

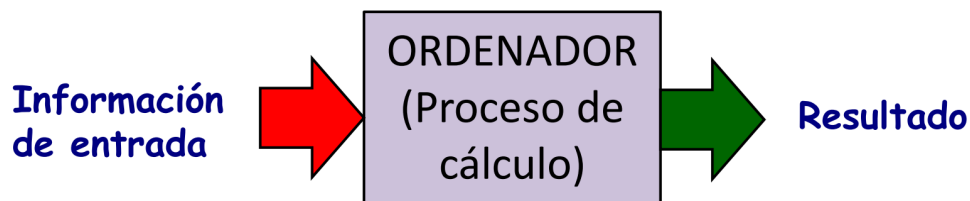
## INTRODUCCIÓN A LA ALGORITMIA

### 1. ¿QUÉ ES UN ALGORITMO?

Puede que sea la primera vez que oyes esta palabra o nunca te hayas parado a pensar en su significado. Un algoritmo básicamente se trata de un conjunto ordenado y finito de operaciones que permite encontrar la solución a un problema cualquiera. Es decir, es un conjunto de instrucciones claras y ordenadas que se le asignan a un ordenador para que proporcione los resultados que deseemos.

Para que un ordenador pueda leer el algoritmo y seguir todos los pasos hasta proporcionar el resultado final le debemos comunicar:

- 1.Cuál es la información de entrada: los datos conocidos.
2. Cómo debe procesar la información: el orden de las instrucciones y las acciones que debe hacer con los datos dados
3. Qué resultado se quiere obtener



Por ejemplo, si queremos que el ordenador sume  $264 + 323$  deberíamos:

1. Introducir información de entrada: 264, 323
2. Proceso de la información: le indicamos que los datos se deben sumar así que el ordenador sigue el paso y lo hace
3. Resultados: el ordenador nos enseñará que el resultado es 587

### CARACTERÍSTICAS DE UN ALGORITMO

Definido: No debe permitir dobles interpretaciones.

General: Debe soportar la mayoría de las variantes que se puedan presentar en la definición del problema.

Finito: En tamaño y tiempo de ejecución.

### LENGUAJES DE PROGRAMACIÓN

A un ordenador, como es lógico, no se le pueden dar las instrucciones como a un humano. Los ordenadores tienen sus propios “idiomas” que se conocen como lenguajes de programación de forma que las instrucciones que queremos que haga debemos “traducirlas” a su lenguaje.

Los lenguajes de programación definen cómo se escriben las instrucciones que entiende el ordenador y el nivel de dificultad de las instrucciones (el nivel que entiende el ordenador) y qué órdenes (o instrucciones) se le pueden dar. El nivel de dificultad y el tipo de instrucciones que se pueden dar a un ordenador suele ser similar en los distintos lenguajes de programación.

Los lenguajes de programación trabajan con 3 elementos:

1. INFORMACIÓN: Esto hace alusión a los datos con los que trabajarán los programas. Pueden ser de dos tipos: numéricos y alfanuméricos (caracteres).

Los datos se pueden agrupar en estructuras muy simples (constantes o variables) o más complejas formando matrices o registros.

2. OPERACIONES: Son los cálculos que se pueden realizar con la información. Las operaciones básicas que se pueden hacer y que nos van a servir a nosotros son:

suma: +

resta: -

Multiplicación: \*

Potenciación: ^

División: /

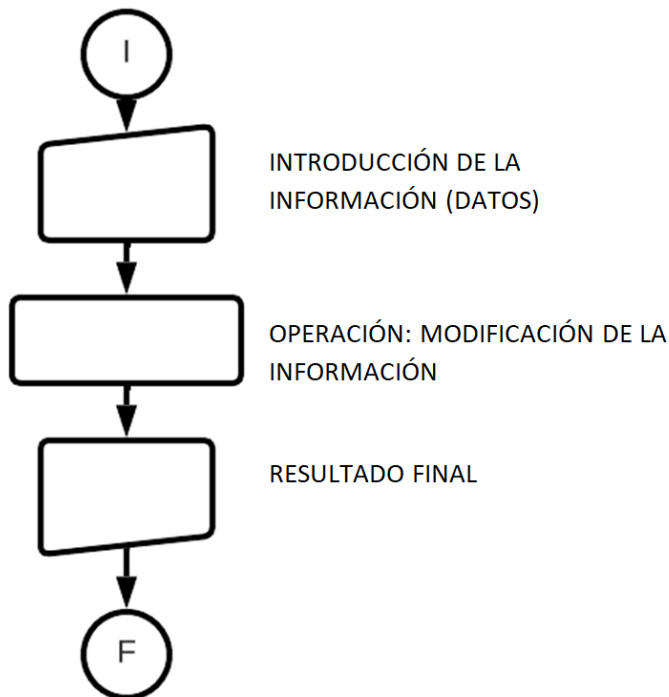
3. INSTRUCCIONES: Las que le proporcionamos nosotros al ordenador a través del lenguaje de programación.

## 2. ESTRUCTURA BÁSICA DE UN ALGORITMO: DIAGRAMA DE FLUJO

### ORGANIGRAMAS

Los organigramas o diagramas de flujo son representaciones gráficas de los algoritmos, es decir, como si cogiéramos un algoritmo (en un determinado lenguaje de programación que a primeras es difícil de entender) y decidimos hacer un esquema para poder entenderlo mejor.

Este esquema está formado a base de símbolos conectados con flechas que indican el camino que sigue la información aportada y en cada símbolo por el que pasa dicha información, esta se amplía o se modifica.



## SÍMBOLOS QUE UTILIZAREMOS A LA HORA DE REPRESENTAR UN ALGORITMO:

### 1. COMIENZO Y FINAL DEL ALGORITMO

Siempre vamos a iniciar y finalizar el algoritmo con un círculo. En el caso de inicio dentro del círculo pondremos la letra I y para indicar que se ha acabado la letra F.



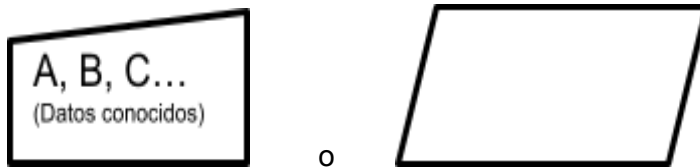
### 2. LÍNEA DE FLUJO

Se trata de una flecha que sale de un símbolo y apunta al siguiente. Muestra el orden de operación de los procesos.



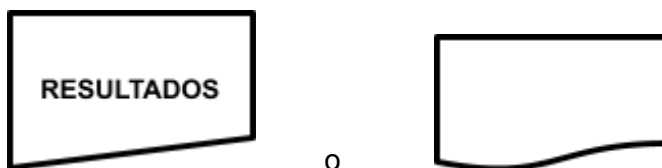
### 3. ENTRADA DE DATOS

Este símbolo se sitúa al inicio del algoritmo justo después del inicio y sirve para indicar que esa es la información que le damos al ordenador.



### 4. SALIDA DE DATOS

Con este símbolo se sitúa antes del final del algoritmo y con este indicamos los resultados que nos proporciona el ordenador tras seguir todas las instrucciones anteriores.



### 5. DESCRIBIR OPERACIONES (PROCESO)

Representa un conjunto de operaciones que cambian el valor, forma o ubicación de datos que pasan siguiendo el flujo de información. En su interior se indica las modificaciones que se quieren hacer a la información.



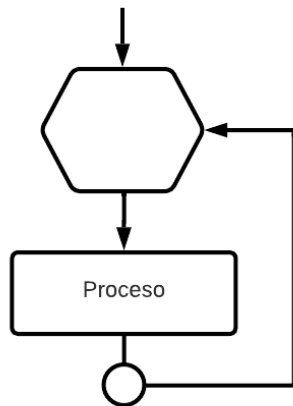
### 6. BUCLES

Estas son estructuras repetitivas, es decir, indican que cuando la información pasa por ellas vuelve a pasar las veces que se indiquen de forma que

cuando la información pasa se modifica y la nueva información vuelve al inicio del bucle modificándose de nuevo.



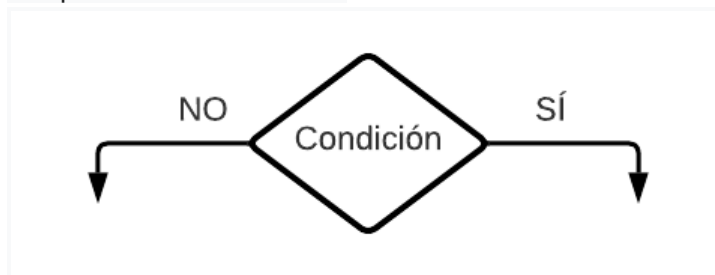
Generalmente los encontraremos de la siguiente manera:



Estos se explicarán más a fondo en el apartado de bucles.

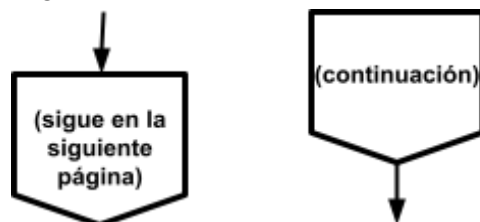
## 7. CONDICIONALES

Muestra una operación condicional que determina cuál de los dos caminos tomará el programa. Se trata de una pregunta de si o no de forma que dependiendo de la información que contenga el flujo, tomará un camino u otro si cumple o no esa condición.



## 8. CONECTOR

Este símbolo se utiliza cuando queremos continuar el algoritmo en otra página o en otra columna:



Dentro de estos suele haber números ya que en un algoritmo complejo puede haber varios y así se sabe qué línea de flujo continúa en cada caso.

### 3. CONCEPTO DE VECTOR Y MATRIZ

A pesar de que normalmente tenemos muy interiorizado el concepto de vector y matriz de forma matemática, en algoritmia estos son términos que tienen significados totalmente distintos así que lo primero que hay que hacer es dejar a un lado la definición de vector y matriz dada hasta ahora e introducirlos de nuevo.

Un **vector** se trata de una zona de almacenamiento contiguo que contiene una serie de elementos denominados variables.



(los índices de las v indica la posición de la variable)

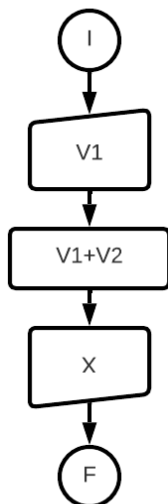
Esto lo que quiere decir es que el vector  $v$  está formado por 3 variables que se encuentran guardadas en la memoria del ordenador de forma que  $v_1=5$ ,  $v_2=2$  y  $v_3=1$ . Los vectores pueden tener las columnas que uno quiera pero siempre una misma fila. En algoritmia cuando usamos elementos de un vector ponemos:  $v_i$  ;  $v(i)$  ;  $v[i]$ .

Las **matrices** funcionan de forma similar a los vectores solo que están formadas por más de una fila. Por ello además del índice que indica la columna en la que estamos, debemos indicar la fila en la que se encuentra la variable que nos interesa.

Para matrices utilizamos:  $A_{i,j}$  ;  $A(i,j)$  ;  $A[i,j]$ .

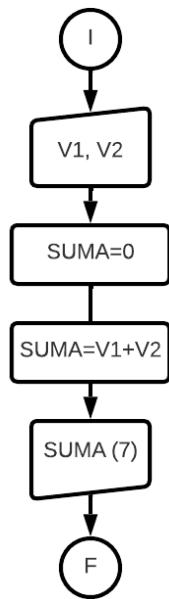
**OJO!** Es muy importante a la hora de escribir un algoritmo introducir bien los datos, es decir, introducir todas las variables que vamos a utilizar a lo largo del proceso que no tienen valor.

#### EJEMPLO



En este caso el algoritmo no funcionaría ya que el ordenador iría a la posición  $v_1$  y adoptaría el valor asignado, en el caso del vector anterior 5. Cuando pasa al proceso, el ordenador verá que tiene que sumar  $v_1(5)$  y  $v_2$ , pero  $v_2$  no sabe lo que vale ya que no lo hemos indicado por lo que no nos proporcionaría un resultado y daría error.

Otra cosa a tener en cuenta es que cada vez que hay una operación se suele igualar a 0 para asegurar que el resultado no varía y crear un nombre o espacio para el resultado de dicha operación.



De esta forma el ordenador procesa que tiene que ir a las posiciones v1 y v2 donde se encuentran las variables 5 y 2 respectivamente.

Luego creará un espacio en su memoria donde la posición se llama SUMA y la variable en esa posición es 0.

Después va a la posición SUMA dentro de su memoria y en la posición suma coloca la operación a la que se iguala ( $v1+v2$ ). Como conoce dichos valores realiza la operación.

Por último, le indicamos que nos muestre qué hay en la posición SUMA, por lo que nos indicará el resultado de la operación, en este caso 7.