

# LECTURA 2: INTRODUCCIÓN A LA PROGRAMACION I

**Los fundamentos de programación** son las bases comunes a todos los programas. Es lo primero que tendrás que aprender incluso antes de elegir el programa con el que quieres programar.

Lo primero que tienes que saber es que el ordenador solo entiende el llamado código binario (1 y 0).

1 = hay corriente

0 = No hay corriente



No te olvides que solo es una máquina eléctrica. Entendernos con él mediante este código es muy difícil, por eso los lenguajes de programación se dividen en dos tipos diferentes dependiendo de su cercanía al lenguaje del ordenador.

**Los lenguajes más cercanos al idioma del ordenador, llamados de bajo nivel**, son muy complicados (casi como el código máquina) y poco usados. El más conocido es el código o lenguaje máquina, un código que el ordenador puede interpretar directamente. Aquí tienes un ejemplo:

```
8B542408 83FA0077 06B80000 0000C383
```

Los **lenguajes de programación de alto nivel** permiten dar órdenes al ordenador con un lenguaje parecido al nuestro (Visual Basic, Pascal, Logo, C++, JavaScript, etc.) y siempre o casi siempre en inglés.

Hay programas de alto nivel como el GML o el Java que son programas interpretados, es decir, se analizan y ejecutan las instrucciones por el propio programa directamente. Otros necesitan un compilador, pero eso no es un problema, solo es un programa (software) que se encarga de traducir el programa hecho en lenguaje de programación para que lo entienda el ordenador.

Con un tipo u otro es igual, lo importante es que los lenguajes, como todo, hay que aprendérselos, pero tienen una ventaja, y es que hay varios puntos en común. Estos puntos son lo que vamos a estudiar aquí, los **fundamentos de programación** común a cualquier lenguaje de alto nivel.

Una vez aprendidos los fundamentos, tendrás que elegir el lenguaje que quieras usar, pero con estos conocimientos, todos te resultarán muy fáciles de aprender, solo tendrás que aprender unas cuantas instrucciones en inglés.

## Programas y Algoritmos

Los lenguajes de programación, cuentan todos en su haber con un juego de "instrucciones". Una instrucción no es más que una orden que nosotros le damos a la máquina.

Y es que, al fin y al cabo, **un programa no es más que una secuencia de instrucciones (escritas en algún lenguaje de programación) pensado para resolver algún tipo de problema**. Hay que tener claro que si no sabemos resolver este problema, no podremos escribir el programa. Si no sabemos que es una suma, sería casi imposible hacer un programa para que nos sume dos números, a no ser que alguien nos ayudara. A ti se te puede ocurrir una manera de resolverlo, a tu compañero, otra, lo importante es que las dos formas de resolverlo lleven al mismo resultado. La forma como resuelves el problema, es lo que se llama **algoritmo**, y es lo que vamos a ver a continuación.

**¿Qué es un Algoritmo?** Un algoritmo es una secuencia de PASOS a seguir para resolver un problema.

Por ejemplo, cuando quiero ver una película de vídeo, podría hacer los siguientes pasos (algoritmo):

- Elijo una película de las de mi colección.
- Compruebo SI TV y vídeo están conectados a la red (y procedo).
- SI la TV está apagada, la enciendo, SI NO, pues no. Y lo mismo con el vídeo.
- Introduzco la película en el vídeo. Dejo el estuche sobre el vídeo.
- SI la TV no está en el canal adecuado, la cambio, SI NO, pues no.
- Cojo los mandos a distancia (el del TV y el del vídeo).
- Me pongo cómodo.
- Pulso PLAY en el mando del vídeo.

Fíjate bien en unos detalles que son fundamentales y que aparecen en este algoritmo:

- ✓ La descripción de cada paso no me lleva a ambigüedades, los **pasos** son absolutamente **explícitos y no inducen a error**.
- ✓ **El número de pasos es finito**. Tienen un principio y un fin.

Según lo visto, una mejor definición de algoritmo sería: **“Un algoritmo es una sucesión finita de pasos (no instrucciones como en los programas) no ambiguos que se pueden llevar a cabo en un tiempo finito.”**

Este "lenguaje" el algoritmo está escrito en nuestro idioma, pero ahora necesitamos acercarnos a un poco más al lenguaje del ordenador. Pero **el primer paso para realizar un programa es sacar su algoritmo**, es como explicar lo que queremos que haga nuestro programa. Ahora entiendes por qué decíamos antes que si no sabemos nosotros resolver el problema, no podríamos crear el programa. No seríamos capaces de hacer su algoritmo.

Ahora que ya tenemos el algoritmo, para el siguiente paso se puede utilizar dos formas: Sacar el **diagrama de flujo** del algoritmo o **su pseudocódigo**. Algunos programadores hacen los dos.

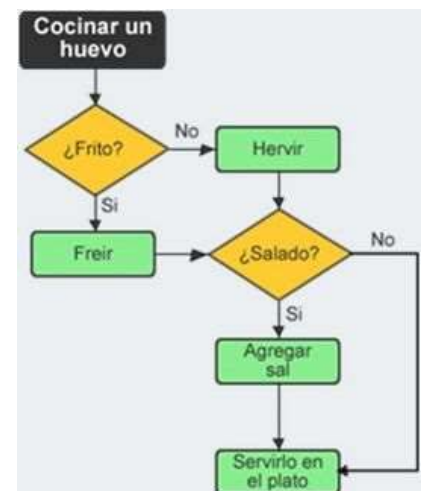
**Diagrama de Flujo.** Un diagrama de flujo es una representación gráfica del algoritmo. Expresamos los pasos del algoritmo mediante un esquema con unos símbolos establecidos. Un diagrama de flujo debe proporcionar una información clara, ordenada y concisa de todos los pasos a seguir. **Los diagramas de flujo** representan la secuencia o los pasos lógicos para realizar una tarea mediante unos símbolos. Dentro de los símbolos se escriben los pasos a seguir. Un diagrama de flujo debe proporcionar una información clara, ordenada y concisa de todos los pasos a seguir. Veamos un ejemplo:

Un algoritmo para cocinar un huevo para otra persona sería:

- Pregunto si quiere el huevo frito.
- Si me dice que sí, lo frío, si me dice que no, lo hago hervido.
- Una vez cocinado le pregunto si quiere sal en el huevo.
- Si me dice que no lo sirvo en el Plato. Si me dice que si le hecho sal y después lo sirvo en el plato.

Ahora que ya sabemos todos los pasos, mediante el algoritmo, podemos hacer un esquema con estos pasos a seguir. Este esquema será el Diagrama de Flujo.

Si uno tiene experiencia puede prescindir del algoritmo escrito, pero siempre tendremos que tenerlo en mente para hacer el diagrama de flujo sin equivocarnos.

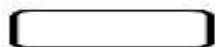


## Reglas Básicas Para la Construcción de un Diagrama de Flujo

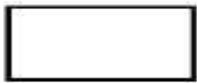
- ❖ Todos los símbolos han de estar conectados
- ❖ A un símbolo de proceso pueden llegarle varias líneas
- ❖ A un símbolo de decisión pueden llegarle varias líneas, pero sólo saldrán dos (Si o No, Verdadero o Falso).
- ❖ A un símbolo de inicio nunca le llegan líneas.
- ❖ De un símbolo de fin no parte ninguna línea.
- ❖ En el Símbolo de decisión a tomar los valores de salida pueden ser SI o NO o VERDADERO o FALSO.
- ❖ El símbolo de Inicio o Final del Diagrama pueden ser un cuadrado con los bordes redondeados o una elipse.
- ❖ Se pueden utilizar colores para los símbolos

Los símbolos que se usan para realizar los diagramas de flujo son los siguientes

### SIMBOLOS FUNDAMENTALES



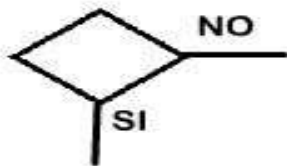
INICIO Y FIN DEL PROCESO



REALIZAR UN PROCESO (OPERACIÓN MATEMÁTICA POR EJEMPLO)



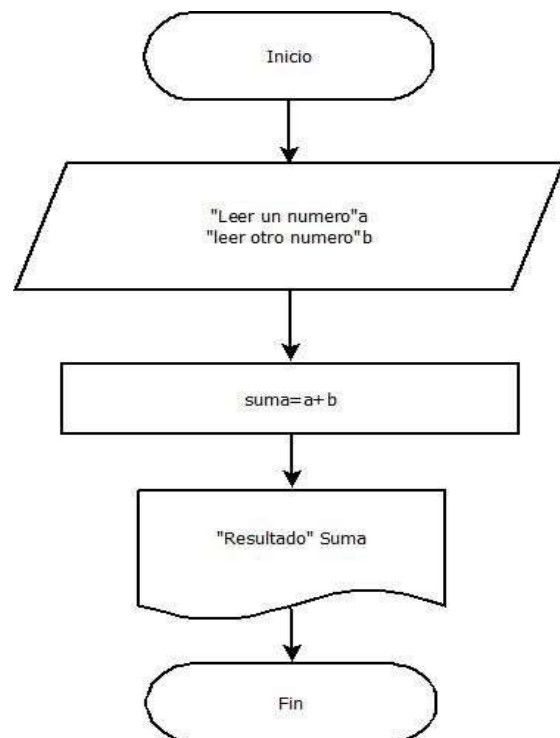
ENTRADA DE DATOS Y/O SALIDA DE DATOS



TOMAR UNA DECISIÓN (UNA PREGUNTA).  
LA RESPUESTA A LA PREGUNTA PUEDE SER SI O NO

**Ejemplos de Diagramas de Flujo:** Veamos un primer ejemplo muy sencillo.

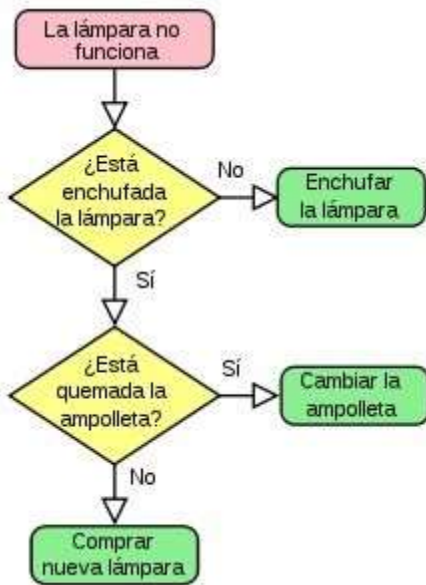
Queremos hacer un programa informático que nos sume dos números y nos dé el resultado en pantalla.



El símbolo de resultado es un símbolo usado en los diagramas para soluciones con el ordenador. Es el símbolo de salida del resultado por la pantalla del ordenador. Ves que es muy sencillo, hay que ir poniendo los pasos lógicos que se deben seguir para realizar la tarea o el programa.

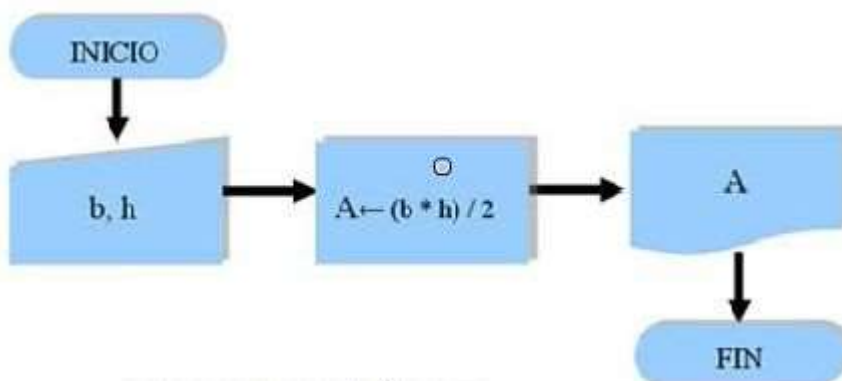
En el ejercicio tenemos el inicio y el fin, una entrada de datos, para meter los 2 números, una operación a realizar, la suma, y un resultado a mostrar. Cada uno de esos pasos con su símbolo correspondiente en el diagrama.

**Otro ejemplo** de un diagrama de flujo para una operación sencilla: Imaginemos que tenemos una lámpara o bombilla y queremos hacer el diagrama de flujo para saber qué hacer cuando esta no funciona.



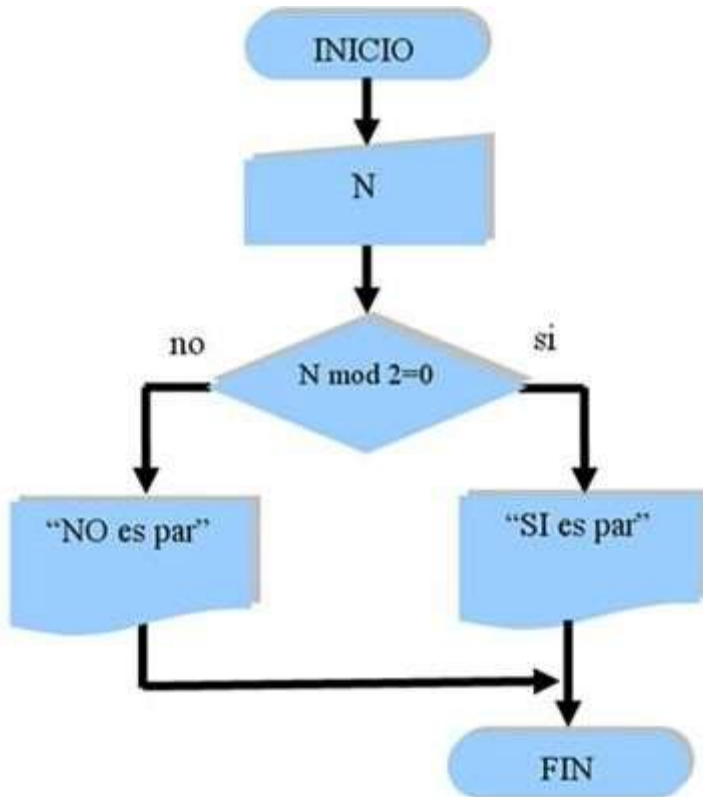
**¿Hacemos otro?**

Bueno vamos hacer uno que nos muestre el resultado del área de un triángulo en pantalla.



Como ves en este ni siquiera hemos puesto las operaciones dentro de los símbolos ya que con la forma del símbolo ya se entiende.

No me hemos usado mucho el símbolo de tomar un decisión, por eso vamos hacer **uno en el que nos diga si el número es par o impar**:

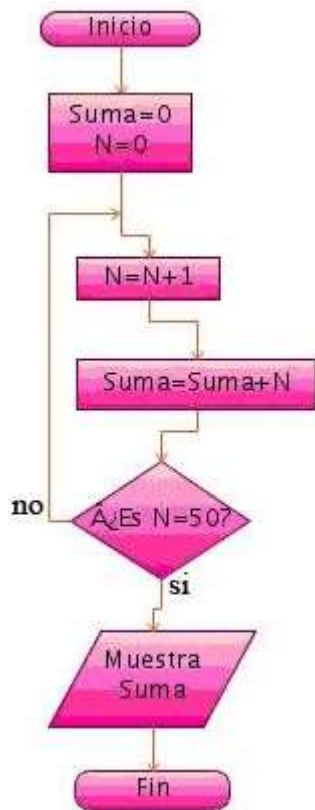


Si te das cuenta mod significa dividir entre 2. Como ya debes saber si divido un número entre 2 y el resto es 0 el número es par. Bien pues hay esta la decisión. ¿Al dividirlo entre 2 el resto es 0? Hay 2 posibilidades. Si lo es se ve en pantalla "Si es par", si no lo es se ve en pantalla "No es par". Eso es la toma de decisiones.

Además los diagramas de flujo no solo valen para informática, incluso podemos hacer uno para cocinar un huevo, como vimos al principio.

Bueno ahora vamos a hacer uno un poco más complicado. Tenemos que hacer **un diagrama de flujo para mostrar la suma de los 50 primeros números**. Lo primero es poner a cero la suma y dar el primer número a sumar que será el 0.

Fíjate que el diagrama acaba cuando N, que es el número en cada momento, es 50. Mientras no sea 50 el programa vuelve a la tercera secuencia que será sumarle un número al anterior  $N = N + 1$ . Intenta comprenderlo y ver que hace. Puedes realizar mentalmente el diagrama para el número 0 y verás cómo lo acabas entendiendo.



## PREGUNTAS DE CONTROL

1. ¿Cuál son las bases comunes de los programas?
2. ¿Cuál es el código que entienden la maquina?
3. ¿Cómo se llaman los programas que están más cerca al lenguaje máquina y cuáles son los más fácil de usar?
4. ¿Qué es una instrucción, un programa y un algoritmo?
5. ¿Cómo escribirías el algoritmo para los siguientes problemas?
  - A) Cepillarse los dietes
  - B) Dibujar un cuadrado
6. ¿Cuáles son las dos características básicas de un algoritmo y con base en ellas elabore un definición más completa de algoritmo?
7. ¿Cuál es el primer paso que se deba dar para elaborar un programa?
8. ¿Cuáles son las dos formas que existen para expresar un algoritmo?
9. ¿Qué es un diagrama de flujo?
10. ¿Cuáles son las reglas básicas para construir un diagrama de flujo?
11. ¿Cuáles son los símbolos que más se usan en la construcción de un diagrama de flujo?
12. Diseñar el diagrama de flujo para dibujar un rectángulo con un lado1 que mide 80 y un lado 2 que mide 120
13. Diseñar el diagrama de flujo que permita calcular el área de un triángulo que tienen 20 de base y 30 de altura
14. Diseñar el diagrama de flujo para calcular el área de cualquier triángulo rectángulo
15. Hallar uno de los lados de un rectángulo cuya área es de 25 cm<sup>2</sup> y uno de sus lados mide 5 cm.
16. Hallar el área y el perímetro de un círculo cuyo radio mide 2 cm
17. Hallar el área de un pentágono regular de 6 cm de lado y 4 cm de apotema
18. Hacer el diagrama de flujo para sumar dos números leídos por teclado y escribir el resultado.
19. Hacer un diagrama de flujo que permita leer 2 números diferentes y nos diga cuál es el mayor de los 2 números
20. Crear un diagrama de flujo de procesos en el que se almacenen 3 números en 3 variables A, B y C. El diagrama debe decidir cuál es el mayor y cuál es el menor