

## Algoritmos y estructuras de programación

- 5.1. Fases de creación de un programa
- 5.2. Algoritmos
- 5.3. Representación de Algoritmos
- 5.4. Estructura secuencial

## 5.1 Fases de creación de un programa

El proceso de resolución de problemas en un ordenador conduce a la escritura de un **programa** y su ejecución. Las fases en el desarrollo de un programa pueden resumirse de la siguiente forma:

1. **Analizar el problema** consiste en conocer perfectamente en qué consiste y qué resultados se desean obtener.
2. **Planificación** de la resolución del problema, dividiéndolo, si es complicado, en una secuencia de etapas más simples. Esta fase se lleva a cabo EN UN PAPEL, estableciendo lo más claramente posible la finalidad de cada etapa, los datos que se necesitan de entrada, los datos que producirían en salida, los **algoritmos** (ver la Sección 5.2) que se utilizarán, etc.
3. **Edición** del código fuente, es decir, escritura del mismo utilizando un editor de textos simple (sin formato) y un lenguaje de programación. Los programas fuente serán almacenados en ficheros de texto, normalmente en el disco duro del ordenador.
4. **Compilación y ejecución** del programa al lenguaje máquina.
5. **Corrección de errores** del programa. Los errores se corregirán en el código fuente, repitiendo los pasos 3 y 4 tantas veces como sea necesario. Si se producen errores en la lógica del programa, es decir, si el programa “funciona” pero produce resultados incorrectos, hay que modificar el algoritmo volviendo al paso 2. Estos errores son los más difíciles de detectar.
6. **Documentación**. Una vez que el programa funcione correctamente, es conveniente **revisar** el código fuente para ordenarlos, eliminar cálculos innecesarios e incluir las líneas de comentario necesarias, que normalmente deben incluir unas **breves** explicaciones al principio del código sobre la finalidad del programa y sus argumentos de entrada y de salida.

## 5.2 Algoritmos

Un ordenador es capaz de realizar “sólo” determinadas acciones sencillas, tales como sumar, comparar o transferir datos, pero los problemas que normalmente interesa resolver son más complejos. Para resolver un problema real es necesario, en primer lugar, encontrar un método de resolución y, posteriormente, determinar la sucesión de acciones sencillas (susceptibles de ser ejecutadas por un ordenador) en que se descompone dicho método.

No todos los métodos de solución de un problema pueden ser puestos es práctica en un ordenador. Para que un procedimiento pueda ser implantado en un ordenador debe ser:

- **Preciso:** estar compuesto de pasos **bien definidos** (no ambiguos) y **ordenados**.
- **Definido:** si se sigue dos veces, se obtiene el mismo resultado cada vez.
- **Finito:** tener un número finito de pasos.

Un procedimiento o método para resolver un problema que cumpla los requisitos anteriores se dice que es un **algoritmo**. Se puede dar por tanto la siguiente definición:

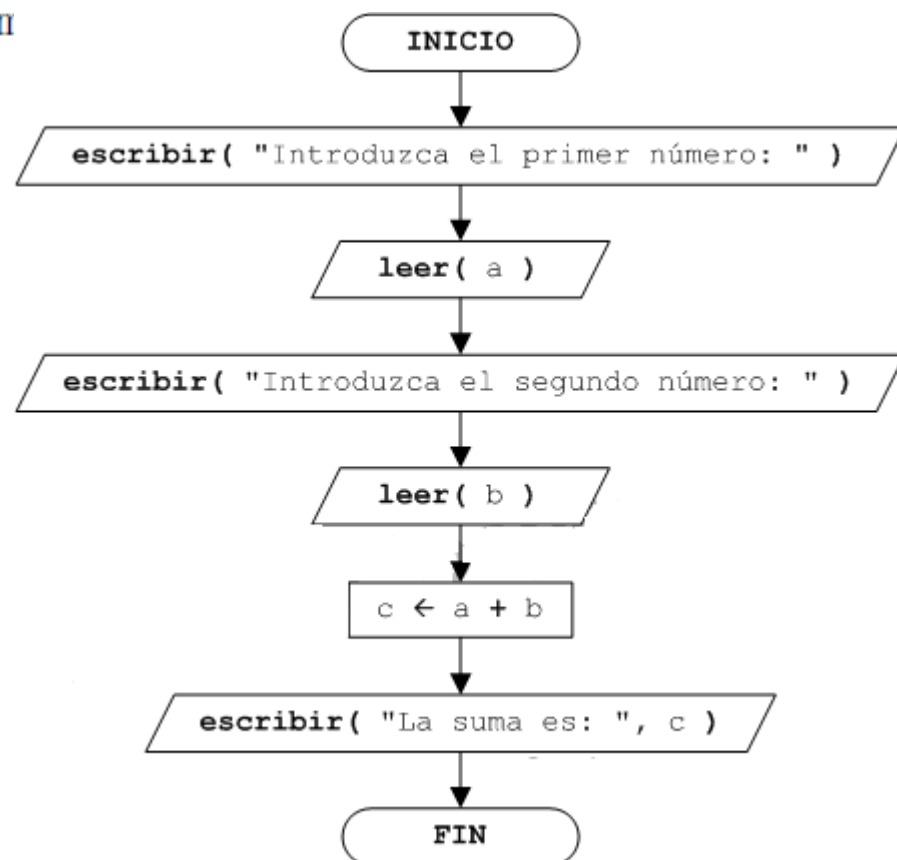
Un **algoritmo** es un método para resolver un problema mediante una secuencia de pasos bien definidos, ordenados y finitos.

Para que se pueda ejecutar el algoritmo es preciso, además, que se disponga de las “herramientas” adecuadas para llevar a cabo cada uno de los pasos. Si no es así, estos deberán, a su vez, ser descompuestos en una secuencia (algoritmo) de pasos más simples que sí se puedan llevar a cabo.

Un **programa de ordenador** es una sucesión de órdenes que describen un algoritmo, escritas de forma que puedan ser entendidas por el ordenador.

En un algoritmo (y por tanto en un programa) se distinguen las siguientes acciones:

- **Entrada:** es la información de partida que necesita el algoritmo
- **Proceso:** es el conjunto de todas las operaciones a realizar.
- **Salida:** son los resultados obtenidos.



## 5.3 Representación de algoritmos

Las dos herramientas más utilizadas comúnmente para describir algoritmos son:

**Diagramas de flujo:** son representaciones gráficas de secuencias de pasos a realizar. Cada operación se representa mediante un símbolo normalizado el Instituto Norteamericano de Normalización (ANSI - American National Standards Institute). Las líneas de flujo indican el orden de ejecución. Algunos de los símbolos principales se muestran en la Figura 5.1, como son: **Inicio/Fin** del algoritmo, **Lectura/Escritura** de datos que el programa necesita o genera (por ejemplo, lectura de datos que se teclean o escritura de datos en un fichero); **Proceso** conjunto de instrucciones secuenciales; **Decisión** es una bifurcación en el flujo del algoritmo en base a que se verifique o no cierta condición (ver la Sección 5.5).

Los diagramas de flujo suelen ser usados sólo para representar algoritmos pequeños, ya que abarcan mucho espacio.

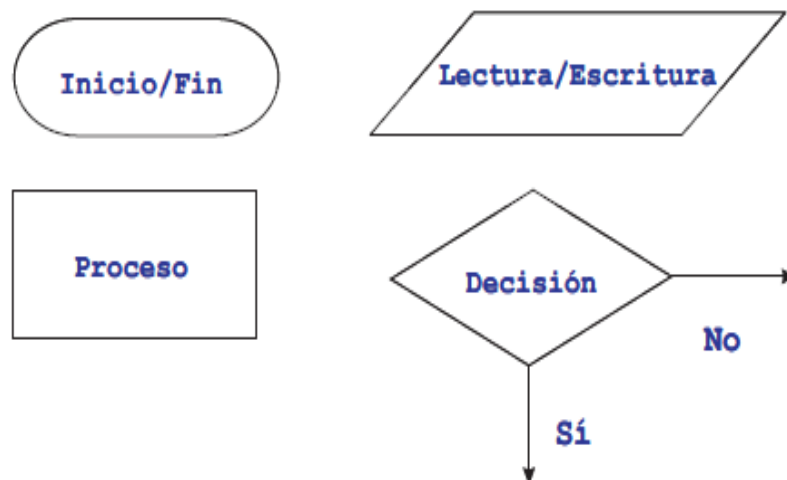


Figura 5.1: Símbolos en diagramas de flujo.



**Pseudocódigos:** describen un algoritmo de forma similar a un lenguaje de programación pero sin su rigidez, de forma más parecida al lenguaje natural. Presentan la ventaja de ser más compactos que los diagramas de flujo, más fáciles de escribir para las instrucciones complejas y más fáciles de transferir a un lenguaje de programación. El pseudocódigo no está regido por ningún estándar.

En estos apuntes usaremos las palabras **LEER**, **ESCRIBIR** para representar las acciones de **lectura de datos** (el programa recibe datos desde algún sitio) y **salida de datos** (el programa escribe información en algún medio)

El Algoritmo 5.2 y la Figura 5.2 muestran respectivamente el pseudocódigo y el diagrama de flujo del algoritmo para calcular la altura de una persona en pulgadas y pies a partir de la altura en centímetros introducida por el teclado.

**Algoritmo 5.2** Calcular una altura en pulgadas (1 pulgada=2.54 cm) y pies (1 pie=12 pulgadas), a partir de la altura en centímetros, que se introduce por el teclado.

```
Inicio
1- ESCRIBIR: 'Introduce la altura en centímetros: '
2- LEER: altura
3- CALCULAR  $pulgadas = altura / 2.54$ 
4- CALCULAR  $pies = pulgadas * 12$ 
5- ESCRIBIR: 'La altura en pulgadas es: ', pulgadas
6- ESCRIBIR: 'La altura en pies es : ', pies
Fin
```

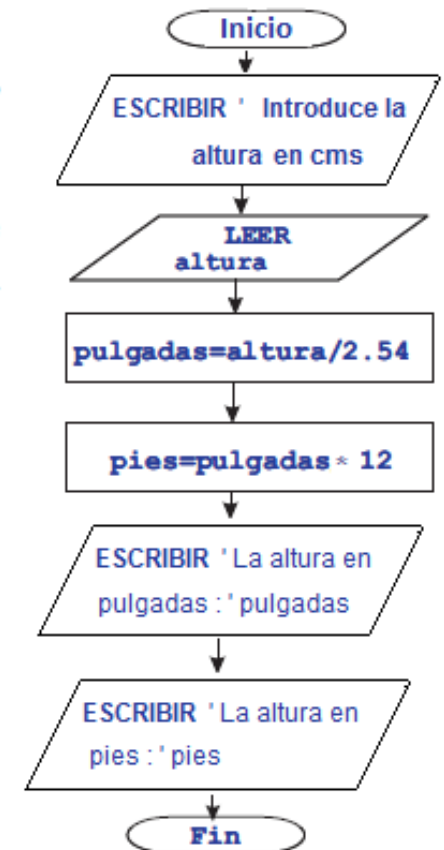


Figura 5.2: Diagrama de flujo para calcular la altura de una persona en pulgadas y pies a partir de la altura en cms. Entrada por el teclado.

## 5.4 Estructura secuencial

Es aquella en la que una acción (instrucción) sigue a la otra en el orden en el que están escritas. Su representación y el diagrama de flujo se muestra en la Figura 5.3. Los Algoritmos 5.1 y 5.2 son ejemplos de algoritmos secuenciales.

...  
Instrucción 1  
Instrucción 2  
Instrucción 3  
...

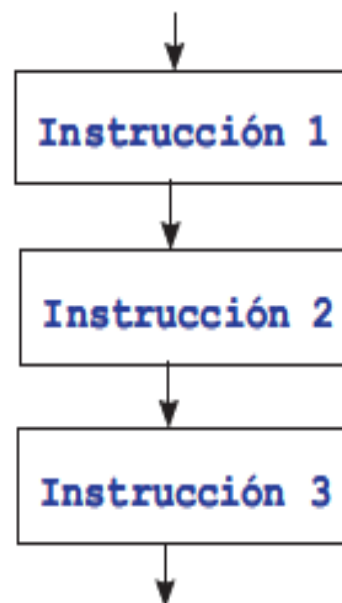


Figura 5.3: Estructura secuencial de instrucciones.

# Práctica P01

## CELSIUS TO FARENHEITH

1. Escribe con Notepad++ un algoritmo que pida la temperatura de grados Celsius por teclado y calcule y muestre la temperatura en grados Fahrenheit (\*). Llama al algoritmo : [cels\\_to\\_far.txt](#)

\* Nota = en este caso busca tu mismo como se realiza la conversión

2. Realiza un programa python usando el algoritmo como base del programa. Llama al programa : [cels\\_to\\_far.py](#)

3. Dibuja el diagrama de flujo con Paint y lláma al nuevo diagrama : [cels\\_to\\_far.jpg](#) *(Si quieres usa la imagen .jp que te proporciona el tutor)*