

## Capítulo 2

# *Conceptos de Programación*

### **Objetivo:**

El Alumno conocerá e identificará los conceptos básicos de programación y los aplicará en la creación de algoritmos.

### **2.1 Lenguaje**

Es una serie de símbolos que sirven para transmitir uno o más mensajes (ideas) entre dos entidades diferentes. A la transmisión de mensajes se le conoce comúnmente como comunicación.

La comunicación es un proceso complejo que requiere una serie de reglas simples, pero indispensables para poderse llevar a cabo. Las dos principales son las siguientes:

1. Los mensajes deben correr en un sentido a la vez.
2. Debe forzosamente existir 4 elementos: Emisor, Receptor, Medio de Comunicación y Mensaje.

Tipos de lenguaje:

- **El lenguaje humano** se basa en la capacidad de los seres humanos para comunicarse por medio de signos. Principalmente lo hacemos utilizando el signo lingüístico. Aun así, hay diversos tipos de lenguaje. El lenguaje humano puede estudiarse en cuanto a su desarrollo desde dos puntos de vista complementarios: la ontogenia, que remite al proceso de adquisición del lenguaje por el ser humano, y la filogenia.
- **El lenguaje animal** se basa en el uso de señales sonoras, visuales y olfativas a modo de signos para referirse a un referente o un significado diferente de dichas señales. Dentro del lenguaje animal están los gritos de alarma, el lenguaje de las abejas, etc.

- **Los lenguajes formales** son construcciones artificiales humanas, que se usan en matemática y otras disciplinas formales, incluyendo lenguajes de programación. Estas construcciones tienen estructuras internas que comparten con el lenguaje humano natural, por lo que pueden ser en parte analizados con los mismos conceptos que éste.

## 2.2 Lenguaje de Programación

Es un conjunto de símbolos, caracteres y reglas (programas) que les permiten a las personas comunicarse con la computadora.

Los lenguajes de programación tienen un conjunto de instrucciones que nos permiten realizar operaciones de entrada/salida, cálculo, manipulación de textos, lógica/comparación y almacenamiento/recuperación. Se clasifican en:

- **Lenguaje Máquina:** Son aquellos cuyas instrucciones son directamente entendibles por la computadora y no necesitan traducción posterior para que la CPU pueda comprender y ejecutar el programa. Las instrucciones en lenguaje máquina se expresan en términos de la unidad de memoria más pequeña el bit (dígito binario 0 o 1).
- **Lenguaje de Bajo Nivel (Ensamblador):** En este lenguaje las instrucciones se escriben en códigos alfabéticos conocidos como mnemotécnicos para las operaciones y direcciones simbólicas.
- **Lenguaje de Alto Nivel:** Los lenguajes de programación de alto nivel (*C#*, *C++*, *Visual Basic*, *Java*, etc.) son aquellos en los que las instrucciones o sentencias a la computadora son escritas con palabras similares a los lenguajes humanos (en general en inglés), lo que facilita la escritura y comprensión del programa.

## 2.3 Programa

Un programa, o también llamado programa informático, programa de computación o programa de computadora, es simplemente un conjunto de instrucciones para una computadora. Las computadoras necesitan de los programas para funcionar, y un programa no hace nada a menos que sus instrucciones sean ejecutadas por el procesador. Un programa se puede referir tanto a un programa ejecutable como a su código fuente, el cual es transformado en un ejecutable cuando es compilado.

Generalmente el código fuente de los programas es escrito por profesionales conocidos como programadores. El código fuente es escrito en un lenguaje de programación que sigue uno de los siguientes dos paradigmas: imperativo o declarativo. El código fuente puede ser convertido en una imagen ejecutable

por un compilador. Cuando se pide que el programa sea ejecutado, el procesador ejecuta el programa instrucción por instrucción, hasta que el programa termina.

De acuerdo a sus funciones, los programas pueden ser clasificados en software de sistema y software de aplicación. Pueden ejecutarse muchos programas de forma simultánea en una misma computadora, a lo cual se le llama multitarea.

## 2.4 Algoritmo

Un algoritmo (del latín, *dixit algorithmus* y éste a su vez del matemático persa *al-Jwarizmi*) es una lista bien definida, ordenada y finita de operaciones que permite hallar la solución a un problema. Dado un estado inicial y una entrada, a través de pasos sucesivos y bien definidos se llega a un estado final, obteniendo una solución. Un algoritmo es una serie de pasos organizados que describe el proceso que se debe seguir, para dar solución a un problema específico.

En la vida cotidiana se emplean algoritmos en multitud de ocasiones para resolver diversos problemas. Algunos ejemplos se encuentran en los instructivos (manuales de usuario), los cuales muestran algoritmos para usar el aparato en cuestión o inclusive en las instrucciones que recibe un trabajador por parte de su patrón. También existen ejemplos de índole matemática, como el algoritmo de la división para calcular el cociente de dos números, el algoritmo de Euclides para calcular el máximo común divisor de dos enteros positivos, o el método de Gauss para resolver un Sistema lineal de ecuaciones.

Existen dos tipos de algoritmos que son:

- **Cualitativos:** Son aquellos en los que se describen los pasos utilizando palabras. Como ejemplo podemos decir que la utilización de un directorio (búsqueda de un teléfono). Para poder buscar un teléfono en un directorio, se debe conocer el algoritmo que se va a utilizar, es decir la forma en que están codificados los nombres de las personas, para así lograr encontrarlos y localizar el número telefónico correspondiente.
- **Cuantitativos:** Son aquellos en los que se utilizan cálculos numéricos para definir los pasos del proceso. Como ejemplo podemos citar los pasos para resolver una ecuación de segundo grado, los pasos lógicos y secuenciales para obtener una nomina, entre otros.

### 2.4.1 Ejemplo de algoritmo

**Objetivo:** Algoritmo para cambiar un foco quemado:

#### **Pasos a Seguir:**

1. Situar escalera bajo el foco quemado.

2. Elegir un foco de reemplazo (de la misma potencia que el anterior).
3. Subir por la escalera hasta alcanzar el foco.
4. Girar el foco contra las manecillas del reloj hasta que esté suelto.
5. Ubicar el foco nuevo en el mismo lugar que el anterior.
6. Enroscar en el sentido de las manecillas del reloj hasta que quede apretado.
7. Bajar de la escalera.

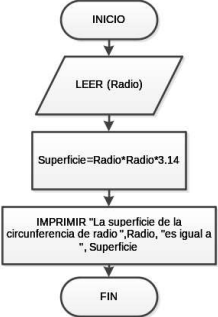
## 2.5 Lenguajes Algorítmicos

Son una serie de símbolos y reglas que se utilizan para describir de manera explícita un proceso.

Los tipos de Lenguajes Algorítmicos son:

**Gráficos:** Es la representación gráfica de las operaciones que realiza un algoritmo (diagrama de flujo).

**No Gráficos:** Representa en forma descriptiva las operaciones que debe realizar un algoritmo (pseudocódigo).

Pseudocódigo	Diagrama de Flujo ( <i>Flowchart</i> )
<p>Su estructura es similar a la de un programa. Es un punto intermedio entre el lenguaje natural y el lenguaje de alto nivel:</p> <pre> Algoritmo SuperficieCircunferencia INICIO   CONST   PI = 3.1416   ENTERO  Radio, Superficie   LEER Radio   Superficie = PI*Radio*Radio   IMPRIMIR "La superficie de la   circunferencia de radio",   Radio,"es igual a ",   Superficie FIN           </pre>	<p>Indica el orden en que deben ejecutarse las acciones. Ofrece mucha claridad y facilidad para diseñar el algoritmo pero a la hora de realizar tareas de actualización dificulta el proceso.</p>  <pre> graph TD   INICIO([INICIO]) --&gt; LEER[/LEER (Radio)/]   LEER --&gt; PROC[Superficie=Radio*Radio*3.14]   PROC --&gt; IMPR[IMPRIMIR "La superficie de la circunferencia de radio",Radio,"es igual a ",Superficie]   IMPR --&gt; FIN([FIN])           </pre>

## **2.6 Práctica y actividad para el alumno**

1. Realizar la elaboración, presentación y defensa, de al menos dos algoritmos utilizados para solucionar problemas, basados en el ejemplo del foco quemado.



## Capítulo 3

# *Entidades primitivas para el desarrollo de algoritmos*

### **Objetivo:**

El alumno conocerá las reglas para desarrollar habilidades que le permitan cambiar fórmulas matemáticas a expresiones válidas para la computadora, además de diferenciar y aplicar correctamente la utilización de constantes, variables, tipos de datos, entre otros elementos que son fundamentales para la construcción de algoritmos.

### **3.1 Tipos de Datos**

Para guardar datos internamente y representar cualquier situación en la computadora, es necesario que los datos tengan un tipo de dato asociado con ellos. Mediante un dato se puede representar un simple carácter, tal como ‘g’, el nombre de una persona, tal como “Geraldynne”, un valor entero, tal como 5, etc.

Los tipos de datos determinan la naturaleza del conjunto de valores que puede tomar una variable y se clasifican en: Diagrama 3.1

#### **3.1.1 Tipos de datos Simples**

La principal característica de los datos simples es que hacen referencia a un único valor a la vez, es decir, una variable almacena un único elemento (valor). Dentro de este grupo se encuentran:

- **Datos Numéricos:** Permiten representar valores escalares de forma numérica, así como también, realizar operaciones aritméticas comunes tales como: multiplicar dividir, sumar y restar. En el ámbito computacional, este tipo incluye a:

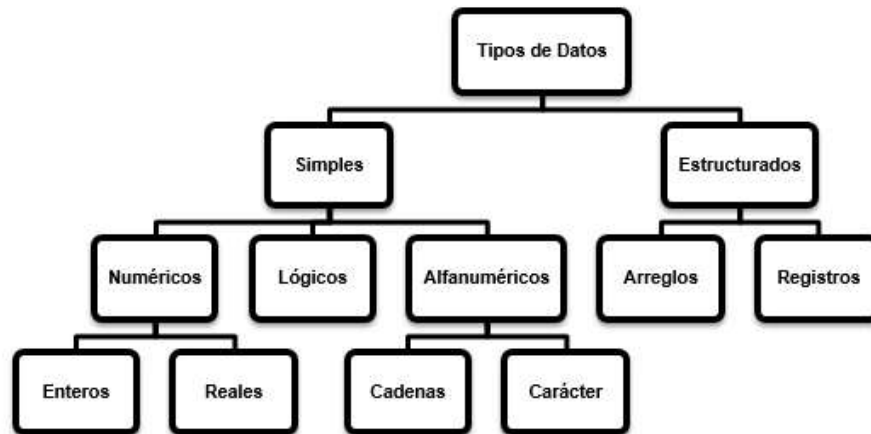


Figura 3.1: Jerarquía de tipo de datos

**ENTEROS** Son los números que no contienen componentes fraccionarios y, por tanto, no incluyen el punto decimal; pueden ser positivos o negativos.

Ejemplo:

- 23, -345, 9, -9.
- Número de autos vendidos en un día.
- Número de pacientes con cita para un día determinado
- Número de días trabajados por un empleado

**REALES** Son los números que contienen una parte fraccionaria y, por tanto, incluyen el punto decimal; pueden ser positivos o negativos. Ejemplo:

- 3.14, -45.7, 6.7, -6.7
- El sueldo de un trabajador
- El precio de venta de un producto
- Cantidad de queso a comprar

- **Datos Alfanuméricos:** Este tipo de dato puede almacenar letras (A-Z, a-z), dígitos (0 - 9) y caracteres especiales (\$, #, \_, entre otros<sup>1</sup>). Este tipo incluye a:

**CADENAS** Es una secuencia de caracteres alfanuméricos, que permiten representar valores descriptivos, esto incluye a:

- Nombres de personas o de empresas

- Dirección de personas o empresas
- Descripción de artículos
- Nombre de artículos
- Nombre de un país
- La marca de una computadora, entre otros.

Este tipo de datos se representan encerrados entre comillas (“”).

Por ejemplo:

- “Facultad de Informática”
- “2008”
- “Culiacán Rosales Sinaloa”
- “Nissan 200sx 1.6lts”

Es posible representar números como alfanuméricos, pero estos pierden su propiedad matemática, es decir, no es posible hacer operaciones aritméticas con ellos.

Por ejemplo:

- Número telefónico de personas o empresas
- El código postal de una dirección
- El número de cuenta de un alumno de la UAS
- El número de cuenta de una tarjeta de crédito

**CARÁCTER** Es un tipo de que permite almacenar sólo un símbolo alfanumérico, es decir, almacena un sólo carácter válido para la computadora según el código ASCII. Normalmente se encierra entre apostrofes ( ‘ ’ ). Almacenan un solo carácter que puede ser una mayúscula o minúscula, un dígito o caracteres especiales (como el signo @, & , !, entre otros<sup>1</sup>).

Por ejemplo:

- ‘a’, ‘B’, ‘1’, ‘#’
  - El género de una persona (‘F’ para Femenino o ‘M’ para Masculino )
  - El estado civil de una persona (‘S’ para Soltero o ‘C’ para Casado )
  - El tipo de gobierno (‘M’ para Municipal, ‘E’ para Estatal o ‘F’ para Federal).
- **Datos Lógicos:** Son aquellos que sólo pueden almacenar uno de dos posibles valores *False* (falso) y *True* (verdadero), ya que representan el resultado de una comparación entre otros datos (numéricos o alfanuméricos).  
Ejemplos:

---

<sup>1</sup>Para ver la lista completa del código ASCII, ir a ANEXO A

Comparación	Resultado (dato lógico)
¿5 es mayor que 4?	True
¿"Rogelio" es igual a "Rogelio"?	True
¿"Informatica" es igual a "Informática"?	False
"Ana" es igual a "Fernanda"?	False

Tabla 3.1: Ejemplos de Datos Lógicos.

Las **palabras reservadas** son aquellas palabras que tienen un significado especial en el pseudocódigo o lenguaje de programación que se utilice. No se pueden utilizar con otro propósito.

Las palabras reservadas, que se utilizarán, en este material, para representar a cada tipo de dato descrito anteriormente son:

Tipo de dato	Palabra reservada para su representación
Numérico Entero	ENTERO
Numérico Real	REAL
Alfanumérico (más de un carácter)	CADENA
Alfanumérico (un sólo carácter)	CHARACTER

Tabla 3.2: Palabras Reservadas

### 3.1.2 Tipos de Datos Estructurados

Los datos estructurados son aquellos que se caracterizan por el hecho de que con un sólo identificador, se hace referencia a un grupo de componentes que pueden ser a su vez un dato simple o estructurado, entre estos se encuentran:

**Arreglos.** Se forma por un conjunto de elementos de un mismo tipo de dato simple. Pueden almacenar a más de un elemento (valor) a la vez, bajo un mismo identificador, que se diferencian por la posición que tienen cada elemento dentro del arreglo de datos.

**Registros.** Es un grupo de elementos de datos agrupados bajo un nombre. Estos elementos de datos, conocidos como miembros, pueden tener diferentes tipos.

### 3.1.3 Variables

Es un espacio en la memoria de la computadora que permite almacenar temporalmente un dato durante la ejecución de un proceso, su contenido puede

cambiar durante la ejecución del programa.

Las variables sirven para representar y manejar datos. Todo dato que vaya a ser introducido a la computadora o que vaya a ser generado, calculado a partir de otros datos para obtener algún resultado debe identificarse y manejarse en forma de variable.

Para poder reconocer una variable en la memoria de la computadora, es necesario darle un nombre con el cual podamos identificarla dentro de un algoritmo.

La sintaxis<sup>2</sup> para su definición es:

```
TipoDato Identificador
ó
TipoDato Identificador = Valor
```

Donde:

- TipoDato            Indica el tipo de dato que tendrá la variable.
- Identificador      Es el nombre de la variable.
- Valor                Es el valor que toma la variable, en caso de inicializarla (opcional).

Ejemplo:

- CADENA NombreAlumno
- ENTERO TotalVentas = 0
- ENTERO TotalVentas

Las variables se clasifican en:



Figura 3.2: Clasificación de Variables

La clasificación de las **variables por su contenido** ya se ha explicado en la sección “Tipo de Datos” al inicio de este capítulo.

<sup>2</sup>Sintaxis es la forma correcta (orden) de escribir una instrucción en un pseudocódigo o lenguaje de programación.

**Las variables por su uso son las siguientes:**

**Variables de Trabajo.** Son aquellas variables que reciben el resultado de una operación matemática completa y que se usan normalmente dentro de un programa.

Ejemplo:

- `Suma = Calificacion1 + Calificacion2 + Calificacion3`
- `Promedio = Suma / 3`

**Contadores.** Se utilizan para llevar el control del número de ocasiones en que se realiza una operación o se cumple una condición. Utiliza incrementos generalmente de uno en uno.

Ejemplo:

- `TotalAlumnosOjosAzules=0`, y su cumplimiento en cada ocasión
- `TotalAlumnosOjosAzules = TotalAlumnosOjosAzules + 1`

**Acumuladores.** Forma que toma una variable y que sirve para llevar la suma acumulativa de una serie de valores que se van leyendo o calculando progresivamente.

Ejemplo:

- `TotalVenta=0`, y su cumplimiento en cada ocasión
- `TotalVenta = TotalVenta + PrecioProducto`

### 3.1.4 Constantes

Una constante es un dato de tipo numérico, lógico o alfanumérico, almacenado en la memoria de la computadora, que no cambia durante la ejecución del programa, es decir, son valores fijos, en consecuencia invariables.

La sintaxis utilizada para su definición es:

```
CONST TipoDato Identificador = Valor
```

Donde:

<code>CONST</code>	Identifica la acción que se va a realizar, la cual es definición de constantes simbólicas.
<code>TipoDato</code>	Indica el tipo de Dato que tendrá la constante.
<code>Identificador</code>	Es el nombre de la constante.
<code>Valor</code>	Es el valor que toma la constante (obligatorio).

Ejemplo:

- `CONST REAL PI = 3.1416`
- `CONST NombreFacultad = "Facultad de Informática"`
- `CONST BOOLEAN Resultado=True`
- `CONST ENTERO Porcentaje = 30`

**NOTA:** Una definición incorrecta sería omitir el valor que se debe asignar a la constante, por ejemplo: `CONST ENTERO Porcentaje`

## 3.2 Identificadores

Un identificador es una secuencia de caracteres que sirve para identificar una posición en la memoria de la computadora, que nos permite acceder a su contenido. Además, los identificadores representan los datos de un programa (constantes, variables, tipos de datos, entre otros) que van a ser calculados o generados a partir de otros datos para obtener algún resultado.

Ejemplo:

- NombreAlumno
- NumeroHoras
- Calificacion

Las reglas y recomendaciones para nombrar identificadores son:

Reglas	Recomendaciones
Debe comenzar con una letra (A a Z, mayúsculas o minúsculas).	Evite poner todas las letras en mayúsculas o minúsculas
Los demás caracteres pueden ser letras (a, b, c, ... ,z), el signo de subrayado ( <u>_</u> ) o dígitos (0, 1, 2, ... ,9).	Evite empezar con un signo de subrayado
No deben contener espacios en blanco.	Evite el uso de abreviaturas
	Use PascalCasing para nombres con varias palabras: inicial de cada palabra en mayúsculas sin espacio entre palabras.
	Debe definirse de tal manera que indique lo que está representando.

Ejemplos:

Identificador	Resultado	Cumple con las reglas
NombreAlumno	<b>Correcto</b>	Sí
1Telefono	<b>Incorrecto</b>	No, debido a que inicia con número
SuelTrab	<b>Incorrecto</b>	No, debido a que es una abreviación
Respuesta12	<b>Correcto</b>	Sí
IVA	<b>Correcto</b>	Sí, debido a abreviación estandarizada
Calificación Final	<b>Incorrecto</b>	No, debido a que incluye un espacio en blanco y acento



- Relacionales, si sus operadores son relacionales y el resultado de la evaluación es BOOLEAN (*True* o *False*).
- Mixta, si se combinan los tipos de expresiones anteriores para su construcción.

## 3.4 Operadores y Operandos

Operadores son elementos que relacionan de forma diferente, los valores de una o más variables y/o constantes. Es decir, los operadores nos permiten manipular valores.

Tipos de Operadores:

- Aritméticos
- Lógicos, y
- Relacionales

Operandos son números constantes o variables. El resultado de una operación aritmética será un número.

### 3.4.1 Operadores Aritméticos

Los operadores aritméticos permiten la realización de operaciones matemáticas con los valores (variables y constantes). El resultado de una operación aritmética será un dato tipo numérico.

Los operadores aritméticos pueden ser utilizados con tipos de datos enteros o reales, para formar expresiones aritméticas en la realización de cálculos. Si ambos son enteros, el resultado es entero; si alguno de ellos es real, el resultado es real. Los operadores aritméticos para las operaciones básicas son: También,

Símbolo	Operación
*	Multiplicación
/	División
MOD	Módulo (residuo de la división entera)
+	Suma
-	Resta

se encuentran los operadores aritméticos unarios, cuyo objetivo es afectar a un sólo operando como lo es el cambio de signo, estos símbolos son los utilizados en las operaciones de suma y resta.

Ejemplo:

- $X1 = -A$

- $X2 = +B$        $X1 = (A + -B) * (-C + -D)$

### 3.4.2 Funciones Matemáticas

Cada lenguaje de programación tiene su conjunto de funciones matemáticas predefinidas, las cuales se ejecutan haciendo referencia a su nombre. Algunas necesitan, para arrojar un resultado, que se suministre información adicional (parámetros o argumentos). Las funciones matemáticas más utilizadas son:

Función matemática	Significado
<b>POTENCIA</b>	
POW(Base, Exponente)	Devuelve el resultado de elevar la Base al Exponente indicado.
<b>RAÍZ CUADRADA</b>	
SQRT(Valor o Dato de Entrada)	Devuelve la raíz cuadrada de su entrada

Ejemplo:

- $X1 = POW(10, 2)$
- $X2 = 5 * 3.1416 * POW(Radio, 2)$

### 3.4.3 Operadores Relacionales

Sirven para la elaboración de expresiones lógicas mediante las cuales se pueden plantear condiciones o comparaciones y que dan como resultado un valor booleano (*True, False*).

Los operadores relacionales son:

Símbolo	Operación
>	Mayor que
<	Menor que
>=	Mayor o igual que
<=	Menor o igual que
<>	Diferente
==	Igual (igualdad lógica)

Las expresiones lógicas simples se forman relacionando variables, constantes, subexpresiones aritméticas, mediante algunos de los operadores relacionales.

### 3.4.4 Operadores Lógicos

Estos operadores se utilizan para formar expresiones lógicas complejas. Se establecen relaciones entre valores lógicos y sus operandos pueden ser resultado de una expresión relacional. Los operadores lógicos son:

Operador	Significado
AND	Conjunción
OR	Disyunción
NOT	Negación

La tabla de verdad de los operadores lógicos es:

Expresion1	Expresion2	AND	OR	NOT Expresion1
TRUE	TRUE	TRUE	TRUE	FALSE
TRUE	FALSE	FALSE	TRUE	FALSE
FALSE	TRUE	FALSE	TRUE	TRUE
FALSE	FALSE	FALSE	FALSE	TRUE

## 3.5 Operador de Asignación

Se utiliza para asignar valores o expresiones a una localidad de memoria (variable). Es una operación destructiva debido a que si la variable tenía asignado un valor anteriormente, éste se destruye, conservando ahora el nuevo valor.

Su sintaxis es:

`Variable = expresión o valor`

Donde:

- $X = A + (B - 5)$
- $TotalVenta = 100$
- $TotalVenta = Precio * 2$
- $Aprobado = Calificacion \geq 6) AND (Calificacion \leq 10)$
- $Resultado2 = Resultado$

### 3.5.1 Prioridad de los Operadores en General

La computadora al evaluar una expresión aritmética, lógica, relacional o mixta, primero la analiza completamente, y va evaluando de izquierda a derecha cada componente de acuerdo al orden de precedencia que tienen los operadores, el cual se muestra en la Tabla 3.3

Nivel de prioridad	Operadores
1	( ) asociación
2	Operaciones unarias (cambio de signo)
3	*, /, MOD, NOT
4	+, -, AND
5	>, <, >=, <=, <>, ==, OR

Tabla 3.3: Prioridad de operadores en general.

Por lo anterior, al elaborar una expresión, debemos respetar la jerarquía del orden de aplicación, ya que:

- Todas las expresiones entre paréntesis se evalúan primero.
- Las expresiones con paréntesis anidados se evalúan de dentro a fuera, el paréntesis más interno se evalúa primero.
- Los operadores en una misma expresión con igual nivel de prioridad se evalúan de izquierda a derecha.
- Si al menos uno de los operandos de una expresión es una función matemática, primero se evalúa la función y después el operador.

### 3.6 Actividades Prácticas

Especifique el tipo de dato que ocuparía para almacenar en la computadora los siguientes datos:

- El código Postal.
- El total de frutas de un árbol.
- El género (sexo) de una persona.
- El teléfono de una empresa.
- Estado civil de una persona.
- El pago por hora de un obrero.
- El precio de venta unitario de un producto.
- Si una persona cuenta con tarjeta de crédito del banco bancosur.
- El número de placas de un automóvil.
- El consumo en Kilowatts de una casa habitación.

De la siguiente lista de Expresiones, indique la causa porque son incorrectas, en caso de serlo:

- $X1 = A + B * C -$
- $X2 = (A + B) * C))$
- $X3 = (A+)B * C$
- $X4 = A + -B * C + 5$
- $X5 = A + * - B/2$
- $X6 = \text{Parcial1} + \text{Parcial2} + \text{Parcial3}/3$

Indique el orden en la evaluación y el resultado que obtendría la computadora, de las siguientes expresiones, además indique el tipo de dato del resultado:

1.  $X_1 = 7 + 5 - 6$
2.  $X_2 = 7 - 5 + 6$
3.  $X_3 = 9 + 7 * 8 - 36/5$
4.  $X_4 = 7 * 8 * (160 \text{MOD} 3) / 5 * 13 - 28$
5.  $X_5 = (3 * 4) + (4 - 2) / (9 * (5 - 3))$

Para las siguientes expresiones tome en cuenta los valores de las siguientes variables:

**A= 5 y B=16.**

6.  $X_6 = (A * 2) > (B + 6 * 2)$
7.  $X_7 = A * 5 + 45.78/4 <= (B * 3/A)$
8.  $X_8 = \text{NOT}(15 >= 7 * 2) \text{OR}(43 - 8 * 2/4 <> 3 * 2/2)$

Convertir las siguientes formulas a expresiones válidas por la computadora:

$$X_9 = 3xy^2 - 2x^2y$$

$$X_{10} = \frac{6 + 18}{2}$$

$$X_{11} = \frac{\frac{a}{b} + b}{x - \frac{y}{z}}$$

$$X_{12} = \sqrt{a^2 + b^2}$$

$$X_{13} = \left( \frac{a + b}{c + d} \right)^3$$

$$X_{14} = 4x + 5y/2z^2$$

$$X_{15} = \pi \cdot r^2$$

$$X_{16} = \pi r^2 h$$

$$X_{17} = \frac{x^2}{a^2} - \frac{y^2}{b^2}$$

$$X_{18} = 5x^5 + 7x^4 - 5x^3 + 3x^2 - 8x + 4$$

Nota: puede utilizar las funciones `SQRT()` y `POW()`.