

# Introducción

Este libro surge con el propósito de acercar al lector a los aspectos más importantes que encierran los sistemas operativos monopuesto ante la creciente demanda de personal cualificado para su administración. Con tal propósito, puede servir de apoyo también para estudiantes del Ciclo Formativo de Grado Medio de **Informática** y las **Ingenierías Técnicas**.

Hoy en día, existen muchos usuarios y profesionales de la Informática que discuten las ventajas e inconvenientes de algunos sistemas operativos monopuesto y prefieren limitarse al uso exclusivo de uno de ellos. Aquí no hay preferencia por ningún sistema en particular, ni se intenta compararlos para descubrir cuál es el mejor de todos, sino enriquecer los contenidos al exponer sus principales características, manejo y métodos para conseguir la coexistencia entre ellos.

Para todo aquel que use este libro en el entorno de la enseñanza (ciclos formativos o universidad), se ofrecen varias posibilidades: utilizar los conocimientos aquí expuestos para inculcar aspectos genéricos de los sistemas operativos monopuesto o simplemente centrarse en preparar a fondo alguno de ellos. La extensión de los contenidos aquí incluidos hace imposible su desarrollo completo en la mayoría de los casos.

Ra-Ma pone a disposición de los profesores una página web para el desarrollo del tema donde se incluyen las soluciones a los ejercicios expuestos en el texto, un glosario, bibliografía y diversos recursos para suplementar el aprendizaje de los conocimientos de este módulo. Puede solicitar el medio de acceso a dichos recursos a [editorial@ra-ma.com](mailto:editorial@ra-ma.com), acreditándose como docente y siempre que el libro sea utilizado como texto base para impartir las clases.



# 1

# Elementos, estructura y funciones de un sistema operativo

## OBJETIVOS DEL CAPÍTULO

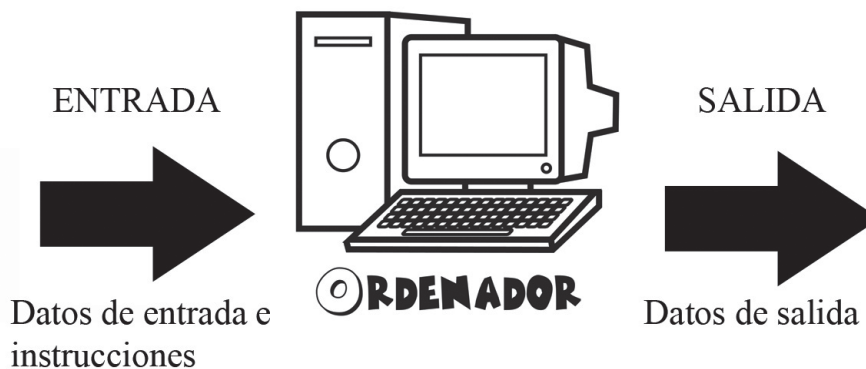
- ✓ Conocer qué es un sistema informático.
- ✓ Trabajar con los distintos sistemas de representación.
- ✓ Saber qué es un sistema operativo.
- ✓ Ver los elementos y estructura de un sistema operativo.
- ✓ Conocer las funciones de un sistema operativo.
- ✓ Identificar los procesos y sus estados.
- ✓ Ver los distintos tipos de sistemas operativos.
- ✓ Conocer algunos de los sistemas operativos actuales para microordenadores.
- ✓ Conocer cómo arrancar y parar el sistema operativo.

## 1.1 EL SISTEMA INFORMÁTICO

La palabra **informática** proviene de la contracción de los vocablos **INFORM**ación y auto**MÁTICA**. El diccionario de la Real Academia Española la define como “*el conjunto de conocimientos científicos y técnicas que hacen posible el tratamiento automático de la información por medio de ordenadores*”.

La palabra **ordenador, computador o computadora**, que comúnmente se utiliza para referirse a las máquinas informáticas, está definida en el diccionario de esta forma: “*máquina electrónica dotada de una memoria de gran capacidad y de métodos de tratamiento de la información, capaz de resolver problemas aritméticos y lógicos gracias a la utilización automática de programas registrados en ella*”.

Un ordenador transforma los datos de entrada en una salida conteniendo la información requerida. Los datos son conjuntos de símbolos utilizados para representar un valor numérico, un objeto o una idea en la forma adecuada para ser objeto de tratamiento. La representación de los datos se hace en base a un determinado código que le confiere su significado. Así, por ejemplo, para representar números se suelen utilizar los dígitos decimales, aunque se estudiarán muchos otros códigos o formas de representación.



*Figura 1.1. Representación esquemática del funcionamiento de un ordenador*

El vertiginoso desarrollo de la informática en las últimas décadas ha impulsado, a su vez, multitud de áreas de la sociedad (tecnología, economía, ciencias, investigación...), de manera que hoy en día parece imposible pensar en la mayoría de las actividades de nuestra sociedad sin que aparezca algún sistema informático.

La mayoría de las personas utilizan diariamente un ordenador, ya sea directa y voluntariamente (escribir una carta, consultar el correo electrónico, escuchar música, navegar por Internet, ver una película, realizar cálculos complejos...) o indirectamente (sacar un billete de tren, utilizar un cajero automático, hacer la declaración de la renta, etc.).

Un **sistema informático** puede definirse como un conjunto de partes interrelacionadas. Un sistema informático típico emplea un ordenador que usa dispositivos programables para capturar, almacenar y procesar datos. Dicho ordenador, junto con la persona que lo maneja y los periféricos que lo envuelven, resultan de por sí un ejemplo de un sistema informático.

Estructuralmente, un sistema informático se puede dividir en partes, pero funcionalmente es indivisible, en el sentido de que si se divide, pierde alguna de sus propiedades esenciales. Por eso, un sistema informático sin alguna de sus partes no funcionaría.

Todo sistema informático está compuesto por tres elementos básicos:

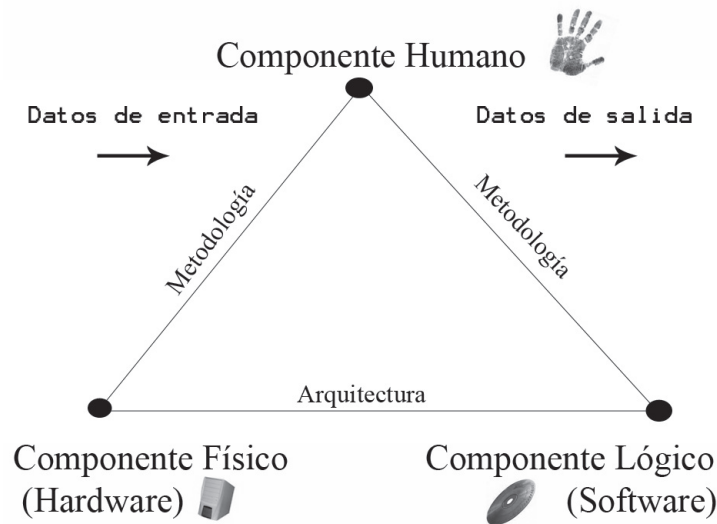
- Un **componente físico** (hardware): incluye las placas, circuitos integrados, conectores, cables y sistema de comunicaciones.
- Un **componente lógico** (software): permite disponer de un lenguaje lógico para comunicarse con el hardware y controlarlo. Hay dos tipos de software:
  - **Software de base**: es el conjunto de programas necesarios para que el hardware tenga capacidad de trabajar (hacen posible que la pantalla funcione, que represente lo que se escribe desde el teclado, comunicarse con los periféricos, etc.). Recibe también el nombre de **sistema operativo**.



### ¿SABÍAS QUE...?

El software de base hace posible que la pantalla funcione, que represente lo que se escribe desde el teclado, que el equipo se comunique con los periféricos, etc.

- **Software de aplicación**: son los programas que maneja el usuario (tratamiento de textos, bases de datos, hojas de cálculo...).
- **Componente humano**: está constituido por las personas que participan en la dirección, diseño, desarrollo, implantación y explotación de un sistema informático.



*Figura 1.2. Representación de la estructura de un sistema informático*

### 1.1.1 TIPOS DE SISTEMAS INFORMÁTICOS

Los sistemas informáticos se pueden clasificar de dos maneras:

1. Por su **uso** se pueden clasificar en:

- **Sistemas Informáticos de uso general:** son los que se utilizan para varios tipos de aplicaciones (es el caso de los ordenadores personales).
- **Sistemas Informáticos de uso específico:** son los que se caracterizan por ejecutar uno o unos pocos programas (es el caso de los robots industriales o de los videojuegos).

2. Por sus **prestaciones** se pueden clasificar en:

- **Supercomputadores:** son equipos con gran capacidad de cálculo. Se utilizan en el entorno técnico-científico y en la realización de simulaciones, por ejemplo, la *Roadrunner* (Correcaminos), que se utiliza para simular explosiones atómicas, ya que es capaz de realizar 1.000 billones de cálculos por segundo (1 **petaflops** o 1.000 **teraflops**).
- **Sistemas grandes, computadoras centrales o mainframes:** son equipos utilizados para dar soporte a grandes redes de comunicaciones con cientos e incluso miles de usuarios. Por ejemplo, para el procesamiento de las transacciones bancarias.

La distinción entre supercomputadores y computadoras centrales no es sencilla, pero se puede decir que las supercomputadoras se centran en los problemas limitados por la velocidad de cálculo, mientras que las computadoras centrales se centran en problemas limitados por los dispositivos de E/S y la fiabilidad de las transacciones.

- **Sistemas medios o miniordenadores:** son equipos con capacidad para soportar cientos de usuarios con un coste y unas prestaciones inferiores a los grandes sistemas. Son también conocidos como servidores de redes con **terminales tontos** sin capacidad de cálculo propia. Actualmente están en desuso, siendo remplazados por microordenadores interconectados entre sí (denominados **estaciones de trabajo**) y con un **servidor**. Es preciso distinguir estas estaciones de trabajo de las referenciadas en el punto siguiente.
- **Estaciones de trabajo:** son equipos monousuarios muy potentes y especializados, algunos con tecnología RISC, que permiten conseguir un aumento de prestaciones (por ejemplo, para diseño y CAD). Siguiendo las tendencias de rendimiento de las computadoras en general, los microordenadores actuales son más potentes que las mejores estaciones de trabajo de anteriores generaciones. Como resultado, el mercado de las estaciones de trabajo se está volviendo cada vez más especializado, ya que muchas operaciones complejas que antes requerían sistemas de alto rendimiento pueden ser ahora realizadas por computadores de propósito general. Sin embargo, el hardware de las estaciones de trabajo está optimizado para situaciones que requieren un alto rendimiento y fiabilidad, donde un microordenador podría dejar de responder.
- **Microordenadores:** son equipos monousuarios menos potentes que se pueden clasificar de varias maneras. Entre ellas se encuentran: ordenadores profesionales y personales, ordenadores de oficina y domésticos, y ordenadores de escritorio y portátiles.

## ACTIVIDADES 1.1



- Busque información sobre varias supercomputadoras.

## 1.2 EL ORDENADOR

Un ordenador puede definirse como “*el dispositivo, o conjunto de dispositivos programables, compuesto por una o varias unidades de procesamiento, recursos y equipos periféricos destinados al procesamiento de grandes cantidades de datos, capaz de efectuar distintos tipos de cálculos, incluyendo operaciones aritméticas y lógicas a gran velocidad*”.

La primera “máquina” diseñada para realizar cálculos fue el ábaco. Siglos después (alrededor del año 1620), aparecieron las calculadoras mecánicas basadas en engranajes, capaces de realizar operaciones elementales. Estas no eran máquinas automáticas, ya que requerían la intervención humana durante el proceso de cálculo.

Estos dispositivos mecánicos fueron evolucionando hasta que, en 1890, Herman Hollerith creó una máquina para realizar el censo de EE.UU. Esta máquina utilizaba un sistema electrónico para la lectura de las tarjetas perforadas y un sistema mecánico para calcular. En 1924, la compañía fundada por Hollerith cambia de nombre para denominarse **International Business Machines (IBM)**. Comienza la época de los computadores.

### 1.2.1 ARQUITECTURA DEL ORDENADOR

En 1944, John **von Neumann** describió en su famoso modelo, un computador con programa almacenado en memoria eléctrica. Este modelo se utilizó en la construcción del **EDVAC** (*Electronic Discrete Variable Automatic Computer*) en 1952 y es la que se utiliza en la mayoría de los ordenadores actuales.

Según la arquitectura de von Neumann, un computador está formado por:

- Una **Unidad Aritmético-Lógica (ALU)**, que realiza cálculos y comparaciones y toma decisiones lógicas (determina si una afirmación es cierta o falsa mediante las reglas del Álgebra de Boole).



A mediados del siglo XIX, **George Boole** desarrolló la idea de que las proposiciones lógicas podían ser tratadas mediante herramientas matemáticas. Las proposiciones lógicas (asertos, frases o predicados de la lógica clásica) son aquellas que únicamente pueden tomar valores Verdadero/Falso, o preguntas cuyas únicas respuestas posibles sean Sí/No. Según Boole, estas proposiciones pueden ser representadas mediante símbolos y la teoría que permite trabajar con estos símbolos, sus entradas (variables) y sus salidas (respuestas), a través de la **Lógica Simbólica** desarrollada por él. Dicha lógica simbólica cuenta con operaciones lógicas que siguen el comportamiento de reglas algebraicas. Por ello, al conjunto de reglas de la Lógica Simbólica se le denomina **ÁLGEBRA DE BOOLE**.

- Una **Unidad de Control (UC)**, que interpreta cada una de las instrucciones del programa en lenguaje máquina y, de acuerdo con su microprogramación, ir generando las señales lógicas para que se realicen las modificaciones sobre los registros y/o las posiciones de la memoria principal correspondientes a dicha instrucción de la máquina.
- La **Memoria**, que está formada por los elementos que permiten almacenar y recuperar la información y una serie de **Registros** donde se almacena información temporalmente.





En un sentido más amplio, el concepto de memoria puede referirse también a sistemas externos de almacenamiento, como las unidades de disco.

- Los sistemas de **Entrada/Salida**, que permiten la comunicación con los dispositivos periféricos.

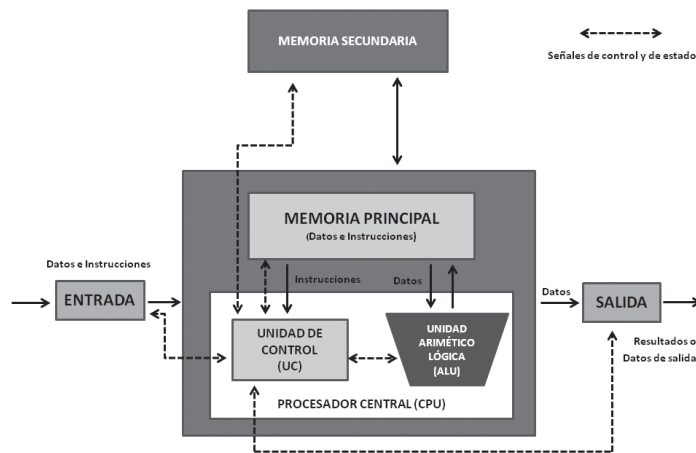


Figura 1.3. Arquitectura de un ordenador

- La **Unidad Central de Proceso (CPU)**, que no es más que un circuito secuencial que interpreta y ejecuta instrucciones. La CPU se compone de los dos primeros elementos del computador (la UC y la ALU) y de los registros. Habitualmente, la CPU es un microprocesador fabricado en un chip.



### ¿SABÍAS QUE...?

Un chip es un único trozo de silicio que contiene millones de componentes electrónicos.

## ACTIVIDADES 1.2



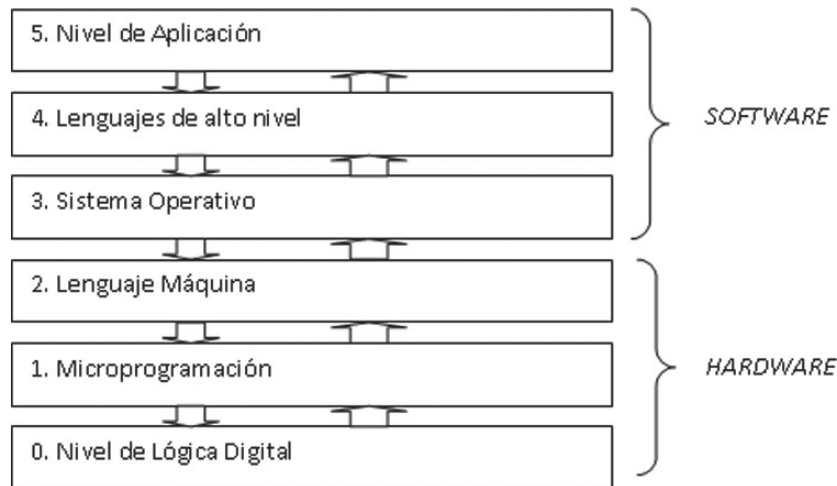
- Busque información sobre von Neumann.



### 1.2.2 VISIÓN JERÁRQUICA DE UN COMPUTADOR

Para comprender el funcionamiento de un computador es necesario tener una visión global de su estructura jerárquica. Es decir, es importante darse cuenta de que el sistema completo está formado a su vez por subsistemas relacionados entre sí de forma escalonada.

- **Nivel 0 - Nivel de Lógica Digital.** Se corresponde con el hardware real de la máquina (dispositivos y circuitos electrónicos).
- **Nivel 1 - Microprogramación.** En este nivel se encuentran los microprogramas, cuya tarea consiste en interpretar las instrucciones de nivel superior.
- **Nivel 2 - Lenguaje Máquina.** Es el nivel inferior accesible por el usuario y se corresponde con el conjunto de instrucciones que forman el lenguaje directamente interpretable por el hardware, los modos de direccionamiento, los tipos de datos, organización del subsistema de memoria, etc.
- **Nivel 3 - Sistema Operativo.** Es el conjunto de programas que proporcionan facilidades a los niveles superiores en la gestión de los recursos del sistema. De esta manera, se crea un entorno favorable para que el usuario interactúe con los niveles inferiores de la máquina.
- **Nivel 4 - Lenguajes de Alto Nivel.** Son lenguajes de programación con alta abstracción respecto al hardware. El programador encuentra un entorno amigable para la codificación de algoritmos.
- **Nivel 5 - Nivel de Aplicación.** Es el nivel más alejado de la realidad física en el cual el usuario no tiene en cuenta los niveles inferiores.



*Figura 1.4. Niveles de una computadora*

## 1.3 **NORMATIVA LEGAL RELATIVA A LA INFORMÁTICA**

Los **derechos de autor** son un conjunto de normas y principios que regulan los derechos morales y patrimoniales que la ley concede a los autores, por el hecho de haber creado una obra literaria, artística, científica o didáctica, esté publicada o inédita.

Los **derechos morales** incluyen dos aspectos específicos: el derecho al reconocimiento de la paternidad de la obra (autoría) y el derecho de un autor a preservar la integridad de la obra (es decir, a negarse a la realización de modificaciones u obras derivadas de la misma).

Los **derechos patrimoniales** son aquellos que se reflejan sobre el patrimonio y permiten satisfacer necesidades valorables en dinero.

Se suele utilizar la denominación de **copyright** (derecho de copia) para especificar la parte patrimonial de los derechos de autor.

Una obra pasa al dominio público cuando los derechos patrimoniales han expirado. Esto sucede trascurrido un plazo desde la muerte del autor. Por ejemplo, en el derecho europeo, 70 años desde la muerte del autor. Dicha obra entonces puede ser utilizada de forma libre, respetando los derechos morales.

En España se desarrollan los derechos de autor en la Ley de la Propiedad Intelectual.

### 1.3.1 **TIPOS DE LICENCIA DE SOFTWARE COMERCIAL**

Una **licencia de software** es una autorización formal con carácter contractual que un autor de un software da a un interesado para ejercer actos de explotación cumpliendo una serie de términos y condiciones establecidas dentro de sus cláusulas.

Desde el punto de vista del software comercial, existen distintos grupos de licencias:

- **Licencias OEM:** se trata de un tipo de licencia que supedita su venta a que forme parte de un equipo nuevo, estando prohibido venderlo si no es bajo esta condición. Si bien afecta más que nada a sistemas operativos, también puede afectar a otro tipo de software. Aunque el software comprado bajo este tipo de licencia implica la propiedad del mismo por parte del que lo compra, los fabricantes pueden poner ciertas limitaciones a su uso, como el número máximo de veces que se puede reinstalar.

Los programas adquiridos bajo este tipo de licencia NO se pueden vender ni ceder a terceros, salvo en las mismas condiciones en las que se compraron (es decir, como parte de un equipo).

- **Licencias Retail:** son las versiones de venta de software. En este caso, el programa es de la entera propiedad del usuario, pudiendo éste cederlo libremente a terceros o venderlo.
- **Licencias por volumen:** es un tipo de licencia de software destinado a grandes usuarios (empresas), normalmente bajo unas condiciones similares a las de las licencias OEM, aunque sin estar supeditadas a equipos nuevos.

Básicamente, se trata de estipular un determinado número de equipos que pueden utilizar el mismo código de licencia, quedando el fabricante de dicho software autorizado para hacer las comprobaciones que considere oportunas para ver que las licencias que se están utilizando son las adquiridas.

Normalmente, estas licencias se venden en paquetes de  $x$  número de licencias (por ejemplo, en paquetes de 25 licencias como mínimo).

Este tipo de licencia NO se puede ceder a terceros, ni total ni parcialmente.

## ACTIVIDADES 1.3



- Si cuenta con una versión de Windows en su ordenador, averigüe si es una versión *OEM*, *retail* o si dispone de una *licencia por volumen*.

### 1.3.2 TIPOS DE LICENCIA DE SOFTWARE LIBRE

Desde el punto de vista del software libre, existen distintos grupos de licencias:

- **Licencias BSD** (*Berkeley Software Distribution*). Este tipo de licencias se utilizan en el software distribuido junto a los sistemas operativos BSD (sistemas operativos derivados del sistema Unix desarrollados por la Universidad de California en Berkeley). Este tipo de licencias mantienen la protección de los derechos de autor únicamente para la renuncia de garantía y para requerir la adecuada atribución de la autoría en trabajos derivados, pero permite la libre redistribución y modificación, incluso si dichos trabajos tienen propietario. Son muy permisivas y son fácilmente absorbidas al mezclarse con las licencias GNU GPL, con las cuales son compatibles.
- **Licencias GNU GPL** (Licencia Pública General de GNU). Este tipo de licencias conservan los derechos de autor y permiten la redistribución y modificación bajo términos diseñados para asegurarse de que todas las versiones modificadas del software permanecen bajo los términos más restrictivos de la propia GNU GPL. Esto hace que no se pueda crear un producto con partes no licenciadas GPL. Por lo tanto, el conjunto tiene que ser GPL.

Es decir, esta licencia obliga a que si se reutiliza en un mismo programa código “A” licenciado bajo licencia GNU GPL y código “B” licenciado bajo otro tipo de licencia libre, el código final “C”, independientemente de la cantidad y calidad de cada uno de los códigos “A” y “B”, deberá estar bajo licencia GNU GPL.

Aproximadamente el 60% del software licenciado como software libre emplea una licencia GPL.



### ¿SABÍAS QUE...?

**GNU** es un acrónimo recursivo que significa *GNU No es Unix*.

- **Licencias GNU LGPL** (Licencia Pública General Limitada de GNU). La licencia GPL ofrece grandes beneficios pero hay algunas veces en las que ofrece ciertas restricciones. Por ejemplo, cuando se utilicen partes o bibliotecas de software GPL en un software propietario o distribuido bajo otra licencia, estaría obligada a distribuir todo su software bajo GPL, lo que, seguramente, no les interesaría.

Sin embargo, con este tipo de licencias, que es una versión reducida de la licencia GNU GPL LGPL, no es obligatorio que el software derivado tenga que ser libre.

Otras actividades que no sean copia, distribución o modificación no están cubiertas en esta licencia y están fuera de su alcance.

Como ejemplo de este tipo de licencia se puede citar el proyecto OpenOffice.org de Sun Microsystems.

- **Licencias MPL** (Licencia Pública de Mozilla). Este tipo de licencia evita el efecto de la GPL (es decir, que si se utiliza código licenciado GPL, el desarrollo final tiene que ser obligatoriamente licenciado GPL). No obstante, la licencia MPL no es tan excesivamente permisiva como las licencias BSD.

Tiene un gran valor porque fue el instrumento que empleó Netscape Communications Corp. para liberar su Netscape Communicator 4.0 y empezar con el proyecto Mozilla.

Se utilizan en gran cantidad de productos de software libre de uso cotidiano en todo tipo de sistemas operativos, como por ejemplo, Mozilla Firefox.

- **Licencias Copyleft** ©. Este tipo de licencia permite ejercer el derecho de autor permitiendo la libre distribución de copias y versiones modificadas de una obra u otro trabajo, exigiendo que los mismos derechos sean preservados en las versiones modificadas.

Hay que hacer constar que el titular de los derechos de autor (*copyright*) de un software bajo licencia *copyleft* puede también realizar una versión modificada bajo su *copyright* original y venderla bajo cualquier licencia que desee, además de distribuir la versión original como software libre.

Esta técnica ha sido usada por una serie de empresas que realizan software libre (por ejemplo MySQL).

- **Multilicencia**. Con este tipo de licencias, una obra se puede publicar bajo los términos de varias licencias. De esta manera, terceras personas pueden reproducir, distribuir, comunicar públicamente, interpretar y transformar, por cualquier medio, la obra, en cualquier momento o lugar, licenciando o multilicenciando, según sea el caso, la obra original o la obra derivada, bajo los términos de una de las licencias o bajo los términos de un subconjunto de ellas, sujeto a las condiciones particulares que recojan las definiciones de cada una de las licencias.

## 1.4 CÓMO MEDIR LA INFORMACIÓN

La cantidad mínima de información es la contenida en un **bit** (acrónimo procedente del inglés: *BI*nary *di*giT), que representa el 0 ó el 1, el sí o el no, el blanco o el negro (cuando se utiliza para codificar un archivo gráfico), el verdadero o el falso, etc. Es, por ejemplo, la cantidad de información que necesitaba un padre a la puerta del paritorio para saber si se trataba de niño o de niña; actualmente esta pregunta se desvela en las primeras ecografías, pero sigue conteniendo 1 bit de información (código: 0 ⇒ niño, 1 ⇒ niña, o viceversa).

Un **byte** es el número de bits necesarios para almacenar un carácter y, como generalmente los códigos más usuales utilizan 8 bits para representar cada carácter, siempre que no se indique lo contrario, se considera que 1 byte es equivalente a 8 bits.

Tanto los bits (**b**) como los bytes (**B**) tienen toda una serie de múltiplos que se forman de manera sistemática añadiendo el prefijo correspondiente:

**Tabla 1.1** Equivalencia entre las distintas unidades

1 Kilobyte (KB)	= $2^{10}$ Bytes	=1.024 Bytes	≈ $10^3$ Bytes
1 Megabyte (MB)	= $2^{20}$ Bytes	=1.048.576 Bytes	≈ $10^6$ Bytes
1 Gigabyte (GB)	= $2^{30}$ Bytes	=1.073.741.824 Bytes	≈ $10^9$ Bytes
1 Terabyte (TB)	= $2^{40}$ Bytes	=1.099.511.627.776 Bytes	≈ $10^{12}$ Bytes
1 Petabyte (PB)	= $2^{50}$ Bytes	=1.125.899.906.842.624 Bytes	≈ $10^{15}$ Bytes
1 Exabyte (EB)	= $2^{60}$ Bytes	=1.152.921.504.606.846.976 Bytes	≈ $10^{18}$ Bytes

Para tener una idea de lo que representan estas cantidades de información, pensemos que el fichero del diccionario de la Real Academia incluido en la Biblioteca Encarta ocupa algo más de 12 MB; una imagen de 800 por 600 píxeles usada para el escritorio de Windows en color verdadero ocupa 1,3 MB y, por ejemplo, cada minuto de música estéreo en calidad CD viene a suponer 10 MB en formato *wav*, mientras que solo 1 MB en formato *mp3*.

### ACTIVIDADES 1.4



➤ ¿Cuántos bits hay en 64 KB?, ¿y en 16 TB?, ¿y en 40 GB?

## 1.5 REPRESENTACIÓN DE LA INFORMACIÓN

Internamente, los computadores solo reconocen dos símbolos: el 0 y el 1. En cambio, la información manejada por el hombre se representa mediante cifras decimales y letras mayoritariamente, pero no podemos olvidar las imágenes (gráficos, fotografías y vídeo) y los sonidos.

Los factores a tener en cuenta a la hora de seleccionar la codificación binaria más conveniente para cada tipo de datos son:

- ✓ Coste de la traducción.
- ✓ Coste del almacenamiento de la información.
- ✓ Coste del tratamiento de la información.

En el caso de la información textual, la traducción es muy sencilla, pues se hace letra a letra y no se escatima en el coste de almacenamiento pero, para datos numéricos, la situación es variable y, en las distintas formas de representación, hay que analizar el coste de almacenamiento, la complejidad en la traducción, la facilidad para realizar las operaciones numéricas y los rangos de representación.

Otro aspecto en la representación de la información es la tolerancia a fallos en las operaciones de lectura y escritura y en las transmisiones, lo que se consigue incorporando redundancia en la codificación para detectar y corregir los posibles errores mediante chequeos. Además, cuando la cantidad de información a representar es muy grande, como en el caso de un vídeo, se han buscado métodos de compactación para facilitar su almacenamiento y retransmisión a través de la red.

### 1.5.1 SISTEMAS DE NUMERACIÓN USUALES

Un **sistema de numeración** es una forma de representar cualquier cantidad numérica. Casi todos los sistemas de numeración utilizados en la actualidad son de tipo polinomial. Todo **sistema polinomial** cumple las siguientes características:

- ✓ Todo número es una expresión formada por un conjunto de símbolos, llamados dígitos, cada uno con un valor fijo y diferente a los demás.
- ✓ El número de símbolos distintos que se pueden usar en un determinado sistema de numeración constituye su “base”, es decir, en **base 10 (sistema decimal)** los números que se pueden representar son {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}, en base 2 son {0, 1}, en base 8 son {0, 1, 2, 3, 4, 5, 6, 7} y en base 16 {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}.
- ✓ El valor numérico que expresa una determinada combinación de dígitos en una base de numeración dada depende de dos factores: del **valor** de los dígitos y de la **posición** de cada uno de ellos en el polinomio.
- ✓ Cada posición del dígito tiene un valor intrínseco que aumenta de derecha a izquierda según potencias sucesivas de la base del sistema de numeración empleado. El dígito que aparece en el extremo izquierdo es el de más valor o más peso y el colocado en el extremo derecho es el de menos valor o menor peso.

De forma general, en un sistema de numeración de **base b** ( $b$  entero y mayor que la unidad), un número  $N$  cualquiera (representado como  $N_b$ ) se puede expresar mediante un polinomio de potencias de la base, multiplicadas por un símbolo perteneciente al sistema de numeración. En el siguiente ejemplo se representan algunos números en forma polinomial:

**EJEMPLO 1.1**

$$458 \text{ (base 10)} = 458_{10} = 4 * 10^2 + 5 * 10^1 + 8 * 10^0$$

458<sub>10</sub>: el número 4 es el de mayor peso y el 8 el de menor peso.

$$1110000 \text{ (base 2)} = 1 * 2^6 + 1 * 2^5 + 1 * 2^4 + 0 * 2^3 + 0 * 2^2 + 0 * 2^1 + 0 * 2^0$$

**SISTEMA DE NUMERACIÓN EN BASE DOS**

El **sistema binario** o en **base 2** es el utilizado internamente por el computador. Cada dígito se denomina **bit** y sus valores posibles son 0 y 1. Así, se puede asociar (aunque pudiera ser al contrario):

0 = "Ausencia de Tensión"  
1 = "Presencia de Tensión"

Como se vio de manera general en el punto anterior, un número binario estará formado por un conjunto ordenado de bits. El valor de cada posición del número aumenta de derecha a izquierda según potencias de 2. Así, el valor decimal equivalente de las primeras potencias enteras de 2 son las siguientes:

2 <sup>7</sup>	2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>
128	64	32	16	8	4	2	1

La conversión de un número en base 2 a decimal se realiza fácilmente representando el número mediante su polinomio equivalente y operando éste en base 10.

**EJEMPLO 1.2****CONVERTIR NÚMEROS DE BINARIO A DECIMAL**

$$1011_2 = 1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 1 * 2^0 = \mathbf{11}_{10}$$

$$110,11_2 = 1 * 2^2 + 1 * 2^1 + 0 * 2^0 + 1 * 2^{-1} + 1 * 2^{-2} = \mathbf{6,75}_{10}$$

Tal y como se comentó anteriormente, los bits suelen representarse agrupados en paquetes de ocho, formando lo que se conoce como un **byte**.

**Los números con signo**

La forma más habitual para representar los números binarios negativos es escribirlos como si fuesen positivos y añadirle un nuevo símbolo (el signo -) a la izquierda del dígito más significativo. En el sistema de numeración utilizado en Electrónica digital (que es el binario) supondría trabajar con tres niveles lógicos de tensión: uno para el **0**, otro para el **1** y otro para el signo (-), lo que sin duda complicaría los sistemas digitales. Para evitar el tener que añadir el símbolo (-), cuando se tienen que codificar tanto números positivos como negativos, se usa un bit para indicar el signo del número. Existen varios tipos de codificación del signo; el más sencillo de todos es añadir un bit extra a la izquierda del número en valor absoluto: si el bit es **1**, el número es negativo; si es **0**, el número es positivo.



**EJEMPLO 1.3**

-3 → 1011	-4 → 1100	-7 → 1111
+3 → 0011	+4 → 0100	+7 → 0111

**SISTEMAS OCTAL Y HEXADECIMAL**

El sistema **octal**, o en **base 8**, utiliza los dígitos del 1 al 7.

El sistema **hexadecimal**, o en **base 16**, supone la utilización de 16 dígitos con valores equivalentes a los decimales del 0 al 15; como no se dispone más que de 10 guarismos arábigos, se utilizan a continuación las letras de la A a la F.

**1.5.2 CONVERSIONES ENTRE SISTEMAS DE NUMERACIÓN**

Si se quieren realizar operaciones con números en distintas bases, suele ser necesario hacer cambios de base, esto es, dado un  $N_{b_1}$ , queremos obtener  $N_{b_2}$ .

Para llevar a cabo este cambio, existen distintos métodos, según sea el par de bases de trabajo. Como método general se utiliza el siguiente:

$$N_{b_1} \rightarrow N_{10}$$

$$N_{10} \rightarrow N_{b_2}$$

El primer paso es muy sencillo, simplemente hay que descomponer el número en su representación polinomial y sumar sus elementos.

**EJEMPLO 1.4****CONVERTIR A DECIMAL**

$$5.562,7_8 = 5 * 8^3 + 5 * 8^2 + 6 * 8^1 + 2 * 8^0 + 7 * 8^{-1} = \mathbf{2.930,125}_{10}$$

$$11.001,1_2 = 1 * 2^4 + 1 * 2^3 + 0 * 2^2 + 0 * 2^1 + 1 * 2^0 + 1 * 2^{-1} = \mathbf{25,5}_{10}$$

Para realizar el segundo paso de la conversión, se trata de forma separada la parte fraccionaria y la parte entera del número en base 10 a convertir.

- **Parte entera.** Para realizar la conversión de la parte entera, es necesario calcular los coeficientes sobre la nueva base **b2**.

Para ello se divide el número en base 10 por la base **b2** sucesivamente, obteniendo los coeficientes del resto de cada división hasta que se obtenga un cociente menor que **b2**; ese cociente será el último coeficiente.

## EJEMPLO 1.5

CONVERTIR EL NÚMERO  $5.270_{10}$  A BASE 8

División	Cociente	Resto (coeficiente)
$5.270 : 8$	658	$a_0 = 7$
$658 : 8$	82	$a_1 = 2$
$82 : 8$	10	$a_2 = 2$
$10 : 8$	1	$a_3 = 2$
		$a_4 = 1$

Resultado:  $5.270_{10} = 12.227_8$

## EJEMPLO 1.6

CONVERTIR EL NÚMERO  $517_{10}$  A BASE 2

División	Cociente	Resto (coeficiente)
$517 : 2$	258	$a_0 = 1$
$258 : 2$	129	$a_1 = 0$
$129 : 2$	64	$a_2 = 1$
$64 : 2$	32	$a_3 = 0$
$32 : 2$	16	$a_4 = 0$
$16 : 2$	8	$a_5 = 0$
$8 : 2$	4	$a_6 = 0$
$4 : 2$	2	$a_7 = 0$
$2 : 2$	1	$a_8 = 0$
		$a_9 = 1$

Resultado:  $517_{10} = 1.000.000.101_2$

- **Parte fraccionaria.** Abordaremos ahora la conversión de la parte fraccionaria de un número en base 10.

El proceso es similar al anterior, pero multiplicando la parte fraccionaria por la nueva base **b2**; del resultado obtenido en cada multiplicación, la parte entera será el coeficiente y la parte fraccionaria se multiplicará de nuevo por **b2**.

**EJEMPLO 1.7**

**CONVERTIR EL NÚMERO 0,543<sub>10</sub> A BASE 8**

Multiplicación	Entero (coeficiente)	Resto
0,543 * 8 = 4,344	a <sub>-1</sub> = 4	0,344
0,344 * 8 = 2,752	a <sub>-2</sub> = 2	0,752
0,752 * 8 = 6,016	a <sub>-3</sub> = 6	0,016
0,016 * 8 = 0,128	a <sub>-4</sub> = 0	0,128
0,128 * 8 = 1,024	a <sub>-5</sub> = 1	0,024
		...

Resultado: **0,543<sub>10</sub> = 0,42601...<sub>8</sub>**

Cuando se utilizan los sistemas binarios, octal y hexadecimal, puede emplearse el método general de conversión entre bases, aunque existen métodos más rápidos que no requieren el paso intermedio por base 10, ya que sus bases son potencias de 2. De esta forma, a cada dígito octal le corresponden tres dígitos binarios y a cada dígito hexadecimal, cuatro binarios.

<b>Hexadecimal</b>	2	8	A
<b>Binario</b>	1 0 1 0 0 0 1 0 1 0		
<b>Octal</b>	1	2	1 2

**EJEMPLO 1.8**

**CONVERSIÓN ENTRE BASES**

FA3F<sub>16</sub> = 1111 1010 0011 1111<sub>2</sub>

3.456<sub>8</sub> = 011 100 101 110<sub>2</sub>

**ACTIVIDADES 1.5**



- Convierta a binario los siguientes números: 525<sub>10</sub>, 372<sub>8</sub> y 37F<sub>16</sub>
- Pase a decimal las siguientes cantidades: 111.000<sub>2</sub>, 372<sub>8</sub> y 37F<sub>16</sub>

### 1.5.3 CÓDIGOS ALFANUMÉRICOS

Son aquellos que sirven para representar caracteres alfanuméricos: cifras decimales, letras, caracteres especiales y caracteres de control.

Uno de los códigos más utilizados para este fin es el código **ASCII** (*American Standard Code for Information Interchange*), que utiliza 7 bits para la codificación de letras mayúsculas, minúsculas, cifras numéricas y una serie de signos de puntuación y símbolos gráficos.

Actualmente, se utiliza el código ASCII extendido de 8 bits que codifica todo el alfabeto internacional y permite la compatibilidad con los registros de 8 bits dentro de los ordenadores, además de servir para detectar posibles errores de transmisión.

**Tabla 1.2** Códigos ASCII estándar

*	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	TAB	LF	VT	FF	CR	SO	SI
1	DEL	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	P	q	r	s	t	u	v	w	x	y	z	{		}	~	

Posteriormente, han aparecido nuevos sistemas de codificación que permiten representar un mayor número de símbolos alfanuméricos empleando un mayor número de bits en la codificación. Se debe destacar el código *Unicode*, que emplea 16 bits en la codificación.

## ACTIVIDADES 1.6



➤ Codifique la palabra "Hola" en ASCII.

## 1.6 TIPOS DE SOFTWARE

Se define software como el “conjunto de instrucciones o programas usados por una computadora para hacer una determinada tarea”. Tienen carácter virtual, intangible (al contrario que el hardware) y están almacenadas en los diferentes sistemas de almacenamiento.

Para comprender la relación entre hardware y software, es necesario tener una visión global de la estructura de un ordenador; el sistema completo está formado, a su vez, por subsistemas relacionados entre sí de forma escalonada. En los niveles inferiores, la frontera entre el hardware y el software es bastante difusa (Figura 1.4).

- **Nivel 0 - Nivel de Lógica Digital.** Se corresponde con el hardware real de la máquina (dispositivos y circuitos electrónicos).
- **Nivel 1 - Microprogramación.** Son los comandos que intercambian entre sí los circuitos electrónicos. En este nivel, se encuentran los microprogramas, cuya tarea consiste en interpretar las instrucciones de nivel superior. Podemos considerar este nivel como la frontera entre el hardware y el software.
- **Nivel 2 - Lenguaje máquina.** Es el nivel inferior accesible por el usuario y se corresponde con el conjunto de instrucciones que forman el lenguaje directamente interpretable por el hardware, los modos de direccionamiento, los tipos de datos, organización del subsistema de memoria, etc.
- **Nivel 3 - Sistema operativo.** Es el conjunto de programas que proporcionan facilidades a los niveles superiores en la gestión de los recursos del sistema. De esta manera, se crea un entorno favorable para que el usuario interactúe con los niveles inferiores de la máquina.
- **Nivel 4 - Lenguajes de alto nivel.** Son lenguajes de programación con alta abstracción respecto al hardware. El programador encuentra un entorno amigable para la codificación de algoritmos.
- **Nivel 5 - Nivel de aplicación.** Es el nivel más alejado de la realidad física en el cual el usuario no tiene en cuenta los niveles inferiores.

### 1.6.1 LOS LENGUAJES DE PROGRAMACIÓN

Antes de hablar de los distintos lenguajes de programación, conviene dejar claros varios conceptos:

- **Instrucciones.** Son las distintas órdenes que recibe la computadora para operar con o sobre datos.
- **Programa.** Un programa consiste en una cadena de instrucciones y datos. Estas instrucciones son normalmente ejecutadas en secuencia, con eventuales cambios de flujo causados por el propio programa o eventos externos.
- **Lenguaje de programación.** Es el conjunto de símbolos y reglas empleados para codificar las instrucciones que componen cada programa.

El lenguaje que entienden las computadoras es el **lenguaje máquina**, en el que las instrucciones están formadas por dos campos:

- **Código de operación.** Indica a la máquina la operación a realizar.
- **Operandos.** Indica la dirección de memoria donde se encuentran los operandos o el valor de dichos operandos directamente.

El lenguaje máquina está totalmente ligado al hardware en donde se vaya a ejecutar, es decir, un programa en lenguaje máquina realizado para un determinado procesador no funcionará en otro distinto.

Además, el lenguaje máquina solo nos permite realizar operaciones muy simples, tales como saltos, sumas y restas, lo que implica una programación demasiado laboriosa para implementar cualquier algoritmo, siendo necesario, además, conocer las peculiaridades del hardware que se pretende manejar.

Para evitar estos problemas y facilitar la tarea de programación, surgen los **lenguajes de alto nivel**, que no dependen de la plataforma hardware en la que se vayan a ejecutar y, además, presentan instrucciones más complejas y potentes.

Ejemplos de lenguajes de alto nivel: BASIC, Visual BASIC, C, C++ y Visual C.

La utilización de lenguajes de alto nivel facilita enormemente la tarea de escribir programas pero, para que esas instrucciones sean comprensibles para el procesador, deben ser convertidas antes a lenguaje máquina. Esa conversión se realiza mediante programas traductores, llamados compiladores.

A lo que escribe el programador se le denomina “código fuente”. Al resultado de la “conversión” (compilación) en lenguaje máquina, se le denomina “código objeto”, “binarios” o “archivos ejecutables”.

---

## 1.6.2 LAS APLICACIONES INFORMÁTICAS

Son programas diseñados para resolver las necesidades habituales de los usuarios. Suelen disponer de interfaces que agilizan y facilitan el trabajo al usuario:

- ✓ Gestores de bases de datos.
- ✓ Hojas de cálculo.
- ✓ Procesadores de texto.
- ✓ Correo electrónico.
- ✓ Creación de presentaciones.
- ✓ Diseño.
- ✓ Tratamiento fotográfico.
- ✓ Etc.

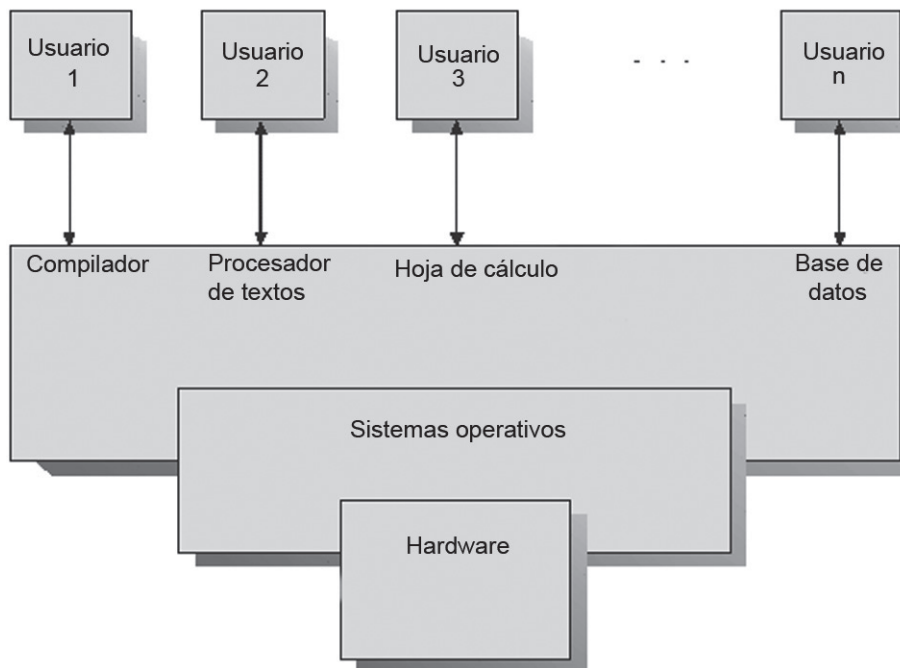
Las empresas desarrolladoras de software suelen comercializar paquetes con las aplicaciones más utilizadas por los usuarios como, por ejemplo, Microsoft Office o Lotus SmartSuite.

## 1.7 EL SISTEMA OPERATIVO

Un **sistema operativo** es un programa o conjunto de programas que actúa como intermediario entre el usuario y el hardware del ordenador, gestionando los recursos del sistema y optimizando su uso.

El sistema operativo es en sí mismo un programa, pero un programa muy especial y quizás el más complejo e importante. Cuando se conecta un ordenador se carga parte del sistema operativo en la memoria y se ejecuta. El sistema operativo *despierta* al ordenador y hace que reconozca a la CPU, la memoria, las unidades de disco y cualquier otro dispositivo conectado a ella, como el teclado, el ratón, la impresora, etc., verificando así que no existan errores de conexión y que todos los dispositivos se han reconocido y trabajan correctamente. A este primer diagnóstico se le denomina **POST**.

El sistema operativo presenta al usuario la máquina de una forma más fácil de manejar y programar que el hardware que está por debajo, es decir, un usuario normal, simplemente abre los ficheros que grabó en un disco, sin preocuparse por la disposición de los bits en el medio físico, los tiempos de espera del motor del disco, la posición de un cabezal, el acceso de otros usuarios, etc.



*Figura 1.5. El SO como intermediario entre los usuarios y el hardware*

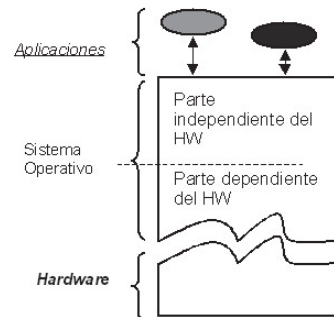


### 1.7.1 FUNCIONES DEL SISTEMA OPERATIVO

A continuación se muestran las funciones principales que realiza todo sistema operativo:

- **Control de la ejecución de los programas.** Para ello, acepta los trabajos, administra la manera en que se realizan, les asigna los recursos y los conserva hasta su finalización.
- **Administración de periféricos.** Coordinando y manipulando los dispositivos conectados al ordenador.
- **Gestión de permisos y de usuarios.** Adjudica los permisos de acceso a los usuarios y evita que las acciones de uno afecten el trabajo que está realizando otro.
- **Control de concurrencia.** Establece prioridades cuando diferentes procesos solicitan el mismo recurso.
- **Control de errores.** Gestiona los errores de hardware y la pérdida de datos.
- **Administración de memoria.** Asigna memoria a los procesos y gestiona su uso.
- **Control de seguridad.** Debe proporcionar seguridad tanto para los usuarios como para el software y la información almacenada en los sistemas.

En concordancia con estas funciones principales, es posible analizar la estructura de un sistema operativo en cinco niveles. Los primeros dos niveles entrarían dentro de la parte del sistema operativo dependiente del hardware, el resto de los niveles pertenecen a la parte portable del mismo (ver Figura 1.6).

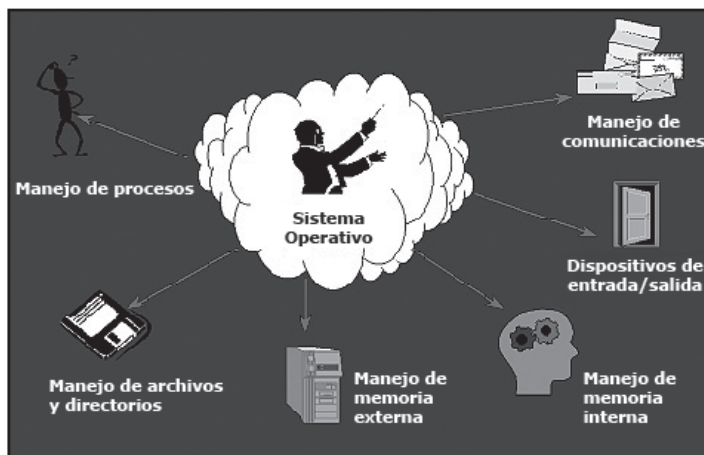


*Figura 1.6. Esquema de relación entre partes software y hardware*

Cada uno de los niveles se comunica con el inmediatamente inferior y superior coordinando sus funciones.

- **Nivel 1 - Gestión del procesador.** En este nivel se encuentra la parte del sistema operativo encargada de la gestión de la CPU. En los sistemas operativos multiproceso (es decir, que pueden ejecutar varios procesos a la vez), este nivel se encarga de compartir la CPU entre los distintos procesos realizando funciones de sincronización, conmutación de la CPU y gestión de interrupciones.
- **Nivel 2 - Gestión de memoria.** Este nivel es el encargado de repartir la memoria disponible entre los procesos. Se realizan funciones de asignación y liberación de memoria, y el control de violación de acceso a zonas de memoria no permitidas.
- **Nivel 3 - Gestión de procesos.** Este nivel es el encargado de la creación y destrucción de los procesos, intercambio de mensajes y detección y arranque de los mismos.

- **Nivel 4 - Gestión de dispositivos.** En este nivel se realiza la gestión de las entradas/salidas (E/S) en función de los dispositivos existentes. Entre otras, se encarga de las funciones de creación de procesos de E/S, asignación y liberación de dispositivos E/S, y planificación de la E/S.
- **Nivel 5 - Gestión de la información.** El objetivo de este nivel es el de gestionar el espacio de nombres lógicos, utilizados para simplificar el acceso a los recursos, ya que mediante estos se sustituyen rutas de acceso que pueden ser muy largas y difíciles de recordar por un solo nombre, encargándose el sistema operativo, de forma totalmente transparente para el usuario, de realizar esa búsqueda de ruta. Otro de sus cometidos es la protección de la información realizando funciones de creación y destrucción de ficheros y directorios, apertura y cierre de ficheros, lectura y escritura de ficheros y protección de acceso.



*Figura 1.7. El SO como director de operaciones sobre el hardware*

Es importante destacar que un mismo sistema operativo puede trabajar en múltiples plataformas hardware, por lo que debe poder adaptarse a las peculiaridades de cada una de ellas.

Imagínese un programador que pretenda realizar una aplicación de gestión de archivos, teóricamente debería conocer las peculiaridades del hardware donde correrá su aplicación a la hora de manipular archivos, pero gracias a la existencia del sistema operativo el programador puede abstraerse de las peculiaridades de la máquina y su aplicación funcionará correctamente, independientemente del hardware que esté por debajo.

## 1.7.2 EL KERNEL

El **Kernel** es el núcleo del sistema operativo. Cuando arranca el ordenador, se carga en memoria y permanece allí, realizando funciones básicas como:

- ✓ Manejo de la memoria.
- ✓ Comunicación entre procesos.
- ✓ Control de interrupciones.
- ✓ Manejo de condiciones de error.
- ✓ Traslado de control de un proceso a otro.
- ✓ Control de periféricos.

### 1.7.3 PROGRAMAS Y PROCESOS

Un **proceso** es un programa en ejecución. Cada proceso se compone de un código que se ejecuta y una estructura de datos, estando ambos cargados en memoria.

No hay que confundir *procesos* con *archivos* o *programas*. Por ejemplo, un compilador C no es un proceso, pero un compilador C ejecutándose será un proceso para el sistema operativo y, por tanto, le asignará recursos (CPU, memoria, etc.) y controlará su ejecución.

La estructura de datos sirve para identificar unívocamente cada proceso y controlar todos los aspectos de su ejecución. Cada estructura contiene básicamente los siguientes datos:

- **Estado actual del proceso:** puede ser *En ejecución*, *Esperando*, *Detenido*, etc.
- **Identificación unívoca del proceso:** a cada proceso se le asigna un PID (código identificador de proceso) que se utilizará para identificarlo ante el sistema operativo.
- **Prioridad del proceso:** cada proceso tiene asignada una prioridad de forma que, en cualquier instante, el proceso que mayor prioridad tiene asignada, de entre los que están en estado *Esperando*, es el que se ejecutará.
- **Zona de memoria asignada:** cada proceso debe tener una zona de memoria independiente que no puede ser interferida por otros procesos.
- **Recursos asociados al proceso:** como ficheros, semáforos, etc.

### PLANIFICACIÓN DE CPU-MULTIPROCESAMIENTO

Existen ordenadores que disponen de varios procesadores, por lo que son capaces de ejecutar al mismo tiempo varias tareas, estos computadores disponen de sistemas operativos especiales denominados **Sistemas multiprocesador**. En los ordenadores convencionales, normalmente solo se dispone de un procesador, por lo que en cada momento solo se estará procesando una tarea (actualmente, los procesadores disponen de varios núcleos que, en determinadas actuaciones, pueden actuar como si fueran varios procesadores). No obstante, es posible crear la ilusión de multiprocesamiento por medio de interrupciones, cambiando el proceso activo cada cierto tiempo aunque no se haya finalizado la tarea en ejecución.

El sistema operativo se encarga de compartir la CPU entre los distintos procesos de la manera más eficiente posible, persiguiendo los siguientes objetivos:

- **Equidad.**
- **Eficiencia.**
- **Bajo tiempo de respuesta.**
- **Rendimiento alto.**
- **Minimización del tiempo de espera.**

Desgraciadamente, todos estos objetivos no se pueden conseguir simultáneamente.