

Introducción a los ordenadores

Versión: 16 de febrero de 2009

Las palabras españolas **informática** y **ordenador** provienen de las francesas **informatique** y **ordinateur**. La palabra francesa **informatique** se construyó como una contracción de las palabras **INFORM**ation y **auto**MATIQUE.

La palabra **computador** (o **computadora**, mayoritariamente utilizada en Sudamérica), proviene del término inglés **computer**. La ciencia que se ocupa de los ordenadores se denomina, en inglés, **Computer Science**, aunque también existe la palabra **Informatics**.

1.1 Componentes básicos de un ordenador

En un ordenador se distinguen principalmente dos aspectos:

- **HARDWARE**, el soporte físico, no modificable (placas, circuitos integrados, chips, módulos, cables, etc.), es decir la “maquinaria”.
- **SOFTWARE**, el conjunto de programas que se ejecutan en el ordenador, *grosso modo* divididos en:
 - Software del sistema o sistema operativo, el conjunto de programas necesarios para que el ordenador tenga capacidad de trabajar (funcionamiento de la pantalla, del teclado, movimientos del ratón, etc.).
 - Software de aplicación que son los programas que maneja el usuario (tratamiento de textos, hojas de cálculo, bases de datos, etc).

El modelo básico de los ordenadores actuales se atribuye a Von Neumann¹. Consta principalmente de la unidad central de proceso, la memoria y los dispositivos de entrada/salida, conectados entre sí como se muestra de forma esquemática en el diagrama mostrado en la Figura 1.1.

1.2 Unidad central de proceso

Unidad central de proceso o **CPU** (**C**entral **P**rocess **U**nit) es el “cerebro” del ordenador: ejecuta las instrucciones de los programas y controla el funcionamiento de los distintos componentes del ordenador. Suele estar integrada en un chip denominado *microprocesador*. Los dos componentes más importantes de la CPU son:

¹John von Neumann (1903-1957), matemático húngaro-estadounidense, pionero de los modernos ordenadores, que en un informe sobre el ordenador EDVAC, propuso el concepto de programa almacenado en la memoria del ordenador, junto con los datos.

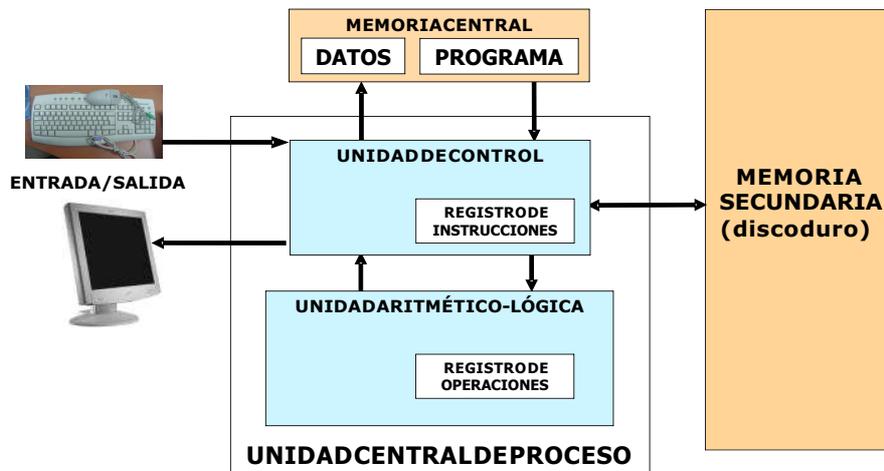


Figura 1.1: Componentes básicos de un ordenador

- La **unidad de control** o **CU** (**C**ontrol **U**nit), dirige y coordina la mayoría de las operaciones del ordenador. Interpreta cada instrucción enviada por un programa e inicia la acción apropiada para realizar esa instrucción. Para cada instrucción, la unidad de control repite un grupo de cuatro operaciones básicas, que constituye lo que se llama **ciclo de instrucción**:
 1. Leer (“fetch”) - es el proceso de obtener una instrucción de un programa o datos de la memoria.
 2. Decodificar (“decode”) - es el proceso de traducir la instrucción en comandos que el ordenador pueda ejecutar.
 3. Ejecutar (“execute”) - es el proceso de llevar a cabo los comandos.
 4. Escribir (“writeback”) - es el proceso de almacenar el resultado del paso de ejecución, “escribiéndolo” en la memoria.
- La **unidad aritmético-lógica** o **ALU** (**A**rithmetic **L**ogic **U**nit), que realiza las operaciones elementales que constituyen el programa (sumar, multiplicar, comparar, etc).

La velocidad de un procesador se suele medir en **hercios (Hz)**: número de operaciones por segundo que puede realizar.

$$\begin{aligned}
 1 \text{ Megahercio (Mz)} &= 10^6 \text{ operaciones/segundo} \\
 1 \text{ Gigahercio (Gz)} &= 10^9 \text{ operaciones/segundo}
 \end{aligned}$$

1.3 Memoria

Son los componentes de hardware en los que se almacena la información procesada por el ordenador. Generalmente, se distinguen entre la **memoria central** y la **memoria secundaria** (véase más adelante). La principal característica de las memorias es su:

- **Capacidad:** indica la cantidad de datos que puede almacenar. Se mide en bits **bit** (**b**inary **u**nit), o múltiplos del bit. Los bits suelen agruparse de ocho en ocho: **1 byte**= 8 bits. Las unidades de almacenamiento están recogidas en la Tabla 1.1.

bit	0 ó 1
byte (B)	8 bits
Kilobyte (KB)	2^{10} bytes = 1024 bytes
Megabyte (MB)	2^{10} KB = 1024 KB
Gigabyte (GB)	2^{10} MB = 1024 MB
Terabyte (TB)	2^{10} GB = 1024 GB

Tabla 1.1: Unidades de almacenamiento de información

1.3.1 Memoria central

Memoria central o principal, almacena tanto la secuencia de instrucciones del programa o programas que se esté ejecutando en cada momento como los datos que éste o éstos necesitan. Se distingue entre:

- Memoria **ROM** (**R**ead **O**nly **M**emory), (memoria de sólo lectura). Es permanente, esto es no se borra al apagar el ordenador y no se puede alterar (o se puede hacer difícilmente). Almacena códigos de programa grabados en fábrica necesarios para el funcionamiento del ordenador, como por ejemplo, la secuencia de instrucciones que hay que ejecutar cuando se enciende el ordenador (BIOS: Basic Input/Output System; POST: Power On Self Test).
- Memoria **RAM** (**R**andom **A**cces **M**emory), (memoria de acceso aleatorio). Almacena las instrucciones de los programas mientras que se ejecutan así como los datos que éstos necesitan o generen. Su contenido es volátil, es decir, se borra al apagar el ordenador. La denominación Acceso Aleatorio es antigua y se comenzó a utilizar para diferenciarla de otro tipo de memoria (cintas magnéticas) que era de acceso secuencial: para acceder a un dato había que recorrer todos los anteriores.

Con el objetivo de que el procesador pueda obtener los datos de la memoria central más rápidamente, la mayoría de los procesadores usan un tipo especial de memoria llamada **memoria caché** ó **RAM caché** (pequeña y de acceso muy rápido). En ella, además se almacenan los datos más recientes y las instrucciones inminentes.

La memoria central es un circuito que se puede imaginar como una enorme tabla que almacena información en cada una de sus **celdas** o **posiciones de memoria** (véase Figura 1.2).

Cada celda es una agrupación de bytes que se denomina **palabra**. Dependiendo del ordenador, existen palabras de 4 bytes (32 bits), 8 bytes (64 bits). Las palabras están numeradas correlativamente, comenzando desde 0. El número de una palabra es la **dirección** de esa palabra. La información almacenada en una palabra es su **contenido**.

1.3.2 Memoria secundaria

Como hemos mencionado, toda la información en la memoria central es accesible directamente en un tiempo muy corto; por ello dicha memoria es cara.

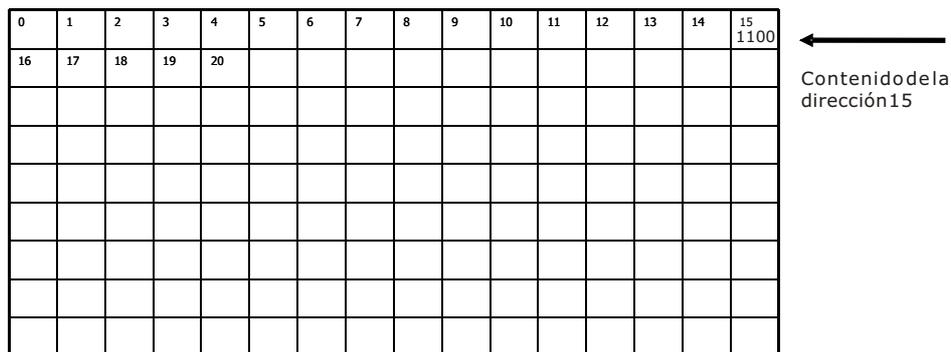


Figura 1.2: Memoria central de un ordenador

Memoria secundaria es una memoria más barata donde, a cambio de mayores tiempos de acceso, se puede conservar gran cantidad de información (disco duro, CD (Compact Disk), DVD (Digital Versatile Disk), memoria flash, etc). En ella se almacena todo lo que la unidad central de procesos (véase la Sección 1.2) no necesita urgentemente. La principal característica de este tipo de memorias es que su contenido no se pierde al apagar el ordenador.

1.4 Dispositivos de entrada/salida

Dispositivos de **entrada/salida** son todos los componentes de hardware anexos que permiten la comunicación entre el ordenador y el usuario. Los dispositivos de entrada son, por ejemplo, teclados, ratones, lectores de discos, micrófonos, etc. Los dispositivos de salida son, por ejemplo, monitores, impresoras, etc.

1.5 Sistema operativo

El **sistema operativo** es el software más importante que se ejecuta en un ordenador. Se trata de un conjunto de mecanismos y reglas básicas de funcionamiento que hace posible una utilización eficiente de los recursos de un ordenador. Ejemplos de sistemas operativos son WINDOWS, MacOS, UNIX, Linux. Se compone de un paquete integrado de programas de dos tipos:

- **Programas de control:** gestionan el hardware y software, por ejemplo colas de impresión etc.
- **Utilidades del sistema:** programas de ayuda al usuario, por ejemplo editores de texto, gestión de correo, etc.

1.6 Lenguajes

Un **lenguaje de programación** es un conjunto de símbolos y reglas sintácticas y semánticas que sirven para describir las órdenes que controlan el comportamiento físico y lógico de una máquina.

Los procesadores de las máquinas sólo son capaces de entender y obedecer programas escritos en **lenguaje-máquina**, cuyas instrucciones son cadenas binarias (formadas por 0 y 1) que se pueden

“cargar” directamente en la memoria central, sin necesidad de traducción. Sin embargo, el lenguaje-máquina es específico de cada tipo de ordenador, por lo que los programas escritos en dicho lenguaje no son portables en general de una máquina a otra. Además, debido a su representación totalmente numérica, son muy difíciles de escribir, leer y corregir.

El lenguaje **ensamblador** facilita (sólo un poco) esas tareas, ya que permite la escritura de las instrucciones básicas “entendibles” por la CPU en una forma más legible para el programador.²

A modo de ejemplo, una instrucción básica ejecutable por la CPU podría ser:

copiar el contenido de la celda 164 en la celda 183.

En lenguaje-máquina esta instrucción podría escribirse (sólo es un ejemplo)

$$\begin{array}{ccc} \underbrace{01101100} & \underbrace{10100100} & \underbrace{10110101} \\ \text{COPIAR} & 164 & 183 \end{array}$$

mientras que en lenguaje ensamblador se podría escribir de forma parecida a:

CP 164 183

Usualmente hay una correspondencia uno a uno entre las instrucciones simples de un código ensamblador y las de un código máquina. Por ello, el ensamblador sigue siendo un lenguaje de *bajo nivel*: por un lado el programador necesita conocer en profundidad la arquitectura de la máquina, por otro los programas escritos en ensamblador no son portables.

Los **lenguajes de alto nivel**, por el contrario, no obligan al programador a conocer los detalles del ordenador que utiliza. Las instrucciones se escriben en un formato flexible y más “humano”. Además, se pueden escribir, de forma sencilla, instrucciones mucho más complicadas: cada instrucción en un lenguaje de alto nivel corresponde a varias (incluso muchas) de lenguaje-máquina. La instrucción de los ejemplos anteriores, se podría escribir:

$A = B$

El ordenador sólo “comprende” los programas escritos en lenguaje-máquina. Cualquier otro debe ser **traducido**.

La traducción de un programa escrito en ensamblador a lenguaje-máquina la realiza un programa específico denominado también **ensamblador**.

Un programa escrito en un lenguaje de alto nivel se denomina **programa fuente**. Para traducirlo al lenguaje-máquina se puede usar:

- Un **compilador**: es un programa informático que traduce un programa fuente a un programa equivalente escrito en lenguaje-máquina, que se denomina **programa objeto**. El programa objeto puede ser almacenado como archivo en la memoria secundaria del ordenador para ser ejecutado posteriormente sin necesidad de volver a realizar la traducción.
- Un **intérprete**: traduce el código fuente instrucción a instrucción y la ejecuta en el instante. No se crea un archivo o programa objeto, de modo que hay que volver a traducir cada vez que usemos el programa fuente correspondiente.

²Fue usado ampliamente en el pasado para el desarrollo de software, pero actualmente sólo se utiliza en contadas ocasiones, especialmente cuando se requiere la manipulación directa del hardware o se pretenden rendimientos inusuales del los equipos.