_).
3. Los identificadores de clases comienzan con mayúscula, para cualquier otro elemento deben iniciar con minúscula.

Ejemplos:

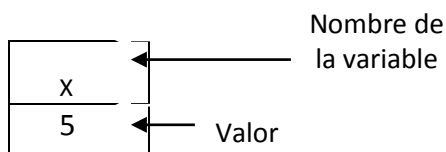
x1, Estado, edad, nombre, num\_seguro, calcular\_Suma ,  
Cpersona

## 2.3 Variables y Constantes

**2.3.1. Variables:** Es un área de almacenamiento temporal con un nombre simbólico cuyo valor puede ser modificado a lo largo de la ejecución de un programa.

Ejemplo:

X = 5



**2.3.2. Constante:** Valor definido que no cambia durante la ejecución de un programa.

Ejemplos: 3.1416

"Universidad"

-256 3

'a'

'\*'

## 2.4 Declaración de Variables

Todos las variables deben ser declaradas antes de ser utilizadas. Esto significa que el compilador debe conocer, *por adelantado*, la variable y el tipo de dato que se estará utilizando. En términos generales, el formato que se emplea para declarar una variable es

Formato:

Tipo\_de\_dato var1, var2,..., varn

Donde:

- *var1..varn* : Es un nombre de identificador.

- *Tipo\_de\_dato*: Permite conocer qué valores se pueden almacenar en ella.

En resumen existen cinco tipos básicos de datos, como mencionamos anteriormente:

entero (para valores numéricos sin punto decimal).  
flotante (para valores numéricos con punto decimal).  
caracter (para un solo símbolo ó carácter).  
cadena (para una sucesión de símbolos o caracteres).  
booleano(para uno de dos valores: falso o verdadero).

Ejemplos:

entero a cont  
flotante salario, monto  
caracter sexo  
cadena nombre, cedula  
booleano error

## 2.5 Expresiones y Operadores Aritméticos

Una *expresión aritmética* es un conjunto de valores, constantes, variables y/o funciones combinadas con operadores aritméticos. Estas expresiones producen resultados de tipo numérico y se construyen mediante operadores aritméticos y funciones matemáticas intrínsecas.

Un *operador* es un símbolo que le indica al compilador que desarrolle una operación matemática o lógica específica.

La siguiente tabla lista los operadores aritméticos:

Operadores	Descripción	Ejemplo
+	Suma	$A + B$
-	Resta	$A - B$
*	Multiplicación	$A * B$
/	División	$A / B$
**	Exponenciación	$A ** B$

Los operadores aritméticos deben aplicarse siguiendo cierta jerarquía:

---

- 1) ( ) paréntesis (se utilizan para alterar el orden jerárquico de evaluación de los operadores).
- 2) \*\* exponenciación
- 3) \*, / multiplicación , división
- 4) +, - suma, resta

Las operaciones con igual jerarquía, se resuelven de izquierda a derecha.

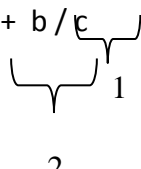
Los siguientes ejemplos representan la conversión de una expresión matemática a expresión algorítmica y el orden de evaluación representado gráficamente.

Ejemplo:

La expresión matemática:  $a + \frac{b}{c}$

en algoritmo se expresa como  $a + b / c$

El orden de evaluación por el computador para la expresión aritmética es:

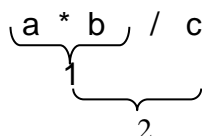


Ejemplo:

La expresión matemática:  $\frac{a * b}{c}$

en algoritmo se expresa como  $a * b / c$  aritméticas.

El orden de evaluación por el computador para la expresión aritmética es:



a)  $(A+B)$

b)  $A+B.C$

c)  $Y^3 + 4$

d)  $A^{x+3} - F$

e)  $\frac{a-b}{c-d}$

f)  $(x^2 - y^2)^3$

## 2.6 Asignación

Es la operación de asignar o cambiar un valor a una variable. Se lleva a cabo de acuerdo al siguiente formato:

Formato: $\text{variable} = \text{expresión}$
--

Donde:

- *variable*: es un nombre de identificador.
  - *expresión*: puede ser una constante, variable o una expresión aritmética.
-



La asignación se lleva a cabo evaluando la expresión aritmética y asignando el resultado obtenido a la variable, definida a la izquierda del símbolo igual (=) de asignación. Por ejemplo:

Ejemplos :

$x=5.7$  /\*a x se le asigna la constante 5.7\*/

$y = m + 3.25 - 11.0 / b$  /\*se evalúa la expresión y se le asigna el resultado a y \*/

## 2.7 Entrada/ Salida.

Para que el computador pueda realizar cálculos requiere de los datos necesarios para ejecutar las operaciones; es decir, necesita una entrada que posteriormente se convertirá en resultados o salida. La obtención de datos de entrada al computador se conoce como *Lectura* y la transferencia de datos hacia un dispositivo interno o externo de un sistema de computadora se denomina *escritura*.

### 2.7.1. Entrada.

Las operaciones de entrada permiten leer valores y asignarlos a determinadas variables. Los datos se introducen al computador mediante dispositivos de entrada (teclado, disco, lector óptico, etcétera) y son almacenados en las direcciones de memoria asignadas a las variables.

En términos generales, el formato es el siguiente:

Formato:  
Leer (var1, var2, var3,...,varn)

Donde :

- *var1 ... varn*: deben ser un nombre de identificador.

La figura 2.2 ilustra la función de un segmento de código de entrada como el siguiente:

```
Ejemplos:  /* Declaración de variables */
            cadena nombre
            flotante salario
            /* Proceso de Lectura de datos */
            Leer ( nombre, salario)
```

Datos de entrada:      *nombre*    *salario*  
                          Pedro    550.50

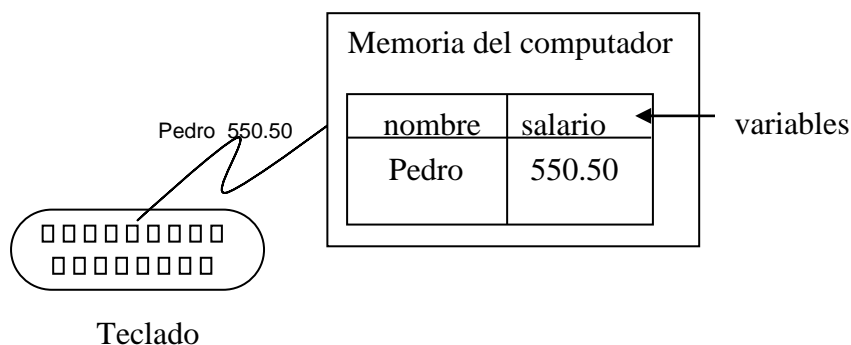


Figura 2.2 Diagrama de Lectura

### 2.7.2. Salida

Las operaciones de salida, permiten desplegar o imprimir los resultados o valores guardados en memoria en un dispositivo de salida (pantalla, impresora, disco, etcétera). Se llevan a cabo bajo los siguientes formatos:

Formatos:

```
Imprimir (lista de variables y/o constantes)
Imprimir ("cadena Constante")
Imprimir (var1,..., varn)
Imprimir ( )
Imprimir ("cadena constante", var)
```

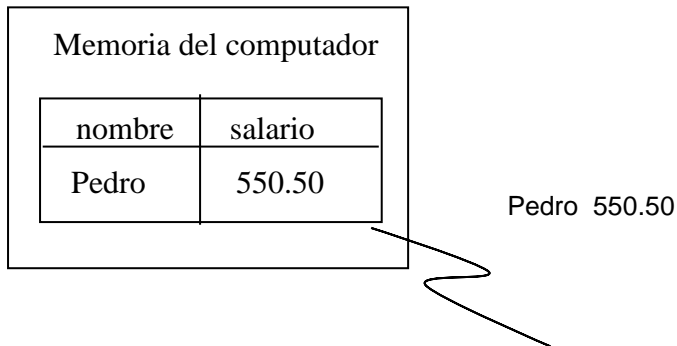
Donde:

- *var1...varn* Es un nombre de identificador.
- *lista variables y/o constantes*: van separadas por comas.

- *cadena constante*: Es una sucesión de letras, dígitos o caracteres especiales entre comillas dobles.

La figura 2.3 ilustra la función de un segmento de código de salida como el siguiente:

Imprimir("Nombre del Empleado= ", nombre, "Sueldo =", salario)



El resultado en pantalla sería:

Nombre del Empleado = Pedro Sueldo =550.50

Figura 2.3 Diagrama de Impresión

Problemas propuestos:

1. Escriba una instrucción que lea el nombre, cedula y edad del estudiante.
2. Escriba una instrucción que imprima el nombre del estudiante.
4. Escriba una instrucción que imprima un mensaje que diga: UNIVERSIDAD TECNOLÓGICA.

5. Escriba las instrucciones necesarias para hacer lo siguiente: lea el nombre y dos notas del estudiante, calcular la nota promedio y después imprima el nombre y la nota promedio.
6. Escriba las instrucciones necesarias para obtener la siguiente salida: que imprima los títulos

Compañía Buenas Ventas		
Nombre	Departamento	Salario

## **Bibliografía**

**Extracto del Capítulo II del libro**  
**“Introducción a la Programación Orientada a Objetos.”**  
**Primera Edición. Imprenta Universitaria. U.T.P.**  
**O. Barraza, F. Krol, L. Meléndez y M. de Velásquez.**