

# **INTRODUCCIÓN A LA ALGORITMIA**

## **CAPITULO 1. ELEMENTOS BÁSICOS**

[Ángel Fidalgo Blanco](#)

Universidad Politécnica de Madrid

Licencia Creative Commons Algunos derechos reservados



## **UNIDAD DIDÁCTICA Nº 1.**

### **ELEMENTOS BÁSICOS**

#### **OBJETIVOS:**

- \* Conocer qué es la programación.
- \* Conocer qué es la algoritmia.
- \* Saber cuáles son los elementos que componen los programas.
- \* Saber cuáles son los elementos que componen la algoritmia.

## **CONTENIDOS:**

1. Introducción.
2. Pasos previos a la programación de ordenadores. Algoritmia.
3. Elementos de los lenguajes de programación y de los algoritmos.
  - 3.1. Elementos de los lenguajes de programación.
    - 3.1.1. La información.
    - 3.1.2. Las operaciones.
    - 3.1.3. Las instrucciones.
  - 3.2. Otros tipos de clasificación para las instrucciones.
  - 3.3. Elementos de los algoritmos.
    - 3.3.1. La información.
    - 3.3.2. Las expresiones.
    - 3.3.3. Las instrucciones.
    - 3.3.4. Flujo de instrucciones.

## **CONOCIMIENTOS PREVIOS:**

- \* Estructura y funcionamiento de un ordenador personal (conveniente, no imprescindible)

### **1. INTRODUCCIÓN.**

Puede que haya visto realizar con un mismo tipo de ordenador diversas tareas; por ejemplo, jugar a "marcianitos", hacer la contabilidad de una empresa, escribir textos, realizar dibujos animados, hacer operaciones complejas y últimamente lo que se denomina realidad virtual.

¿Cómo una misma máquina (el ordenador) puede realizar tareas tan diversas? Realmente un ordenador lo único que hace es procesar información; es decir, toma una información de entrada, la procesa y genera una información de salida. Así en el juego de marcianos la información de entrada puede ser el movimiento de una nave marciana y la trayectoria del disparo; el procesamiento puede ser comprobar si la trayectoria del disparo

intercepta al movimiento de la nave y la información de salida puede ser la explosión o destrucción de la nave marciana.

Para comprender mejor la idea de ordenador y programa pensemos en cómo se utiliza un aparato reproductor de video. El video lo utilizamos para ver películas que alquilamos o compramos. Cada película viene en un video-cassette, y para ver una película lo único que se debe hacer es introducir el video-cassette en el aparato reproductor de video. En función del video-cassette utilizado veremos una u otra película. El video lo único que hace es reproducir la información del video-cassette a través de una serie de imágenes. Un programa sería equivalente a un video-cassette y un ordenador al aparato reproductor de video. En ambos casos el ordenador o el video lo único que hacen es reproducir la información del programa o del video-cassette.

A un ordenador se le debe de indicar a través de una serie de mandatos cual es la información de entrada, como la procesa y cuál es la información de salida. **Al conjunto de mandatos se le denomina programa**, entonces, tendríamos un programa para jugar a "marcianitos"; un programa que realiza la contabilidad; un programa para escribir textos, etc.

Un ordenador es parecido a un niño, en el sentido de que para que haga algo, primero debemos enseñarle a hacerlo. Así por ejemplo, si queremos que un niño aprenda a cruzar la calle él "solito", le daremos una serie de instrucciones:

**1º Si hay un semáforo, esperar a que se ilumine un hombrecito verde y cruzar rápidamente.**

**2º Si no hay un semáforo cerca, buscar un paso cebra y si está cerca hacer lo siguiente:**

**2.1. Ir al paso cebra.**

**2.2. Esperar a que no pase ningún vehículo**

**2.3. Cruzar.**

**3. Si no hay ningún paso cebra cerca, entonces hacer lo siguiente:**

**3.1. Mirar a la izquierda y a la derecha para ver si viene algún coche a alta velocidad o está muy cerca.**

### **3.2. Si no viene ningún coche cruzar rápidamente, en caso de que venga algún coche esperar e ir al paso 3.1.**

Si en lugar de dar al niño todas estas instrucciones le indico que para cruzar la calle lo único que debe hacer es ir de la acera donde está a la de enfrente, posiblemente el niño no sabría qué le queremos decir o simplemente resultaría peligroso.

Un programa se realiza de igual forma, se deben especificar un conjunto de instrucciones (o sentencias) simples, ordenadas en una secuencia que el ordenador debe cumplir. El ordenador, al igual que el niño, entiende solamente instrucciones simples; es decir, instrucciones con poco nivel de dificultad, no podemos dar una instrucción al ordenador del tipo "dime si el año 1993 es bisiesto", en lugar de eso debemos darle un conjunto de instrucciones que indiquen como se calcula un año bisiesto. A un ordenador se le pueden dar instrucciones en diversos "idiomas", a estos idiomas les denominamos lenguajes de programación. Así, las instrucciones que puedo dar a un ordenador utilizando el lenguaje de programación BASIC son distintas a las instrucciones que le puedo dar utilizando el lenguaje de programación C.

Los lenguajes de programación definen como se escriben las instrucciones que entiende el ordenador y el nivel de dificultad de las instrucciones (el nivel que entiende el ordenador) y que 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. Así pues en los siguientes capítulos aprenderemos:

- \* Con qué datos o información trabaja el ordenador.
- \* Cuales son las operaciones que puede realizar con los datos.
- \* Que tipo de instrucciones "entiende" un ordenador.

## **2. PASOS PREVIOS A LA PROGRAMACIÓN DE ORDENADORES. ALGORITMIA.**

Como todos sabemos realizar una película no es un trabajo fácil,

primero se debe escribir el guión, después buscar los actores que mejor se adapten a los distintos papeles, encontrar un buen director, rodar la película, cortar planos, repetir escenas que no han salido como quería el director, etc. También, para realizar un programa de ordenador lo primero que se debe hacer es un guión (qué datos serán los de entrada, como los procesará, cuales serán los datos de salida, etc), **el guión de un programa se conoce con el nombre de algoritmo**. Posteriormente se elige el lenguaje de programación que mejor se adapte al algoritmo, buscar un buen programador, escribir el programa, repetir partes que no hacen lo que quería el programador, etc. Como se puede observar, realizar un programa es una tarea laboriosa que se va haciendo más fácil a medida que aumenta la experiencia, es decir, para ser un buen programador se deben realizar programas, cuantos más mejor.

Si a un programador de BASIC le presentan un programa escrito en C, éste no sabrá qué hace el programa; sin embargo si a un programador de BASIC le presentan el algoritmo del programa C, éste sabrá perfectamente qué hace el programa. La algoritmia es una herramienta común a todos los lenguajes de programación; es decir, da igual el lenguaje de programación que utilicemos para escribir un programa, el algoritmo siempre será el mismo. Esta característica hace que todos los programadores del mundo prefieran estudiar más un algoritmo que un programa, ya que están seguros de que el algoritmo lo entenderán y el programa no.

¿Se imagina usted a un director de cine que haga primero la película y después llame a expertos para que le digan si la película es comercial, si el argumento está desfasado, si hay muchas películas parecidas en el mercado, si debe cambiar alguna escena, etc?. Lo lógico es que el director cuando tiene el guión llame al equipo de colaboradores, hagan los cambios oportunos y se rueda la película. En el primer caso el director se arriesga a que la película no sea aceptada por el público y tendrá que rodarla de nuevo adaptándose a las exigencias del público; en el segundo caso la película saldrá al mercado con garantías de éxito. Lo mismo ocurre cuando se escribe un programa, primero se realiza el algoritmo y sobre él se estudia si el programa es muy grande, si realizará las operaciones que estaban previstas, incluso con el algoritmo se puede ver si el programa funcionará correctamente. Si deseamos realizar un buen programa deberemos previamente haber realizado un buen algoritmo.

A pesar de las ventajas que aporta la algoritmia, hay muchos programadores que no la utilizan porque dicen que es más lento escribir el programa: "hay que hacer primero el algoritmo y después el programa y es

más rápido si sólo se hace el programa". Aunque esto sea cierto, si el lector desea ser un buen programador primeramente deberá hacer algoritmos y después programas. No obstante, si el lector es capaz de realizar un programa y mentalmente tener el esquema del algoritmo, si además cree que el programa no lo va a ver ningún otro programador, ni nunca se modificará, ni nunca se le olvidará y está seguro de que no existe otra forma de hacer mejor el programa, entonces no haga el algoritmo.

Los algoritmos se realizan con un conjunto de símbolos que iremos describiendo a lo largo del libro, para cada sentencia de programación suele haber un símbolo equivalente en algoritmia, no obstante existen una serie de condiciones que debe cumplir cualquier algoritmo:

**Finitud.** Todo algoritmo debe acabar tras un número finito de pasos. Esto quiere decir que el algoritmo debe finalizar en algún momento, no podemos hacer un algoritmo que no acabe nunca. Además cuantos menos pasos tenga el algoritmo mejor será éste. El proceso siguiente es un ejemplo de una combinación de pasos que nunca finalizaría:

1. **Hacer  $I=1$**
2. **Escribe el valor de  $I$ .**
3. **Incrementar el valor de  $I$  en una unidad.**
4. **Ir al paso 2.**

Además todo algoritmo debe tener un punto de inicio (primer paso realizado) y un punto final (último paso realizado).

**Definibilidad.** Cada paso del algoritmo está definido de modo preciso y sin ambigüedad. Cada paso se debe entender perfectamente. Por ejemplo:

#### **Escribir datos personales.**

Es un paso que se entiende pero no está definido, ya que no sabemos cuáles son los datos personales a los que se refiere, en su lugar se debería escribir:

#### **Escribir nombre, dirección, teléfono.**

**Entrada.** Normalmente todos los algoritmos tienen una entrada. La entrada suele ser información que o bien se suministra desde fuera del algoritmo, o bien se expresa dentro del algoritmo con unas condiciones iniciales. Las entradas están representadas por un paso y

una información. El paso está representado en el algoritmo y dice qué acción se lleva a cabo para realizar la entrada; la información específica qué tipo de dato se utilizará en la entrada.

**1. Leer Nombre del usuario del fichero de datos.**

**2. Esperar a que el usuario introduzca por el teclado su edad.**

En el primer paso se indica que el nombre está almacenado en algún lugar y se ha de ir a buscarlo. El tipo de datos es carácter. En el segundo paso se indica que el usuario debe introducir por teclado su edad y el ordenador debe esperar a que el usuario realice esa acción. El tipo de datos es numérico.

**Salida.** Todo algoritmo debe tener una o varias salidas. Las salidas son estados finales que guardan relación con la entrada. Por ejemplo la salida de un algoritmo que realice el factorial de un número está relacionado con la entrada, en este caso mediante una fórmula.

**Efectividad.** Las operaciones y procesos que realiza un algoritmo deben ser bastante básicas, así se podrán hacer de modo fácil y rápido.

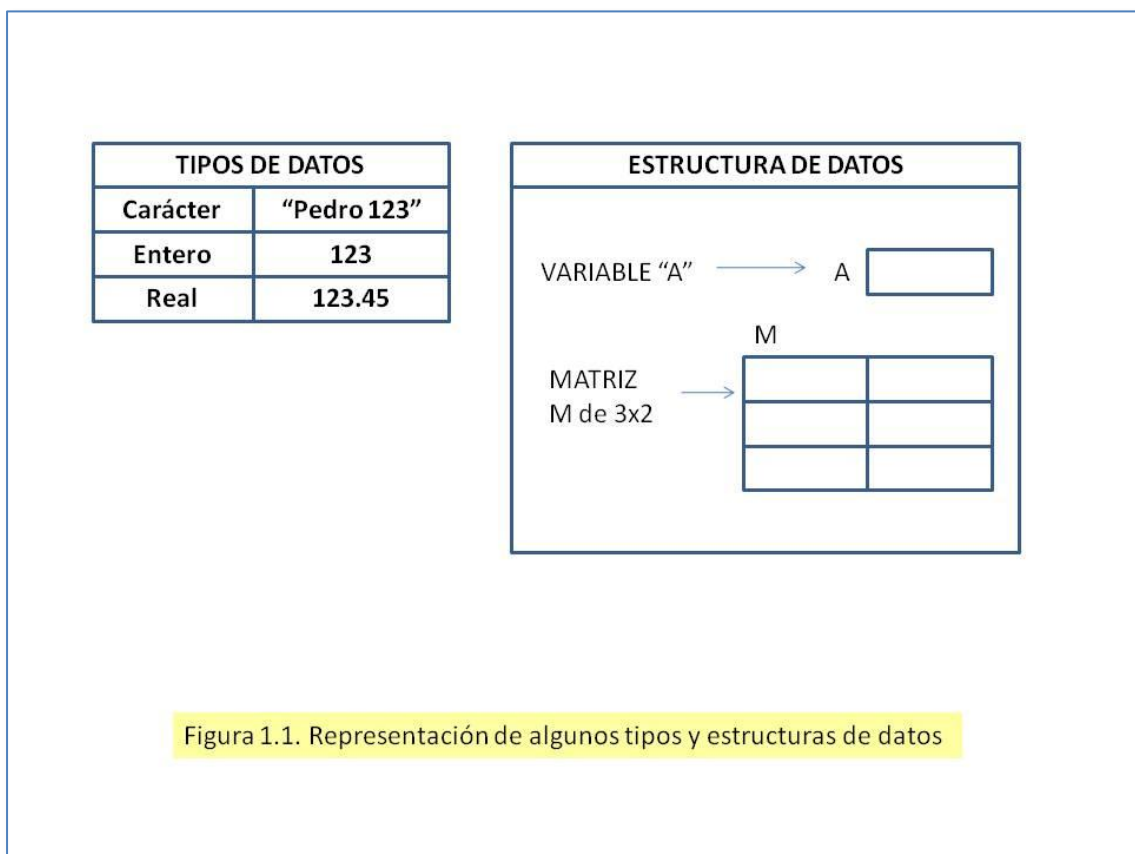
### 3. ELEMENTOS DE LOS LENGUAJES DE PROGRAMACIÓN Y DE LOS ALGORITMOS.

#### 3.1. Elementos de los lenguajes de programación.

Cualquier lenguaje de programación trabaja básicamente con tres elementos: información, operaciones e instrucciones. Se analizarán cada uno de estos elementos por separado y después se describirán las relaciones con el lenguaje de programación C.

##### 3.1.1. La información.

Esta se refiere a los datos con los cuales trabajarán los programas. Los datos suelen ser de dos tipos: numéricos y alfanuméricos (ó caracteres). Los datos se pueden agrupar formando estructuras, las estructuras pueden ser muy simples como las constantes (28) y las variables (x) o muy complejas como las matrices y registros. La **figura nº 1.1** representa distintos tipos y estructuras de datos.





### 3.1.2. Las operaciones.

Se refieren a las operaciones que el ordenador es capaz de realizar con los diversos tipos de datos. Por ejemplo, la suma de dos números se puede realizar por un ordenador directamente; pero para realizar la suma de dos matrices habría que hacer un programa (existen algunos lenguajes de programación que permiten directamente realizar sumas de matrices).

Las operaciones pueden ser realizadas mediante operadores y funciones predefinidas que se aplican a un tipo de datos determinado. Por ejemplo, para los datos de tipo numérico se permiten los operadores suma, resta, división y producto, cuya operación se representaría:

```
a=5*3  
c=6*b
```

Las funciones que dispone el lenguaje de programación para realizar operaciones se denominan predefinidas, ya que vienen definidas de antemano. Su utilización es como sigue:

```
a=sqrt(10)  
b=log(b)
```

Normalmente para definir las operaciones con operadores y funciones se utiliza el término **expresiones**, que pueden ser operaciones con operadores, con funciones o con ambas, un ejemplo sería:

```
a=b*log(b)+sqrt(i)
```

Evidentemente existen expresiones tanto numéricas como de caracteres y estas expresiones pueden trabajar con distintas estructuras de datos.

### 3.1.3. Las instrucciones.

Las instrucciones (también denominadas sentencias) son el conjunto de órdenes que se le pueden dar al ordenador. En función del tipo de orden que indique al ordenador, las instrucciones se clasifican en los siguientes tipos:

#### 3.1.3.1 Instrucciones de entrada y salida (E/ S).

También denominadas de lectura/escritura (L/S) ó input/output (I/O). Estas instrucciones se encargan de la información de entrada y de salida; es decir de la información que necesita un programa para realizar su tarea y de la información que genera el programa. Son instrucciones del tipo: Almacena en la variable "nom" el nombre del usuario ó escribe el contenido de la variable "nom". Normalmente la entrada y la salida se realiza desde un fichero ya creado (información almacenada en un dispositivo magnético) o es el usuario quien suministra o recibe la información a través del teclado y pantalla del ordenador. No obstante la entrada y salida de información en el ordenador puede realizarse de numerosas formas, tantas como dispositivos de entrada y salida se puedan conectar al ordenador. A continuación se describen los dispositivos de entrada y salida más comúnmente utilizados por los ordenadores. Para cada uno de ellos existirían instrucciones de entrada y salida.

### **Dispositivos de entrada.**

Son los utilizados para introducir la información en un ordenador. El dispositivo de entrada más utilizado es el **teclado** (ver figura 1.3.), éste es similar a una máquina de escribir, sólo que en lugar de almacenar la información en un papel, el teclado envía la información a los dispositivos de procesamiento.

La misión básica de un dispositivo de entrada es suministrar información al ordenador. Normalmente el ordenador tiene predefinido un sistema de entrada (puertos), estos puertos están preparados para que se conecte a ellos cualquier dispositivo de entrada que pueda enviar información. Los dispositivos más comunes que se pueden conectar son:

**RATÓN:** Es un dispositivo que se utiliza para desplazar un apuntador en la pantalla en entornos gráficos. El desplazamiento se realiza moviendo el dispositivo a través de una superficie.

**SCANNERS:** Su funcionamiento es similar a una

fotocopiadora, la principal diferencia radica en que la fotocopia en lugar de almacenarse en un papel se almacena en el dispositivo de procesamiento.

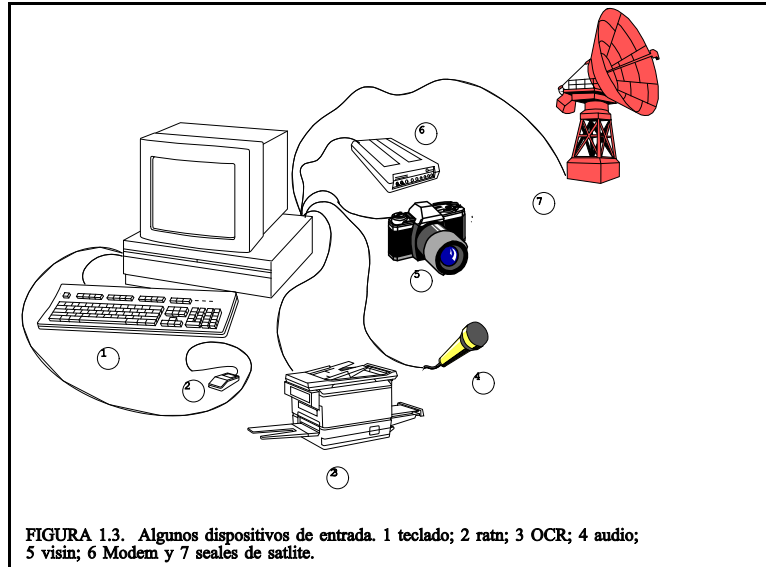


FIGURA 1.3. Algunos dispositivos de entrada. 1 teclado; 2 ratn; 3 OCR; 4 audio; 5 visin; 6 Modem y 7 seales de satlite.

**CÁMARAS/ VIDEO:** Son cámaras y videos normales, la única diferencia radica en que las imágenes en lugar de almacenarse en una película o cinta magnética se almacenan directamente en las unidades de procesamiento.

**SISTEMAS OCR:** Son sistemas que leen documentos y los envían a la unidad de procesamiento. La diferencia con los elementos anteriores es que los sistemas OCR reconocen los caracteres que introducen, por tanto los documentos se almacenan como textos que se pueden modificar.

**MODEMS:** Son sistemas que convierten la señal enviada por una línea telefónica a una señal inteligible por el ordenador. Así pues, el modem almacena la información que llega por una línea telefónica.

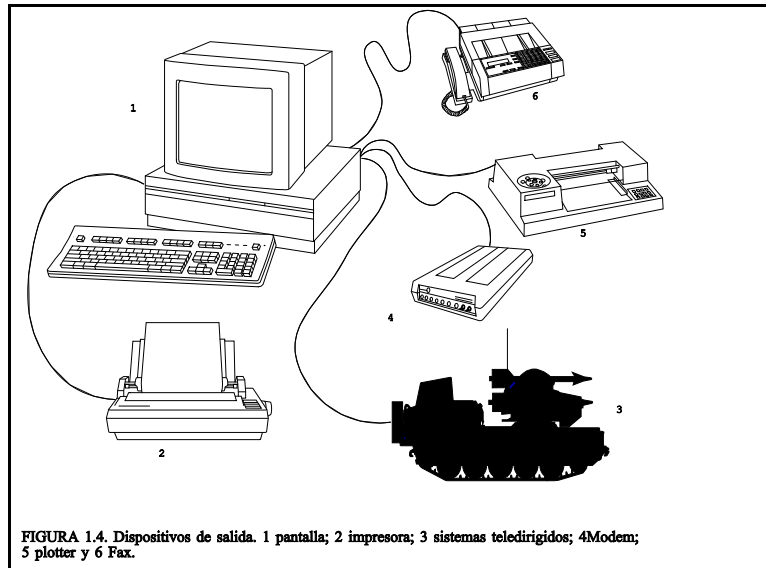
**SISTEMAS AUDIO:** Permiten almacenar en los dispositivos de procesamiento señales de audio, normalmente son enviadas a través de un micrófono o cualquier sistema de reproducción de audio.

Existen muchos dispositivos conectables al ordenador (que envían información a los dispositivos de procesamiento). Realmente se puede conectar cualquier dispositivo que generará alguna señal o información (osciloscopios, medidores de presión, alarmas, etc.).

## Dispositivos de salida.

Operan con la información en sentido inverso al de los dispositivos de entrada; es decir, reciben la información del ordenador. Normalmente estos dispositivos muestran el estado de la información que contiene el ordenador.

El dispositivo de salida más popular es la **pantalla** (ver **figura 1.4.**), la cual permite visualizar tanto la información enviada por los dispositivos de entrada, como la información generada por la ejecución de un programa.



El ordenador suele tener puertos de salida, a los cuales se pueden conectar distintos dispositivos de salida, entre los cuales destacamos:

**IMPRESORAS:** Este dispositivo imprime la información enviada desde el dispositivo de procesamiento. La impresión se suele hacer en papel, pudiendo escribir caracteres alfanuméricos, imágenes, fórmulas, dibujos, etc.

**PLOTTERS:** son dispositivos de salida especializados en trazar planos o cualquier dibujo similar.

**MODEM:** es un dispositivo de salida que se encarga de enviar la información del dispositivo de procesamiento a través de la línea telefónica. Este dispositivo también actúa como dispositivo de entrada, por tanto es un dispositivo de entrada/ salida.

En general se puede conectar al ordenador cualquier dispositivo que para funcionar necesite un control mediante información. Podría ser

un robot, una escalera mecánica, un reloj, etc.

### 3.1.3.2. Instrucciones de control.

Son instrucciones que sirven para dirigir la ejecución de un programa. Normalmente un programa está compuesto por un conjunto de instrucciones que se ejecutan una tras otra. Las instrucciones de control permiten cambiar la secuencia de ejecución. Son instrucciones del tipo: Si ocurre tal condición ejecuta determinadas instrucciones, de lo contrario ejecuta otras; ó cuando llegues a esta instrucción vete a la primera. La **figura nº 1.5** muestra el control de la ejecución para este tipo de sentencias.

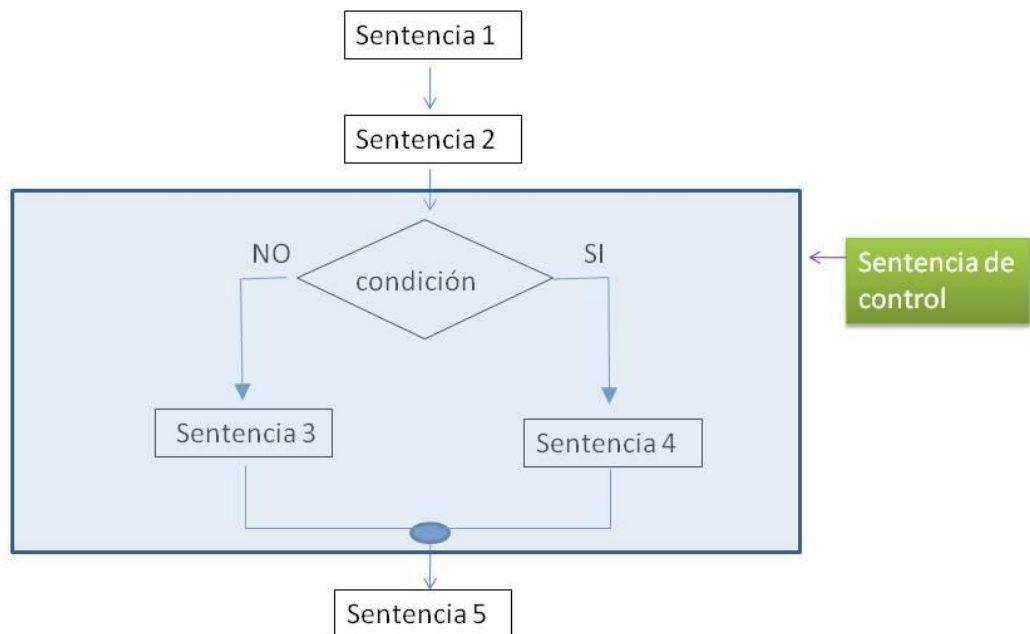
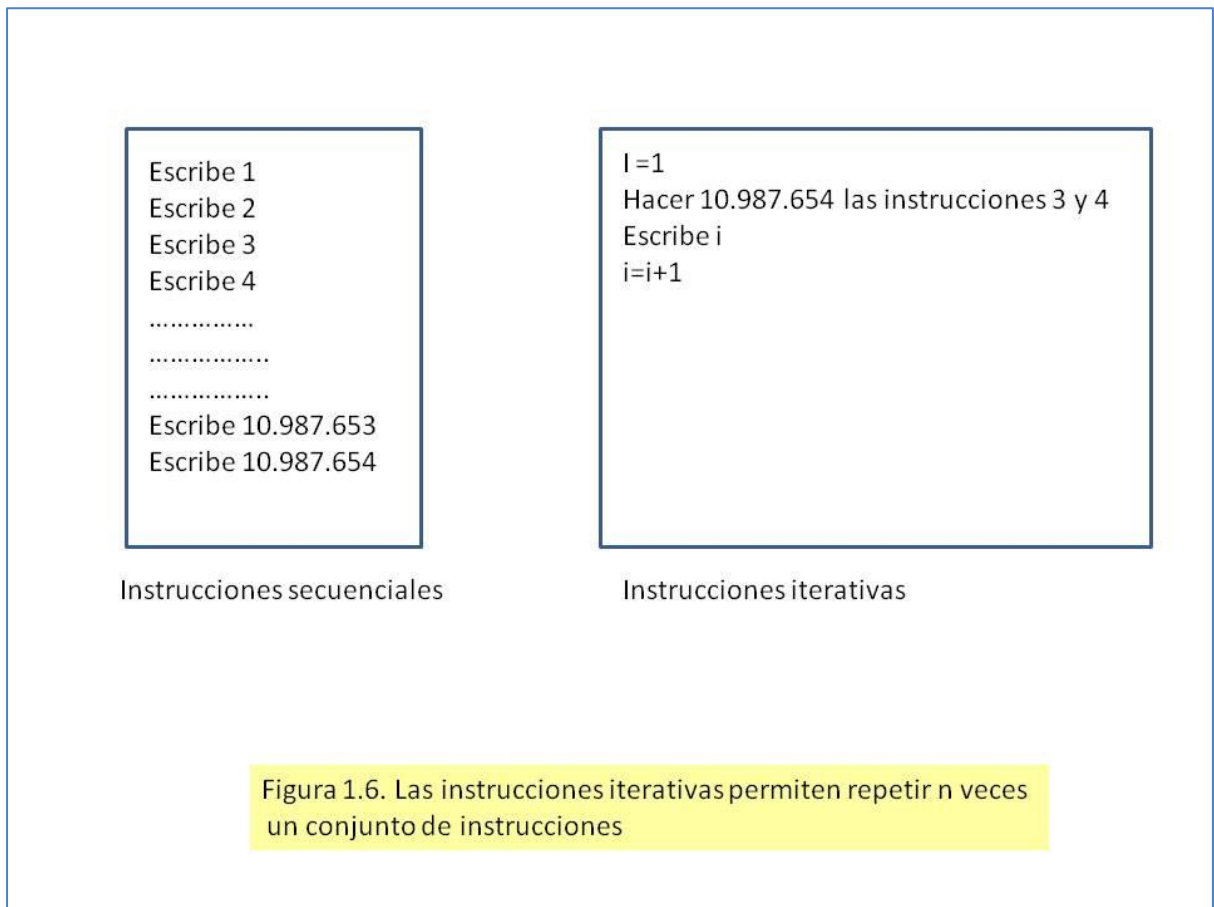


Figura 1.5 Ejemplo de sentencia de control. Si la condición es cierta se ejecuta la sentencia 4 en caso contrario la sentencia 3.

### 3.1.3.3. Instrucciones iterativas.

Son instrucciones que permiten repetir un número determinado de veces un conjunto de instrucciones. Por ejemplo, si deseo realizar un programa que escriba los números enteros del 1 al 10.987.654 se puede realizar de dos formas: escribiendo diez millones novecientos ochenta y siete mil seiscientos cincuenta y cuatro instrucciones de salida o un sencillo bucle con cuatro sentencias. La **figura n° 1.6** muestra la situación citada.



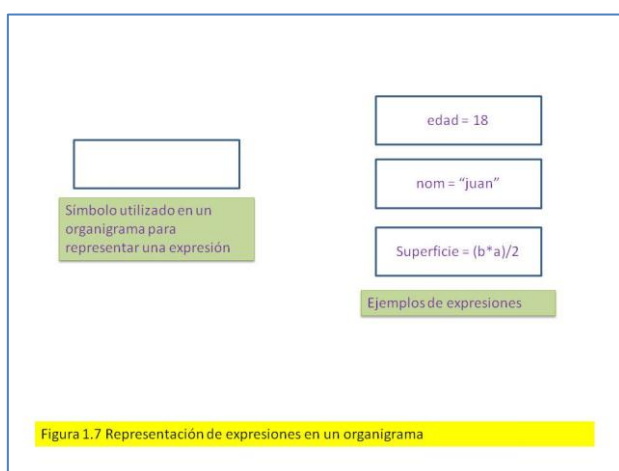
### 3.2. Elementos de los algoritmos.

Los algoritmos están muy ligados a los lenguajes de programación, por tanto es lógico suponer que los elementos de los algoritmos son similares a los elementos de los lenguajes de programación. Esto es completamente cierto, para cada elemento de los lenguajes de programación existe un elemento equivalente en algoritmia, la principal diferencia es su representación. Mientras que para un lenguaje de programación los elementos se representan mediante una sentencia, los elementos de los algoritmos se pueden representar gráficamente. La representación gráfica de un algoritmo se denomina **organigrama**. Un algoritmo indica una secuencia de pasos (parecido a una receta de cocina), la representación del mismo se puede hacer textualmente o gráficamente. En este libro utilizaremos únicamente la representación gráfica; así pues cuando se utilice la palabra organigrama nos estaremos refiriendo a la representación de los algoritmos. Además en los algoritmos también se indica el flujo de las órdenes; es decir la secuencia de ejecución, por tanto en los organigramas también se debe presentar este flujo. A continuación se describen los elementos de los organigramas:

#### 3.2.1. Información.

La información se representa mediante un conjunto de celdas y en cada celda se tiene un dato elemental. A lo largo del curso se describirán y representarán los distintos tipos y estructuras de datos.

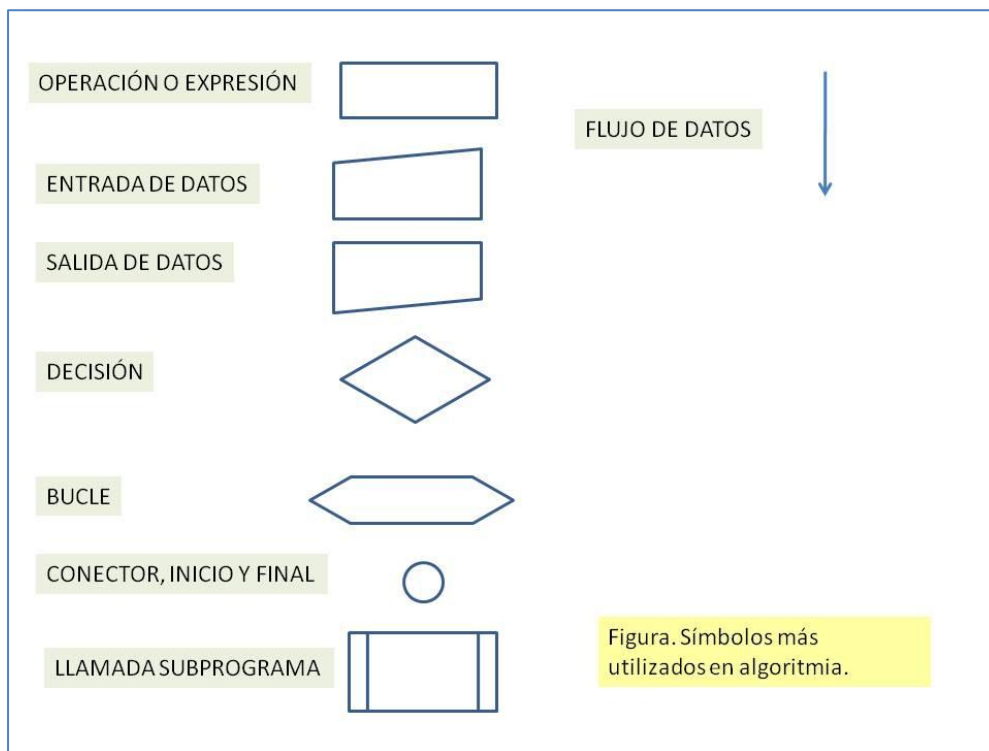
#### 3.3.2. Expresiones.



En su representación no se distingue el tipo de expresión. Todas las expresiones se representan por un rectángulo y dentro del rectángulo se indica la expresión concreta. La **figura nº 1.7** representa el símbolo de expresión, así como varios ejemplos de expresiones.

### 3.3.3. Instrucciones.

En algoritmia cada tipo de instrucción tiene una representación gráfica distinta. Como siempre existen dos elementos: el símbolo gráfico que indica qué tipo de instrucción es y el texto que está dentro del símbolo, que indica la instrucción concreta. A continuación se describen brevemente los símbolos para cada tipo de instrucción más comúnmente utilizados. Los diferentes símbolos utilizados para realizar organigramas se describirán a medida que se analicen las distintas instrucciones que representan.



### 3.3.4. Flujo de instrucciones.

El flujo de instrucciones en los organigramas se representa mediante flechas que indican la secuencia de ejecución. Es muy importante saber qué función se ejecuta primero y cual después, ya que de esta secuencia depende el correcto funcionamiento del programa.

En el ejemplo del niño que intenta cruzar la calle es muy importante que se guarde el orden indicado (primero mirar y después cruzar); ya se imaginará el lector qué ocurriría si se cambiara el orden (primero cruzar y después mirar).



El flujo secuencial se representa por la unión de los símbolos gráficos a través de una flecha, no obstante mediante las instrucciones de control esta secuencia se puede cambiar. La **figura nº 1.10** muestra las dos situaciones.

