

## ALGORITMOS Y PROGRAMAS

### CONTENIDO

- 1.1. Los sistemas de procesamiento de la información.
- 1.2. Concepto de algoritmo.
- 1.3. Los lenguajes de programación.
- 1.4. Datos, tipos de datos y operaciones primitivas.
- 1.5. Constantes y variables.
- 1.6. Expresiones.
- 1.7. Funciones internas.
- 1.8. La operación de asignación.
- 1.9. Entrada y salida de información.

### ACTIVIDADES DE PROGRAMACIÓN RESUELTAS. EJERCICIOS.

---

La principal razón para que las personas aprendan lenguajes y técnicas de programación es utilizar la computadora como una herramienta para resolver problemas. La resolución de un problema exige al menos los siguientes pasos:

1. Definición o análisis del problema.
2. Diseño del algoritmo.
3. Transformación del algoritmo en un programa.
4. Ejecución y validación del programa.

Uno de los objetivos fundamentales de este libro es el *aprendizaje* y *diseño* de algoritmos. Este capítulo introduce al lector en el concepto de algoritmo y de programa, así como las herramientas que permiten <<dialogar>> al usuario con la máquina: los lenguajes de programación.

Un algoritmo es un método para resolver un problema. Aunque la popularización del término ha llegado con el advenimiento de la era informática, algoritmo proviene de Mohammed al-Khowarizmí,

matemático persa que vivió durante el siglo IX y alcanzó gran reputación por el enunciado de las reglas paso a paso para sumar, restar, multiplicar y dividir números decimales; la traducción al latín del apellido en la palabra *algorismus* derivó posteriormente en algoritmo. Euclides, el gran matemático griego (del siglo IV antes de Cristo), que inventó un método para encontrar el máximo común divisor de dos números, se considera con Al-Khowarizmi el otro gran padre de la algoritmia (ciencia que trata de los algoritmos).

El resto del capítulo trata de los datos y las operaciones elementales necesarias para el diseño del algoritmo. El profesor Niklaus Wirth –inventor de Pascal, Modula-2 y Oberon– tituló uno de sus más famosos libros, *Algoritmos + Estructuras de datos = Programas*, significándonos que sólo se puede llegar a realizar un buen programa con el diseño de un algoritmo y una correcta estructura de datos. Esta ecuación será una de las hipótesis fundamentales consideradas en esta obra.

---

## 1.1. LOS SISTEMAS DE PROCESAMIENTO DE LA INFORMACIÓN.

Una definición antigua de computadora es: <<una máquina o aparato electrónico capaz de ejecutar operaciones repetitivas muy complejas a altas velocidades>>. Ahora bien, ésta definición no describe las modernas computadoras. Éstas son más que una máquina de ejecutar operaciones aritméticas. De hecho, los términos procesador de *datos* y *sistemas de procesamiento (tratamiento) de la información* se utilizan con frecuencia en lugar de computadora (*ordenador*, en la jerga informática usual en España).

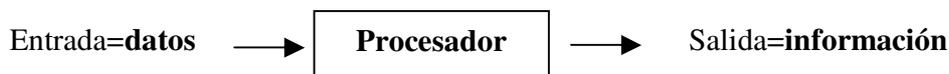
En el uso diario, datos e información son esencialmente sinónimos. Sin embargo, los informáticos suelen hacer una diferencia: datos se refiere a la representación de un hecho, concepto o entidad real (los datos pueden tomar diferentes formas: por ejemplo, palabras escritas o habladas, números y dibujos); *información* implica datos procesados y organizados.

Un sistema en general se define como conjunto de componentes conectados e interactivos, que tienen un propósito y una unidad total. **Sistema de procesamiento de información** es un sistema que transforma datos brutos en *información organizada, significativa y útil*.

La Figura 1.1 muestra los tres componentes de un sistema de proceso de la información: *entrada, salida y procesador*. El *procesador*, que puede ser bastante complicado, se representa por una simple caja y puede aceptar datos llamados entrada, y esta entrada se transforma entonces para producir una información denominada *salida o resultados*.

Basados en este esquema, muchos dispositivos u organismos pueden ser considerados sistemas de procesamiento de la información. Un termostato que controla la temperatura de un edificio es un sistema de procesamiento de la información. La entrada es la temperatura media y la salida es una señal que controla la caldera del aire acondicionado. El corazón de un animal o un ser humano es un sistema complejo de procesamiento de la información.

El conjunto de instrucciones que especifican la secuencia de operaciones a realizar, en orden, para resolver un sistema específico o clase de problemas, se denomina algoritmo. En otras palabras, un **algoritmo** es una *fórmula* para la resolución de un problema.



### Figura 1.1. Sistema de proceso de la información

Para realizar un proceso se le debe suministrar al procesador un algoritmo adecuado. Por ejemplo, al cocinero debe dársele una receta, al pianista la partitura y así sucesivamente, considerando al cocinero y al pianista como procesadores.

Cuando el procesador es una computadora, el algoritmo ha de expresarse de una forma que recibe el nombre de programa. Un programa se escribe en un lenguaje de programación y a la actividad de expresar un algoritmo en forma de programa se le denomina programación. Cada paso en algoritmo está expresado por medio de una instrucción en el programa. Por consiguiente, un programa consta de una secuencia de instrucciones, cada una de las cuales especifica las operaciones que debe realizar la computadora.

Existen dos conceptos importantes a considerar en los sistemas de procesamiento de la información: hardware y software. **Hardware** es el conjunto de componentes físicos de una computadora (Figura 1.2) –equipo físico– y **software** es el conjunto de programas que controlan el funcionamiento de una computadora –equipo lógico–.

EL hardware de una computadora se compone de:

1. La Unidad Central de Proceso, UCP (Central Processing Unit, CPU). La UCP es el conjunto de circuitos electrónicos capaces de ejecutar algunos cálculos sencillos como suma o multiplicación de números. La potencia de una computadora depende completamente de la velocidad y fiabilidad de la UCP.
2. Memoria central. La información procesada por la UCP se almacena normalmente en la memoria central hasta que se terminan los cálculos. Los programas de computadora se almacenan también en la memoria central.
3. Dispositivos de almacenamiento secundario (memoria auxiliar). Diferentes dispositivos, tales como discos y cintas magnéticas, se usan para almacenar grandes cantidades de información. Para ser procesados por la UCP, los datos se almacenan en dispositivos de almacenamiento auxiliar y luego tienen que llevarse a la memoria central.
4. Periférico o dispositivos de entrada/salida (E/S). Estos dispositivos permiten al usuario comunicarse con la computadora. Un sistema de computadoras puede tener diferentes dispositivos periféricos conectados a ella.

En la práctica, una instalación grande de computadora puede tener diferentes UCP, cada una con su propia memoria central compartida, una variedad de dispositivos de almacenamiento secundario y periféricos localizados en diferentes partes de un mismo edificio o diferentes edificios e incluso diferentes ciudades o países.

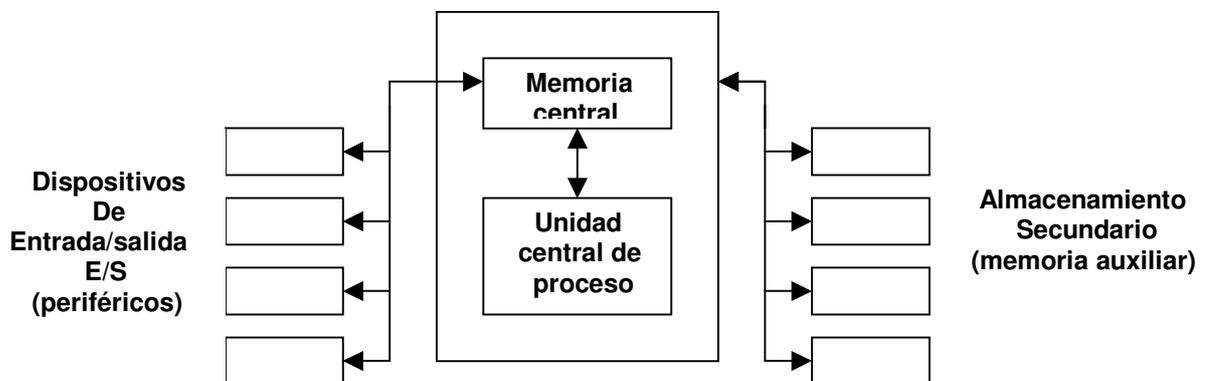
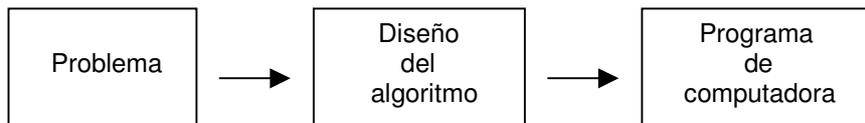


Figura 1.2. Diagrama esquemático de una computadora (hardware)

## 1.2. CONCEPTO DE ALGORITMO

El objetivo fundamental de esta texto es enseñar a resolver problemas mediante una computadora. Un programador de computadora es antes que nada una persona que resuelve problemas, por lo que para llegar a ser un programador eficaz se necesita aprender a resolver problemas de un modo riguroso y sistemático. A lo largo de todo este libro nos referimos a la metodología necesaria para resolver problemas mediante programas, al concepto se denomina **metodología de la programación**. El eje central de esta metodología es el concepto, ya tratado, de algoritmo.

La resolución de un problema exige el diseño de un algoritmo que resuelva el problema propuesto.



**Figura 1.3.** Resolución de un problema

Los pasos para la resolución de un problema son:

1. Diseño del algoritmo que describe la secuencia ordenada de pasos –sin ambigüedades– que conducen a la solución de un problema dado. (Análisis del programa y desarrollo del algoritmo.)
2. Expresar el algoritmo como un programa en un lenguaje de programación adecuado. (Fase de codificación.)
3. Ejecución y validación del programa por la computadora.

Para llegar a la realización de un programa es necesario el diseño previo de un algoritmo, de modo que sin algoritmo no puede existir un programa.

Los algoritmos son independientes tanto del lenguaje de programación en que se expresan como de la computadora que los ejecuta. En cada problema el algoritmo se puede expresar en un lenguaje diferente de programación y ejecutarse en una computadora distinta; sin embargo, el algoritmo será siempre el mismo. Así, por ejemplo, en una analogía con la vida diaria, una receta de un plato de cocina se puede expresar en español, inglés o francés, pero cualquiera que sea el lenguaje, los pasos para la elaboración del plato se realizarán sin importar el idioma del cocinero.

En la ciencia de la computación y en la programación, los algoritmos son mas importantes que los lenguajes de programación o las computadoras. Un lenguaje de programación es tan solo un medio para expresar un algoritmo y una computadora es sólo un procesador para ejecutarlo. Tanto el lenguaje de programación como la computadora son los medios para obtener un fin: conseguir que el algoritmo se ejecute y se efectúe el proceso correspondiente.

Dada la importancia del algoritmo en la ciencia de la computación, un aspecto muy importante será el diseño de algoritmos. A la enseñanza y práctica de esta tarea se dedica gran parte de este libro.

El diseño de la mayoría de los algoritmos requiere creatividad y conocimientos profundos de la técnica de la programación. En esencia, la solución de un problema se puede expresar mediante un algoritmo.

### 1.2.1. CARACTERÍSTICAS DE LOS ALGORITMOS

Las características fundamentales que debe cumplir todo algoritmo son:

- Un algoritmo debe ser preciso e indicar el orden de realización de cada paso.
- Un algoritmo debe estar definido. Si se sigue un algoritmo dos veces, se debe obtener el mismo resultado cada vez.
- Un algoritmo debe ser finito. Si se sigue un algoritmo, se debe terminar en algún momento; osea, debe tener un número finito de pasos.

La definición de un algoritmo debe describir tres partes: Entrada, proceso y salida. En el algoritmo de receta de cocina citado anteriormente se tendrá:

Entrada: Ingredientes y utensilios empleados  
Proceso: Elaboración de la receta en la cocina.  
Salida: Terminación del plato (por ejemplo, cordero).

### Ejemplo 1.1

Un cliente ejecuta un pedido a una fábrica. La fábrica examina en su banco de datos la ficha del cliente, si el cliente es solvente entonces la empresa acepta el pedido; en caso contrario, rechazará el pedido. Redactar el algoritmo correspondiente.

Los pasos del algoritmo son:

1. Inicio.
2. Leer el pedido.
3. Examinar la ficha del cliente
4. Si el cliente es solvente, aceptar pedido; en caso contrario, rechazar pedido
5. Fin

### Ejemplo 1.2

Se desea diseñar un algoritmo para saber si un número es primo o no.

Un número es primo si solo puede dividirse por si mismo y por la unidad (es decir no tiene más divisores que él mismo y la unidad). Por ejemplo, 9, 8, 6, 4, 12, 16, 20, etc, no son primos ya que son divisibles por números distintos en a ellos y a la unidad. Así, 9 es divisible por 3, 8 lo es por 2, etc.

El algoritmo de resolución del problema pasa por dividir sucesivamente el número por 2, 3, 4, . . ., etc.

1. Inicio.
2. Poner X igual a 2 ( $X = 2$ , X, variable que representa a los divisores del número que se busca N).
3. Dividir N por X ( $N / X$ ).
4. Si el resultado de  $N / X$  es entero, entonces N no es un número primo y bifurcar al punto 7; en caso contrario continuar el proceso.
5. Suma 1 a X ( $X \cdot X + 1$ ).
6. Si X es igual a N, entonces N es un número primo en caso contrario bifurcar al punto 3.
7. Fin.

---

1 Esta condición puede ser sustituida por  $X = n \text{ div } 2$  (donde div es el operador división entera). Véase tabla 1.1 página 19

Por ejemplo, si N es 131, los pasos anteriores serian:

1. Inicio.
2.  $X = 2$ .
- 3 y 4.  $131 / X$ . Como el resultado no es un entero, se continua el proceso.
5.  $X \cdot 2 + 1$ , luego  $X = 3$ .
6. Como X no es 131, se bifurca al punto 3.
- 3 y 4.  $131 / X$  resultado no es un entero.
5.  $X \cdot 3 + 1$ ,  $X = 4$ .
6. Como X no es 131, se bifurca al punto 3.
- 3 y 4.  $131 / X \dots$ , etc.
7. Fin.

### Ejemplo 1.3

Realizar la suma de todos los números pares entre 2 y 1000.

El problema consiste en sumar  $2 + 4 + 6 + 8 + \dots + 1000$ .

Utilizaremos las palabras SUMA Y NUMERO (variables, serán denominadas mas tarde) para representar las sumas sucesivas  $(2 + 4)$ ,  $(2 + 4 + 6)$ ,  $(2 + 4 + 6 + 8)$ , etc.

La solución se puede escribir con el siguiente algoritmo:

1. Inicio.
2. Establecer SUMA a 0.
3. Establecer NUMERO a 2.
4. Sumar NUMERO A SUMA .El resultado será el nuevo valor de la suma (SUMA).
5. Incrementar NUMERO 2 unidades.
6. El NUMERO  $\leq 1000$  bifurca al paso 4 ; en caso contrario, escribir el ultimo valor de SUMA y terminar el proceso.
7. Fin.

## 1.3. LOS LENGUAJES DE PROGRAMACIÓN

Como se ha visto en el apartado anterior, para que un procesador realice un proceso se le debe suministrar en primer lugar un algoritmo adecuado. El procesador debe ser capaz de interpretar el algoritmo, lo que significa:

- comprender las instrucciones de cada paso,
- realizar las operaciones correspondientes.

Cuando el procesador es un computadora, el algoritmo se ha de expresar en un formato que se denomina programa. Un programa se escribe en un lenguaje de programación y las operaciones que conducen a expresar en un algoritmo en forma de programa se llaman programación. Así pues, los lenguajes utilizados para escribir programas de computadoras son los lenguajes de programación y programadores son los escritores y diseñadores de programas.

Los principales tipos de lenguajes utilizados en la actualidad son tres:

- lenguaje maquina,
- lenguaje de bajo nivel ( ensamblador),
- lenguajes de alto nivel.