**Algoritmos**

Se puede determinar como un algoritmo a una secuencia de pasos estructurados, ordenados y finitos.

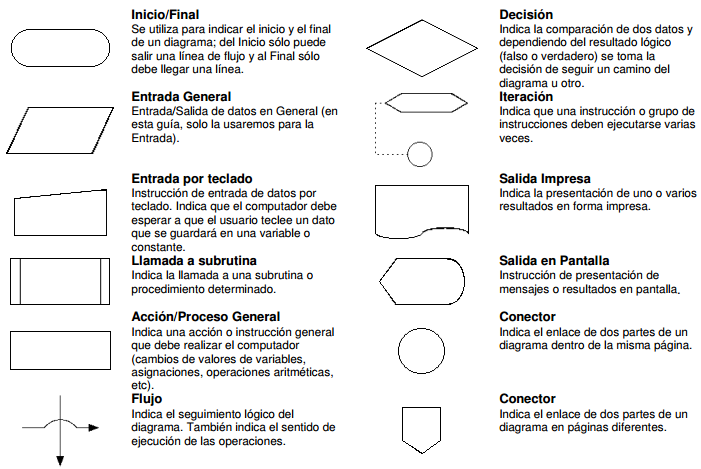
Un software de computación es una secuencia de órdenes que describen un algoritmo, estas deben ser escritas de forma adecuada para que puedan ser interpretadas por el ordenador. Dicho lo anterior se entiende que debe llevarse un proceso o secuencia de acciones que podrían describirse como:

* **Entrada de datos o información**: Son los caracteres de referencia que necesita el algoritmo para dar inicio.
* **Proceso de datos**: Son las consultas u operaciones que realiza el algoritmo.
* **Salida de datos**: Resultados obtenidos.

Para representar estas acciones las herramientas comúnmente más utilizadas en los algoritmos son: Los diagramas de flujo y los Seudocódigos.

**Diagramas de Flujo**

Estos diagramas los podemos describir como graficas de secuencias de acciones a elaborar. Cada proceso que se realiza dentro del diagrama se representa mediante un símbolo normalizado por el ANSI (American National Standars Institute).



**Pseudocódigos**

Son estructuras de algoritmos similares al lenguaje de programación, cuenta con un nivel de complejidad muy bajo para interpretar e implementar a un lenguaje de programación, son más compactos en comparación a los diagramas de flujo.

**Ejemplo**

Calcular la altura en pulgadas (1 pulgada=2.54cm) y pies (1 pie=12 pulgadas), a partir de la altura en centímetros introducida por el teclado.

* **Pseudocódigo**

**Inicio**

1-Imprimir ‘Introduce la altura en centímetros: ‘

2-Leer: altura

3-Calcular pulgada=altura/2.54

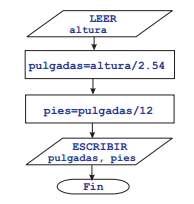
4-Calcular pies=pulgada/12

5-‘Imprimir la altura en pulgadas es:’, pulgada

6-‘Imprimir la altura en pies es:’, pies

**Fin**

* **Diagrama de Flujo**

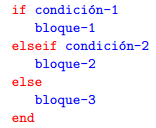
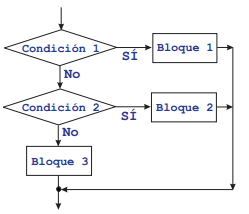


**Condicional multiple IF – ELSEIF - ELSE**.

Esta estructura en su forma más general, nos permite implementar condicionales un poco más complejas en las cuales encadenamos o concatenamos una condición con otra.

Ejemplo

* Si se cumple la condición 1, ejecutar la instancia 1
* Si no se cumple la condición 1, pero si cumple la condición 2, ejecutar la instancia 2
* Y si no cumple con ninguna de las condiciones anteriores, ejecutar la instancia 3
* Diagrama de Flujo

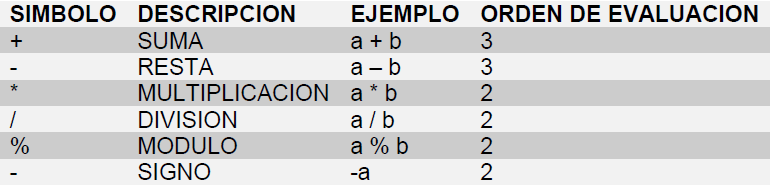
* Pseudocódigo

Inicio

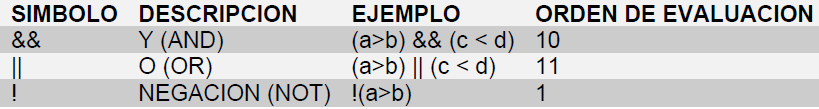
1. Leer X
2. SI X > 0
3. Imprimir ‘El número que digitó es positivo’
4. SI NO, SI X < 0
5. Imprimir ‘El número que digitó es negativo’
6. SI NO,
7. Imprimir ‘El número que digitó es nulo’

Fin

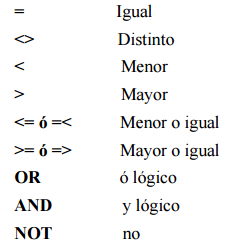
**Operadores Aritméticos:**

****

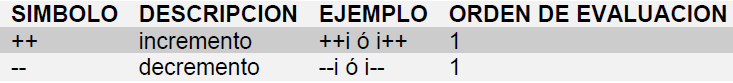
**Operadores lógicos:**

****

**Operadores Relacionales:**

****

**Operador de incremento o decremento**

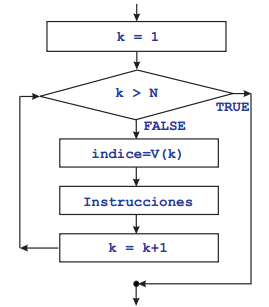
****

**Estructura de repetición FOR**

Esta estructura permite implementar la repetición de un conjunto de órdenes, un número predeterminado de veces.

Para esto se utiliza una variable de control, que también es llamada índice la cual se encarga de tener el control de la cantidad de veces que se repetirá la acción.

Diagrama de flujo



Pseudocódigo

Inicio

1-Leer n

2-Hacer Suma = 0

3-PARA i = 1, 3, 5…., 2\*n-1

4-Hacer Suma = Suma + i

5-Fin PARA

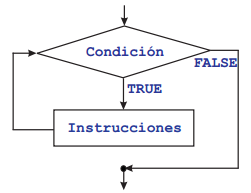
6-Imprimir ‘La suma vale’, Suma

Fin

**Estructura repetitiva WHILE**

Esta estructura permite realizar la repetición de un mismo conjunto de órdenes mientras se verifique una determinada condición, no se define el número de veces que tendrá que repetirse el ciclo.

Diagrama de flujo



Pseudocódigo

Inicio

1. i = 1
2. Mientras que i <=100
3. Imprimir i
4. Hacer i = i + 1
5. Fin Mientras

Fin