


**NOMBRE DEL ESTUDIANTE: \_\_\_\_\_**
**MATERIAL DE APOYO No 10. ALGORITMOS CONCEPTOS BASICOS**
**OBJETIVOS:**

- Conocer los conceptos de los algoritmos.
- Aplicar las habilidades del pensamiento para la solución de problemas.
- Resolver problemas planteados.

**VARIABLES**

Son espacios de trabajo (contenedores) reservar para guardar datos (valores). El valor de una variable para cambiar en algún momento del algoritmo, por lo tanto el valor que contiene una variable es el último que se le haya asignado.

Los espacios de memoria (Variables o Constates) se identifican con un nombre; el nombre debe ser alusivo al tipo de información que esta va a guardar.

**CONSTANTES**

Son datos que, luego de ser asignados, no cambian en ninguna instrucción del Algoritmo. Pueden contener constantes matemáticas ( $\pi$ ) o generadas para guardar valores fijos (3.8, "Jorge", etc).

**Ejemplo 1:** Elaborar un Algoritmo para calcular el área de cualquier triángulo rectángulo y mostrar su resultado.

**PSEUDOCODIGO**

**Paso 1:** Inicio

**Paso 2:** Asignar el número 2 a la constante "Div"

**Paso 3:** Conocer la base del triángulo y guardarla en la variable "Base"

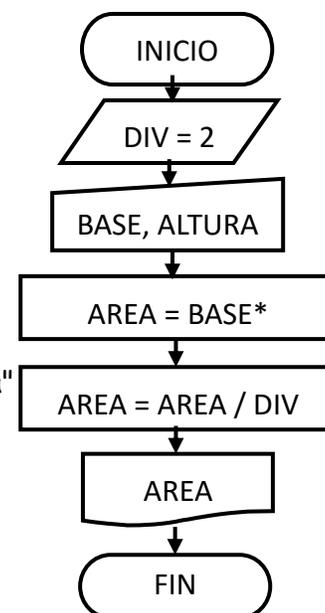
**Paso 4:** Conocer la altura del triángulo y guardarla en la variable "Altura"

**Paso 5:** Guardar en la variable "Area" el valor de multiplicar "Base" por "Altura"

**Paso 6:** Guardar en la variable "Area" el valor de dividir "Area" entre "Div"

**Paso 7:** Reportar el valor de la variable "Area"

**Paso 8:** Final


**DIAGRAMA DE FLUJO**

## ALGORITMO

Inicio

DIV = 2

Escribir "Ingrese el valor de la Base del triángulo"

Leer Base

Escribir "Ingrese el valor de la Altura del triángulo"

Leer Altura

Area = Base \* Altura

Area = Area / Div

Escribir "El área del triángulo es: ", Area

Fin

Hallar el Área de un Triángulo en Scratch.



## CONTADORES

Su estructura es  $(A = A + 1)$  Almacena en una variable (A) el valor de ella misma (A) más un valor constante (1). Es muy útil para controlar el número de veces que debe ejecutarse un grupo de instrucciones.

## ACUMULADORES (A = A + B)

Consiste en almacenar en una variable (A) el valor de ella misma (A) más otro valor variable (B). Es muy útil para calcular sumatorias.

**Ejemplo 2:** Escribir un Algoritmo para calcular la sumatoria de los números entre 1 y 10.

**ALGORITMO**

Inicio

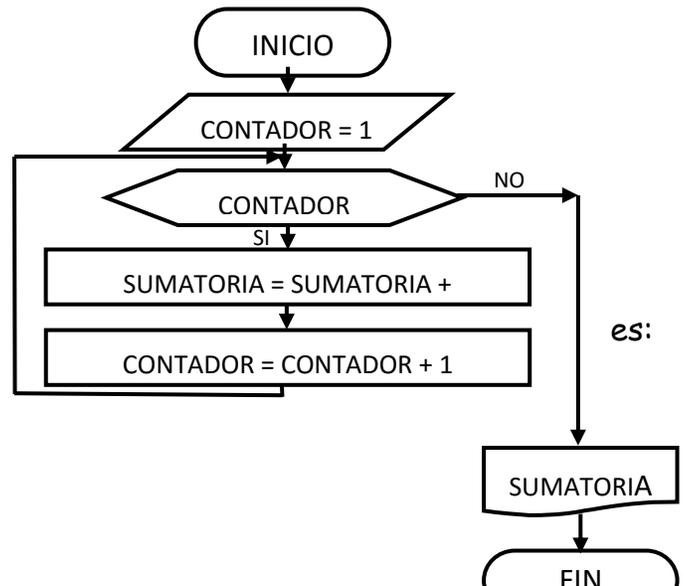
Para Contador = 1 hasta 10 hacer

Sumatoria = Sumatoria + Contador

Fin para

Escribir "La suma de los 10 primeros números", Sumatoria

Fin



```

al presionar
  decir ¿Quieres saber cual es el valor de la suma de los numeros del 1 al 10? por 2 segundos
  fijar Contador a 1
  fijar Sumatoria a 0
  repetir hasta que Contador > 10
    decir unir Contador es: unir Contador unir unir Sumatoria es: Sumatoria por 2 segundos
    fijar Sumatoria a Sumatoria + Contador
    fijar Contador a Contador + 1
  decir unir La suma de los 10 primeros numeros es: Sumatoria
  
```

**TIPO DE DATOS**

Es el tipo de información que se puede trabajar. Es necesario indicar que tipo de información se va utilizar en la solución a un problema y así saber qué operaciones se pueden realizar con ellos.

**DATOS NUMERICOS:** Son números positivos o negativos (1, -20, -10,5, 24.67). Se utilizan cuando se van a realizar operaciones aritméticas (+, -, \*, /, etc).

**PALABRAS:** Están formadas por letras, símbolos y/o números, (Hola, 548, Once-Caldas, ¿Quién?).

## OPERADORES:

- ✓ **Aritméticos**: Posibilitan las operaciones entre datos de tipo numérico y dan como resultado otro valor de tipo numérico. Ejemplo: potencia (^); producto (\*); división (/); suma (+); resta (-); operador de asignación (=)
- ✓ **Alfanuméricos**: Permiten operar con datos de tipo carácter o cadenas. La mayoría de los lenguajes de programación admiten el operador + para realizar la concatenación (unión) de caracteres o cadenas. En Scratch debe utilizarse, para concatenar, el operador &.
- ✓ **Relacionales**: Permiten la comparación entre datos del mismo tipo y dan como resultado dos valores posibles: Verdadero o Falso. Ejemplo: igual a (=); menor que (<); mayor que (>).
- ✓ **Lógicos**: Posibilitan la evaluación lógica de dos expresiones de tipo lógico. Dan como resultado uno de dos valores posibles: Verdadero o Falso. Ejemplo: negación (no); conjunción (y); disyunción (o).

## ORDEN DE EVALUACIÓN DE LOS OPERADORES

Para resolver apropiadamente una expresión, los operadores deben de resolverse según un orden de prioridad (jerarquía), este orden es el siguiente:

1. Paréntesis (se ejecutan primero los más internos)
2. Potencias y Raíces (^ y  $\sqrt{\quad}$ )
3. Productos y Divisiones (\* y /)
4. Sumas y Restas (+ y -)
5. Concatenación (+)
6. Relacionales (=, <, >, >=, <=, !=)
7. Negación (no)
8. Conjunción (y)
9. Disyunción (o)

## EXPRESIONES

Una Expresión está compuesta por valores, funciones, constantes y/o variables, o por una combinación de los anteriores mediante operadores. Son Expresiones:

- ✓ Un valor (1.3, "Jorge")
- ✓ Una Constante o una Variable (divide, base, área)
- ✓ Una combinación de valores, constantes, variables, funciones y operadores que siguen reglas de construcción y orden de evaluación de los operadores (cos 60 + 7 - :altura)

### Las Expresiones pueden ser:

- ✓ **Aritméticas:** Dan como resultado un valor numérico. Contienen únicamente operadores aritméticos y datos numéricos ( $\pi * 20 - :X$ )
- ✓ **Alfanuméricas:** Dan como resultado una serie o cadena de caracteres.
- ✓ **Lógicas:** Dan como resultado un valor "Verdadero" o "Falso". Contienen variables y/o constantes enlazadas con operadores lógicos ( $A > 0$  y  $B \leq 5$ ).
- ✓ **De Asignación:** Estas Expresiones asignan el resultado de una Expresión a una Variable o a una Constante. La Expresión de Asignación (da "área :base \* :altura / 2) asigna (da) el valor resultante de la Expresión Aritmética (:base \* :altura / 2) a la variable área.

Para diseñar algoritmos que posteriormente puedan ser traducidos a un lenguaje de programación, es fundamental saber manejar muy bien los operadores y el orden en el que estos se ejecutan. Las fórmulas deben escribirse en una sola línea para que el computador las evalúe.

### EJEMPLO 3.

Evaluar la expresión muestra  $((5 + 3)^2 - 10) / 3 + 4 * (2 + 4)$  teniendo en cuenta la jerarquía de los operadores:

$$\begin{aligned}
 R/. \quad & ((5 + 3)^2 - 10) / 3 + 4 * (2 + 4) \\
 & = ((5+3) ^ 2 - 10) / 3 + 4 * (2 + 4) \\
 & = ((8 ^ 2) - 10) / 3 + 4 * 6 \\
 & = (64 - 10) / 3 + 4 * 6 \\
 & = 54 / 3 + 4 * 6 \\
 & = 18 + 24 \\
 & = 42
 \end{aligned}$$

### ACTIVIDAD 10.1.

Tomando como modelo el ejemplo anterior y utilizando lápiz y papel, desarrollar paso a paso las siguientes expresiones. Tener en cuenta la jerarquía de los operadores:

1.  $243 + 5 - 6 + 86 - 42 =$
2.  $5 + 86 - 42 - 6 + 243 =$
3.  $7 + (8 * 16) =$
4.  $(7 + 8) * 16 =$
5.  $24 / 4 * 8 =$
6.  $4 * 8 / 24 =$
7.  $(5 + 2) * (4 + 4) = 56$
8.  $7 + 3 * 2 + (2 - 1) = 14$

$$9. 6 * 2 + 4 * 3 + 5 / 2 = 26,5$$

$$10. 5 + 1 * 4 / 2 * 7 - (8 + 2) = 9$$

$$11. 8 + (5 * 6) - 6 = 32$$

$$12. 9 + 5 * 3 = 24$$

$$13. 4 / 2 * (10 - 5) * 3 = 30$$

### ACTIVIDAD 10.2.

Realiza en Scratch al menos 6 de las anteriores expresiones, escribe en el cuaderno los bloques de instrucciones que utilizo en Scratch para representar cada expresión (Código).

### EJEMPLO 4.

Luisa Fernanda quiere llenar 5 cajas de bombones y sabe que en cada caja hay que incluir 12 bombones de menta, 14 de fresa intensa y 10 de limón. Encontrar al menos dos expresiones equivalentes para calcular el número de bombones que necesita Luisa Fernanda.

### SOLUCION

**Expresión 1:**  $5 * 12 + 5 * 14 + 5 * 10$

*¿Qué significa el producto (multiplicación)?*

*¿Qué significa la suma de los productos?*

**Expresión 2:**  $5 * (12 + 14 + 10)$

*¿Qué representa la suma del paréntesis?*

*¿Por qué la suma se debe multiplicar por 5?*

*¿Las dos expresiones dan el mismo resultado?*

### ACTIVIDAD 10.3.

En el cuaderno resuelve las siguientes expresiones, luego realízalas en Scratch. Observa los resultados. Son los mismos?, Por qué?

númeroA = 5 y númeroB = 8

1.  $20 + \text{númeroA}$

2.  $\text{númeroA} + 3 * \text{númeroB}$

3.  $\text{númeroA} > \text{númeroB}$

4.  $\text{númeroA} + 123$

5.  $4 + \text{númeroA} - \text{númeroB}$