

1 Conceptos Generales

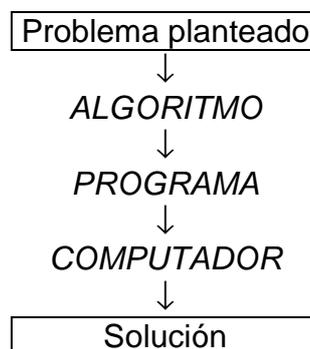
1.1 Definición de Informática

Una posible definición de informática podría ser:

El conjunto de conocimientos científicos y técnicos que hacen posible la resolución de forma automática de problemas por medio de las computadoras.

La informática es, en definitiva, una rama del saber que abarca todos los aspectos del diseño y uso de las computadoras y como consecuencia, debería cubrir todos aquellos problemas con los que nos encontramos a la hora de realizar una tarea con una máquina de forma automática.

La resolución de un problema mediante un ordenador conlleva, a grandes rasgos, la realización de los siguientes pasos:



La informática abarcará el estudio de todas aquellas cuestiones que aparecen en la realización de cada uno de esos pasos. Vamos a ver grandes rasgos cuáles son las principales.

1.2 Concepto de Algoritmo

En un sentido general y no necesariamente informático, un **algoritmo** se puede definir como una serie de operaciones elementales, detalladas, no ambiguas y ordenadas que ejecutadas una a una conducen a la resolución de un cierto problema. Cada una de esas operaciones elementales puede ser considerada una instrucción.

Ejemplo. *Supongamos que el problema consiste en obtener las soluciones de una cierta ecuación de segundo grado:*

$$ax^2 + bx + c = 0$$

Un posible algoritmo para resolver ese problema es el siguiente:

Paso 1. Elevar al cuadrado b.

Paso 2. Realizar el producto $4 \cdot a \cdot c$.

Paso 3. Restar las cantidades obtenidas en los pasos 1 y 2.

Paso 4. Calcular la raíz cuadrada del resultado del paso 3.

Paso 5. Realizar el producto $2 \cdot a$.

Paso 6. Restar b del resultado del paso 4.

*Paso 7. Dividir el resultado del paso 6 por la cantidad obtenida en el paso 5.
Esa cantidad es la primera raíz.*

Paso 8. Sumar el resultado del paso 4 con b y cambiar el signo al resultado.

*Paso 9. Dividir el resultado del paso 8 por la cantidad obtenida en el paso 5.
Esa cantidad es la segunda raíz.*

A la hora de plantear el algoritmo para la resolución de un determinado problema, se contemplan en informática dos campos de estudio teórico importantes:

- **Computabilidad.** Diremos que un problema es computable si admite solución algorítmica, es decir, si se puede establecer un algoritmo para su resolución. La teoría de la computabilidad intenta establecer que problemas son computables, o sea, cuáles son abordables mediante un ordenador.
- **Complejidad.** En los problemas computables es interesante estimar el orden de magnitud de los recursos computacionales que requieren los distintos algoritmos que puedan resolverlos. El estudio de la complejidad se encarga de estimar este orden de magnitud. Los recursos computacionales más importantes contemplados a la hora de procesar un algoritmo son: tiempo de procesamiento y requisitos de dispositivos (memoria, disco, etc.). Un problema puede ser teóricamente computable, pero con un algoritmo de una complejidad tan alta que sea irrealizable en una máquina actual.

El estudio de este paso es vital para la eficaz resolución de un determinado problema: plantear un buen algoritmo mejorará el rendimiento del sistema.

1.3 Concepto de Programa

Un **programa** es la plasmación de un algoritmo en una secuencia de sentencias o instrucciones pertenecientes a un determinado lenguaje de programación, susceptibles de ser eventualmente interpretadas y ejecutadas por el ordenador. Se trata, pues, de la realización práctica de un algoritmo en un ordenador.

Podemos definir **lenguaje de programación**, como el conjunto de símbolos y reglas que conforman un programa, y que nos permiten comunicarnos con la máquina.

De acuerdo a lo cercano o lejano que esté del lenguaje de la máquina, podemos establecer la siguiente clasificación en los lenguajes de programación:

- **Lenguaje máquina.** Está formado por las instrucciones que es capaz de ejecutar directamente la máquina, estando estas construidas mediante cadenas de 0s y 1s. Es el lenguaje más cercano a la máquina, y, por lo tanto, más difícil de entender y de programar para el ser humano, ya que, no solo el lenguaje está lejos del uso normalmente para comunicarnos, sino que, además, el programador necesita conocer la arquitectura física del computador con cierto detalle. Ej. de instrucción en *lenguaje máquina* sería: 000100111001000011
- **Lenguaje ensamblador.** En éste, cada instrucción máquina es traducida al lenguaje humano, usando, para ello, una notación nemotécnica que resume en una pequeña cadena de letras, la operación que realiza la instrucción máquina asociada. Ej. de instrucción en *lenguaje ensamblador* sería: SUM A,B (operación: sumar A y B).

Para que un programa escrito en lenguaje ensamblador pueda ser ejecutado por la máquina, necesita ser traducido al *lenguaje máquina* (único que entiende el ordenador). El programa que se encarga de esta tarea se denomina **ensamblador**.

- **Lenguaje de alto nivel.** El *lenguaje ensamblador* está escrito usando el alfabeto humano, pero sus reglas están lejos de ser la que usamos normalmente para comunicarnos entre nosotros; sigue siendo, por lo tanto, un lenguaje más cercano a la máquina, que al programador. Los *lenguajes de alto nivel*, sin embargo, son lenguajes muchos más próximos al modelo de lenguaje humano, que al de la máquina. Las instrucciones tienen un mayor sentido y su significado es más intuitivo, siendo, por lo tanto, más sencilla la realización de programas para el programador. En general, cada instrucción no se corresponde, como en el *lenguaje ensamblador*, directamente con una del *lenguaje máquina*, sino con un conjunto de ellas. Estos lenguajes, a diferencia de los dos anteriores, son independientes de la máquina, es decir, programas escritos para un ordenador pueden ser, más o menos, fácilmente transportados a otro ordenador. Ej. de instrucción en *lenguaje de alto nivel* sería: $a=3+6$. Ejemplos de lenguaje de alto nivel son: C, Pascal, Fortran, etc.

De acuerdo a cómo se haga la traducción a *lenguaje máquina*, podemos distinguir dos tipos de *lenguajes de alto nivel*:

- **Compilados.** El programa es traducido al lenguaje de la computadora. El programa traductor se denomina **compilador**, y su misión es analizar el *programa fuente* (programa escrito en el lenguaje de alto nivel correspondiente) y generar el *programa objeto*, programa similar al generado por el ensamblador. A continuación, mediante un programa **montador (link)** se introducen todas las rutinas externas. Éstas son partes del programa hechas de antemano, con el fin de facilitar la tarea del programador (por ej., el conjunto de instrucciones máquina que realizan un logaritmo, un seno, etc., o que permiten mostrar un mensaje por pantalla). El resultado es lo que se denomina *programa ejecutable*, es decir, un programa capaz de ser entendido directamente por el ordenador. Ejemplos de programas compilador son: C y Pascal.
- **Interpretados.** No se generan programas objeto, ni ejecutables. Se basan en un *programa intérprete* que toma una instrucción del programa fuente, la convierte a instrucción(es) máquina y la(s) ejecuta; repitiendo el proceso con la siguiente instrucción, y así hasta acabar el programa. Ejemplos de programas interpretados son: Basic y Prolog.

Dos de las principales ramas de la informática relacionadas con la creación de programas son:

- **La teoría de los lenguajes formales.** Se encarga del estudio de la estructura tanto sintáctica, como semántica de los lenguajes de programación.
- **La ingeniería del software.** Trata del estudio de metodologías eficientes para la construcción y mantenimiento del software.

1.4 El Computador

El computador u ordenador tal y como le conocemos actualmente, es el resultado de la evolución de las máquinas creadas por el hombre a lo largo de la historia, para potenciar el cálculo manual, reemplazándolo por el cálculo automático, más rápido, preciso y cómodo.

Para entender la estructura y funciones de cada uno de los principales elementos que componen un computador, vamos a establecer una analogía entre la forma de operar de éste, y el cálculo manual (ver figura 1).

A la hora de realizar un cálculo manualmente, ponemos en juego los siguientes elementos:

- Cerebro, con una triple función:
 - Controlar, dando las ordenes precisas al resto de elementos que intervienen en el cálculo.

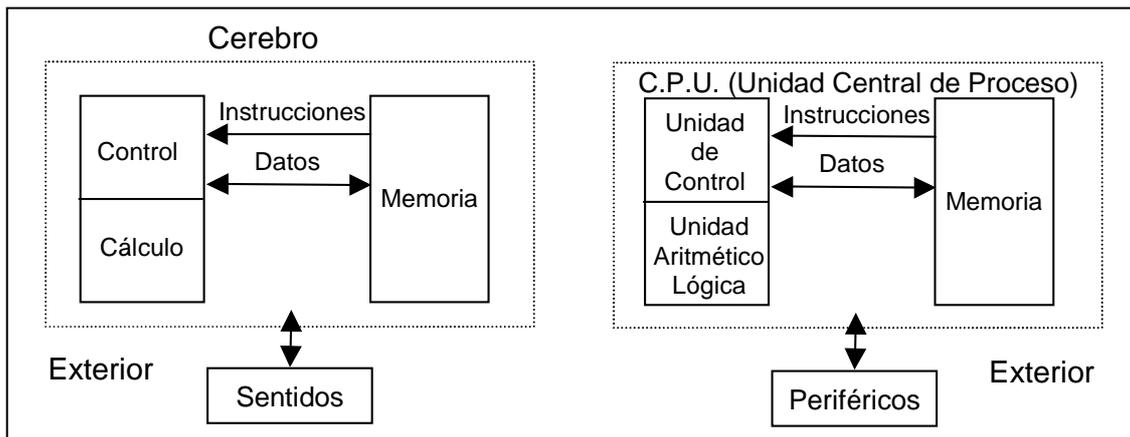


Figura 1 Analogía cálculo manual - cálculo automático.

- Realizar los cálculos oportunos.
- Memorizar tanto lo que queremos hacer y cómo hacerlo, como los datos y resultados.
- Sentidos, mediante los cuales nos comunicamos con el exterior. Cuando la información es de entrada, realizan la conversión de ésta a un formato que sea entendido por el cerebro (ej. vista: luz→impulsos eléctricos). Cuando la información va hacia el exterior realizan la conversión inversa (ej. habla: impulsos eléctricos→señal audible).

Un ordenador se compone básicamente de los mismo elementos, para ser más exactos, de una analogía de cada uno de ellos, como se puede ver en la figura 1. Cada elemento del ordenador realiza la misma función que su homólogo en el cálculo manual.

Según lo visto, podemos definir al ordenador como aquella máquina capaz de adquirir, representar, tratar y transmitir información.

El único lenguaje que entiende el ordenador es el binario, o sea, el construido mediante un alfabeto con dos símbolos {0,1}, el denominado alfabeto binario. La representación, por lo tanto, consiste en transformar cualquier información a este lenguaje.

Una vez representada la información, su tratamiento es el conjunto de procesos llevados a cabo para la resolución de un determinado problema.

1.4.1 Concepto de Hardware y Software

Dos conceptos importantes a la hora del estudio del computador son los de hardware y software.

Hardware. Se define como todo aquello que constituye la parte física o material de un ordenador. Es decir, los circuitos electrónicos y todos los componentes mecánicos tales como discos duros, impresoras, monitores, unidades de cinta, circuitos impresos, etc.

Software. Se define como la parte lógica o de programación de una máquina. Es el conjunto de algoritmos y su representación en la computadora, es decir, los programas. La esencia del software la constituyen el conjunto de instrucciones que forman el programa, no los medios físicos en los que se registran. El software es algo no material y que puede ser fácilmente modificado.

1.4.2 Estudio multinivel del ordenador

Hemos visto como podía ser el esquema básico de un hipotético computador: una memoria, un elemento de proceso (CPU), y una serie de dispositivos que permiten al computador comunicarse con el exterior, y entre cuyas funciones está la de representar la información transformándola a binario, y a la inversa. En la actualidad cada uno de estos elementos está construido mediante circuitos electrónicos, estos a su vez con elementos tales como transistores, diodos, resistencias, etc., y estos con material semiconductor, rigiéndose cada una de éstas partes por unas leyes distintas.

Si nos referimos al software, o sea, a los programas, también habrá una gran diferencia entre lo que se escribe en cualquier lenguaje y las instrucciones básicas que maneja directamente la computadora.

Tenemos, pues, que a la hora de estudiar un computador, existe una gran complejidad, ya que como acabamos de ver, podemos distinguir varios niveles dentro de la estructura de un computador, y cada uno de ellos regido por unas leyes y con unas características distintas de las de los otros. Por esta razón el estudio del computador se va a dividir en varios niveles, cada uno de los cuales hace abstracción de los detalles que no nos interesan en cada momento de los otros, permitiendo centrarse en los aspectos que se deseen estudiar, sin perder la visión global.

Dependiendo de la óptica con que se quiera abordar el estudio de un computador, se va a hacer una división en niveles diferente. Esto ha llevado a diversos autores a proponer su propia división. Entre estas están:

- Un visión basada en el estudio de las piezas con las que se construyen los ordenadores, orientada más hacia el hardware, tal y como proponen Bell y Newell en sus niveles estructurales. Estos autores realizan una división estructural, en la que las primitivas o bloques constructivos de cada nivel son los sistemas o conjuntos constructivos del nivel inferior.
- Otra posibilidad en la mostrada por Levy. Clasificación basada en el software, consiste en realizar el estudio del computador considerando éste como una serie de niveles tal que cada uno de ellos se comporta como un intérprete de las órdenes que recibe del nivel más inferior

Para cada uno de estos niveles, la máquina se comporta de un modo distinto, es decir, utiliza un conjunto de instrucciones y genera un conjunto de

servicios, que aparentemente nada tienen que ver son los de otros niveles, Es como si para cada nivel fuera una máquina diferente, a la que muchos autores han llamada **máquina virtual**.

1.5 Contenido de la Asignatura

Durante el presente curso, tanto en Fundamentos de Informática I como en Fundamentos de Informática II, nos vamos a centrar en el estudio del computador, desde distintos puntos de vista en cada una de esas asignaturas.

De las tareas que es capaz de realizar el computador, recordemos, *adquirir, representar, tratar y transmitir información*, en la presente asignatura nos vamos a centrar en las de *representar y transmitir*, dejando el cómo opera el ordenador al *adquirir y tratar información* para Fundamentos de Informática II.

El estudio de la representación y transmisión de la información lo vamos a realizar tanto desde un punto de vista teórico (temas 2, 3, 4, 5 y 7), como práctico (temas 6 y 8).

El enfoque teórico lo realizaremos mediante el estudio de una introducción a la **Teoría de la Información y la Codificación**, teoría que sienta las bases de las leyes y propiedades más importantes a tener en cuenta al abordar cualquier problema de codificación y transmisión de la información.

En cuanto a la parte práctica, veremos, en el tema 6, una serie de códigos de uso habitual a la hora de representar información en el ordenador, mientras en el tema 8 nos centraremos en el problema de la fiabilidad en la transmisión de la información, viendo algunas soluciones para mejorar esa.

Para acabar aclarar que la división que hemos hecho de la asignatura en punto de vista teórico y práctico, es una forma de estructurar su contenido, más que una realidad, ya que, como veremos, muchas de las conclusiones extraídas de la Teoría de la Información y la Codificación tienen una gran aplicación práctica. Es muy importante tener en cuenta este punto a la hora de abordar el estudio de la Teoría de la Información y la Codificación, ya que se puede tener la sensación de algo puramente teórico, y nada más lejos de la realidad, ya que sus conclusiones prácticas son fundamentales en los modernos sistemas de codificación y transmisión de la información.

Bibliografía

- Alberto Prieto Espinosa, Antonio Lloris Ruiz y Juan Carlos Torres Cantero, "Introducción a la Informática", Ed. McGraw-Hill, 2001, 3ª edición.
- Pedro de Miguel Anasagasti, "Fundamentos de los Computadores", Ed. Thomson, 2002, 8ª edición.
- Andrew S. Tanenbaum, "Organización de Computadoras: un Enfoque Estructurado", Ed. Prentice-Hall, 2000, 4ª edición.