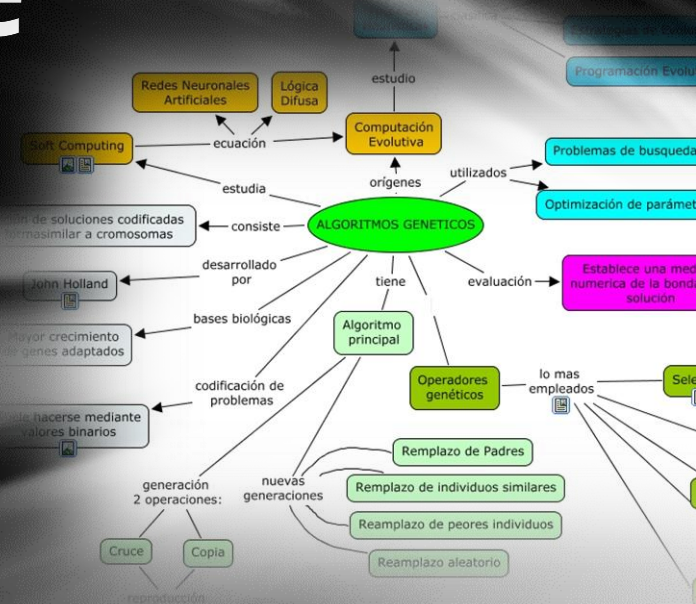


# Fundamentos de Programación

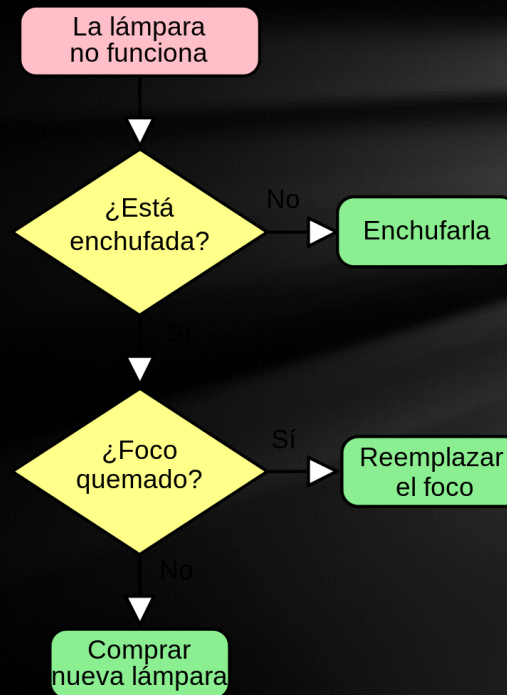
*Algoritmos y pseudocódigo*



# Que es un algoritmo?

- Un algoritmo es como una serie de pasos organizados, que describen el proceso que debe seguir para dar solución a un problema.

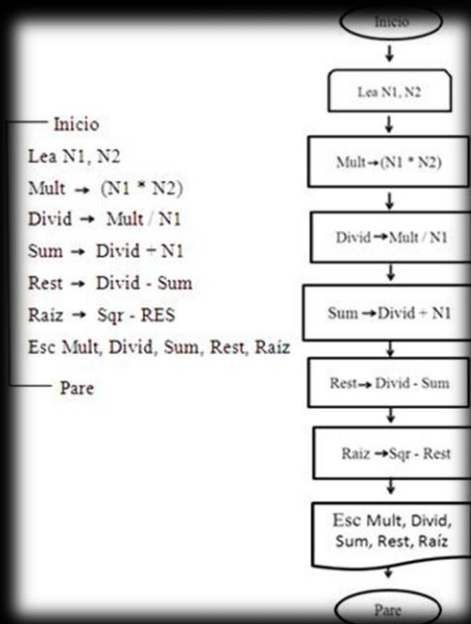
Ejemplo:



# Tipos de algoritmo

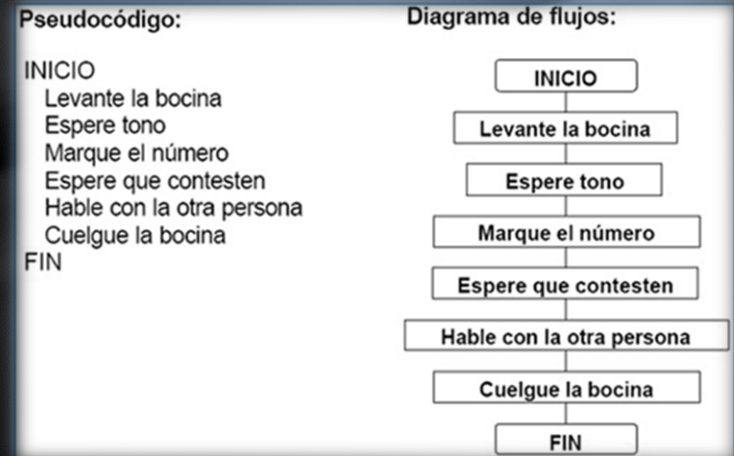
## Cuantitativo

Utilizamos cálculos numéricos para poder definir los pasos.



## Cualitativo

Utilizamos palabras para poder definir los pasos.



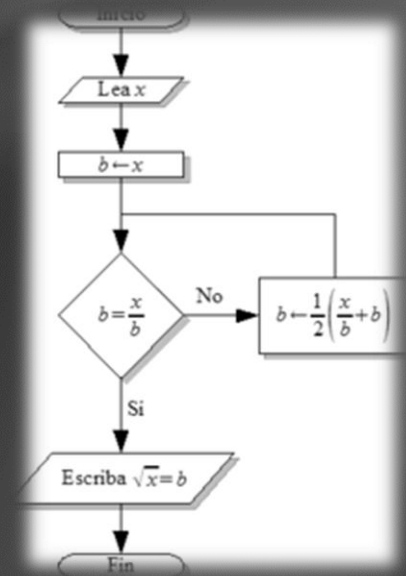
# Lenguajes Algorítmicos

Se utilizan una serie de símbolos y reglas para describir un proceso.

Teniendo en cuenta la forma en que describen el proceso, existen dos tipos de lenguajes algorítmicos:

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 (pseudo código).



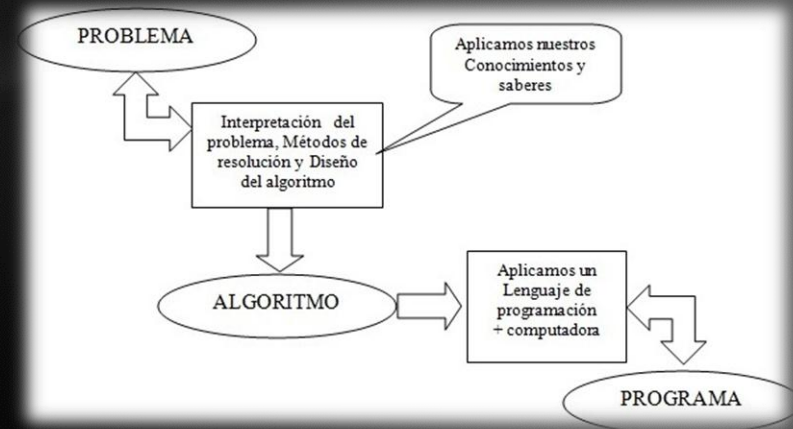
# Las principales características que debe tener un buen algoritmo son:

- Debe tener un punto particular de inicio.
- Debe ser completamente definido y no debe permitir dobles interpretaciones.
- Debe ser general, es decir, soportar la mayoría de las variantes que se puedan presentar en la definición del problema.
- Debe ser finito en tamaño y tiempo de ejecución.
- Debe ser legible, claro, fácil de interpretar y entender.

# Programas

## Codificación del algoritmo en un lenguaje de programación

- E/S: Pasar información del exterior al interior del ordenador y al revés.
- Aritmético-lógicas: Aritméticas:  $+$ ,  $-$ ,  $*$ ,  $...$  ; Lógicas:  $\text{o}$ ,  $\text{y}$ ,  $<$ ,  $>$ ,  $...$
- Selectivas: Permiten la selección de una alternativa en función de una condición.
- Repetitivas: Repetición de un número de instrucciones un número finito de veces.



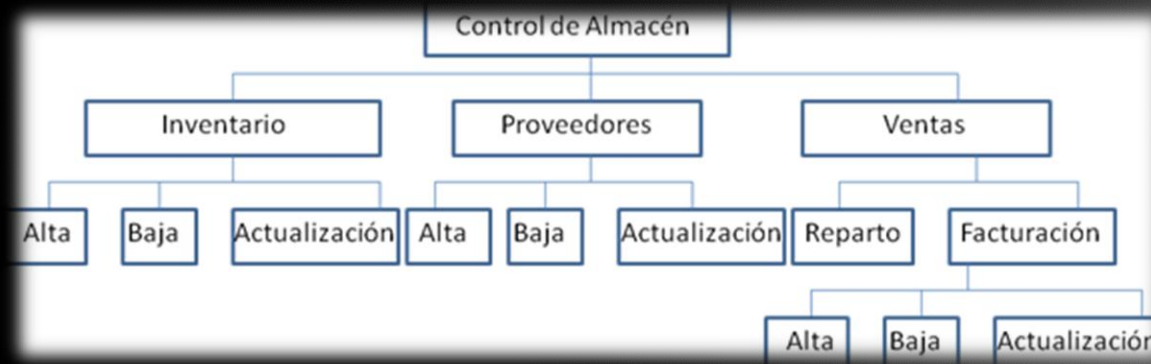
# Codificación

La codificación es el proceso de poner juntos los segmentos de sus datos que parecen ilustrar una idea o un concepto. De esa forma, la codificación es una forma de hacer abstracción a partir de los datos existentes en sus recursos para construir un mayor entendimiento de las fuerzas que intervienen.

- La codificación genera
- La codificación le permite reunir y ver todo el material relativo a una categoría o a un caso a través de todos sus recursos.
- La codificación de sus recursos facilita la búsqueda de patrones y teorías.

# Técnicas de diseño

- TOP DOWN: Es una técnica para diseñar que consiste en tomar el problema en forma inicial como una cuestión global y descomponerlo sucesivamente en problemas más pequeños y por lo tanto, de solución más sencilla.
- BOTTON UP: Esta técnica consiste en partir de los detalles más precisos del algoritmo completando sucesivamente módulos de mayor complejidad, se recomienda cuando ya se cuenta con experiencia y ya se sabe lo que se va a hacer.
- WARNIER ORR: Es una técnica que utiliza una representación semejante a la de cuadros sinópticos para mostrar el funcionamiento y organización de los elementos que conforman el algoritmo.



# Técnicas para la formulación de algoritmos

Hay dos herramientas que tienen mayor uso, que son:

- Diagrama de flujo
- Pseudocódigo

**Ejemplo de un Pseudocódigo para ordenar una lista de elementos (L):**

```
procedimiento Ordenar (L)
  k ← 0
  repetir
    intercambio ← falso
    k ← k + 1
    para i ← 1 hasta n - k hacer
      si  $L_i > L_{i+1}$  entonces
        intercambiar ( $L_i, L_{i+1}$ )
        intercambio ← verdadero
      fin si
    fin para
  hasta que intercambio = falso
fin procedimiento
```

