

## 2.1 ESTRUCTURA DE UN SISTEMA INFORMÁTICO

La estructura de un sistema informático se podría definir como un conjunto determinado de reglas, normas y procedimientos que especifican las interrelaciones que deben existir entre los componentes de un sistema informático y las características que deben cumplir cada uno de estos componentes.

### 2.1.1 ESTRUCTURA MONOLÍTICA

Es la estructura de los primeros sistemas operativos constituidos fundamentalmente por un único programa compuesto de un conjunto de rutinas entrelazadas de tal forma que cada una puede llamar a cualquier otra. Las características fundamentales de este tipo de estructura son:

- Tiene una estructura interna indefinida y sus niveles de funcionalidad no están bien separados.
- Tiene difícil configuración y actualización, así como falta de protecciones y privilegios al entrar en rutinas que manejan diferentes aspectos de los recursos de la computadora, como memoria, disco, etc.
- Dispone de una buena definición de los parámetros de enlace entre las distintas rutinas existentes.
- Suelen estar hechos a medida, por lo que son eficientes y rápidos en su ejecución y gestión, pero carecen de flexibilidad para soportar diferentes ambientes de trabajo o tipos de aplicaciones.

### 2.1.2 ESTRUCTURA JERÁRQUICA

A medida que fueron creciendo las necesidades de los usuarios y se perfeccionaron los sistemas, se hizo necesaria una mayor organización del software, donde una parte del sistema contenía subpartes y estaba organizado en forma de niveles.

Se dividió el sistema operativo en pequeños módulos, de forma que cada uno de ellos estuviera perfectamente definido y con una clara interfaz con el resto de elementos.

El módulo de cada nivel funciona utilizando los servicios del nivel inferior.

De esta forma se facilita la protección y el acceso al sistema.

En esta estructura se basan la mayoría de los sistemas operativos actuales.

### 2.1.3 ESTRUCTURA EN ANILLOS

Esta estructura es una evolución de la anterior.

El sistema está organizado en anillos concéntricos o **rings**. Cada anillo tiene una apertura, conocida como puerta o trampa (*trap*), por donde pueden entrar las llamadas de las capas inferiores. De esta forma, las zonas más internas del sistema operativo o núcleo del sistema estarán más protegidos de accesos indeseados desde las capas más externas. Las capas más internas serán, por tanto, más privilegiadas que las externas.

## 2.1.4 MÁQUINA VIRTUAL

Es una estructura que presenta una interfaz que muestra una máquina que parece idéntica a una máquina real.

Estos sistemas operativos separan dos conceptos que suelen estar unidos en el resto de sistemas: la **multiprogramación** (se denomina así a la técnica que permite que dos o más procesos ocupen la misma unidad de memoria principal) y la **máquina extendida**.



En toda computadora se pueden definir dos máquinas abstractas:

- La **máquina desnuda** o **simple**, que está definida por el hardware. En ella, todas las operaciones son las instrucciones del lenguaje máquina.
- La **máquina extendida**, definida por el hardware y el núcleo del sistema operativo. Es la que generalmente utiliza el usuario.

El núcleo de estos sistemas operativos se denomina monitor virtual y tiene como misión llevar a cabo la multiprogramación, presentando a los niveles superiores tantas máquinas virtuales como se soliciten.

Estas máquinas virtuales no son máquinas extendidas, sino una réplica de la máquina real, de manera que en cada una de ellas se puede ejecutar un sistema operativo diferente, que será el que ofrezca la máquina extendida al usuario.

## 2.1.5 ESTRUCTURA CLIENTE-SERVIDOR

El tipo más reciente de sistemas operativos es el denominado cliente-servidor, que puede ser ejecutado en la mayoría de las computadoras, ya sean grandes o pequeñas.

Esta estructura es altamente modular y los módulos del sistema no tienen acceso al hardware. Sirve para toda clase de aplicaciones; por tanto, es de propósito general y cumple con las mismas actividades que los sistemas operativos convencionales.

El núcleo tiene como misión establecer la comunicación entre los clientes y los servidores. Los procesos pueden ser tanto servidores como clientes.

Los servicios se efectúan mediante la técnica de *message passing*.

Dicha técnica actúa de la manera siguiente:

1. El proceso cliente solicita al núcleo un servicio mediante un mensaje.
2. El núcleo recibe el mensaje, toma las decisiones de planificación y envía el mensaje al proceso servidor.
3. El proceso servidor ejecuta la función solicitada y devuelve al núcleo un mensaje con el resultado de la operación.
4. El núcleo reenvía el mensaje al proceso cliente indicando que el servicio se ha cumplido.

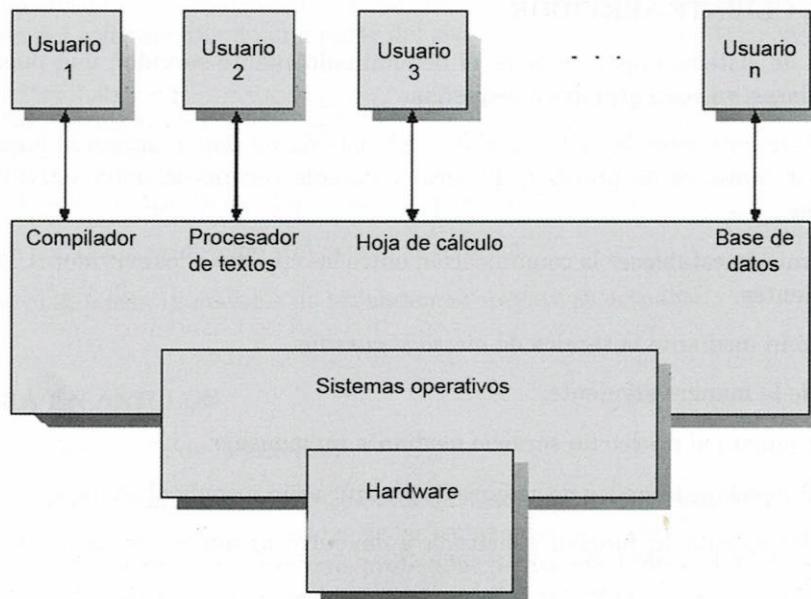
Por ejemplo, un programa de aplicación normal es un cliente que llama al servidor correspondiente para acceder a un archivo o realizar una operación de entrada/salida sobre un dispositivo concreto. A su vez, un proceso cliente puede actuar como servidor para otro. Este funcionamiento ofrece gran flexibilidad, ya que el núcleo proporciona solamente funciones muy básicas de memoria, entrada/salida, archivos y procesos, dejando a los servidores la mayoría de los servicios que el usuario final puede usar. Estos servidores han de tener mecanismos de seguridad y protección que, a su vez, serán filtrados por el núcleo que controla el hardware.

## 2.2 ARQUITECTURA DE UN 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.



## 2.2.1 EVOLUCIÓN DE LOS SISTEMAS OPERATIVOS

Los sistemas operativos han ido evolucionando a lo largo de los años. A continuación, se presenta las distintas arquitecturas que han ido desarrollándose.

### 2.2.1.1 Proceso en serie

En los primeros computadores, desde finales de los años 40 hasta mediados de los 50, el programador interactuaba directamente con el hardware; no había sistema operativo. La operación con estas máquinas se efectuaba desde una consola consistente en unos indicadores luminosos, unos commutadores, algún dispositivo de entrada y una impresora. Los programas en código de máquina se cargaban a través del dispositivo de entrada. Si se detenía el programa por un error, la condición de error se indicaba mediante los indicadores luminosos. El programador podía examinar los registros y la memoria principal para determinar la causa del error. Si el programa continuaba hasta su culminación normal, la salida aparecería en la impresora.

Este modo de operación podría denominarse **proceso en serie** porque refleja el hecho de que los usuarios tenían que acceder al computador uno detrás de otro.

Estos sistemas presentaban dos problemas principales:

- **Planificación:** la mayoría de las instalaciones empleaban un formulario de reserva de tiempo de máquina. Un usuario podía reservar una hora y terminar a los 45 minutos (produciendo un desperdicio del tiempo de computador) o no alcanzarle el tiempo reservado (dejando trabajos inconclusos).
- **Tiempo de preparación:** un programa sencillo cargaba un compilador y un programa fuente, guardaba el programa compilado (programa objeto), lo montaba y, después, cargaba el programa objeto junto con las funciones comunes. Cada uno de estos pasos podía implicar montar y desmontar cintas o preparar paquetes de tarjetas.

### 2.2.1.2 Sistemas por lotes

Para mejorar el uso del ordenador, se desarrolló el concepto de **sistema operativo por lotes (batch)**. El primer sistema operativo con esta arquitectura fue desarrollado a mediados de los 50 por General Motors.

Está basado en el uso de un elemento software conocido como **monitor**. Para entender cómo funciona este esquema, se va a ver desde dos puntos de vista: el del monitor y el del procesador.

- Desde el punto de vista del monitor, él es quien controla la secuencia de sucesos. Para que esto sea posible, gran parte del monitor deberá estar siempre en la memoria principal y disponible para su ejecución. Esta parte del monitor se conoce como **monitor residente**. El resto del monitor consta de utilidades y funciones comunes que se cargan como subrutinas en los programas de los usuarios al comienzo de cualquier trabajo que las necesite.

El monitor lee los trabajos del dispositivo de entrada y, a medida que los lee, el trabajo actual se ubica en la zona del programa de usuario y toma el control. Cuando el trabajo termina, se devuelve el control al monitor, quien lee el siguiente trabajo. Los resultados de cada trabajo se imprimen y entregan al usuario.

- Desde el punto de vista del procesador, él es quien ejecuta las instrucciones del programa de usuario hasta que encuentre una condición de finalización o de error, tras lo cual va en búsqueda de la instrucción siguiente del programa monitor.

Debe quedar claro que es el monitor el que gestiona el problema de la planificación. Con respecto a la preparación de los trabajos, también el monitor se encarga de ello. Con cada trabajo se incluyen instrucciones de una forma primitiva de lenguaje de control de trabajos (JCL - *Job Control Language*), que es un tipo especial de lenguaje de programación empleado para dar instrucciones al monitor.

Durante la ejecución del programa de usuario, cada instrucción de entrada origina la lectura de unos datos. La instrucción de entrada en el programa del usuario hace que se invoque una rutina de entrada, que forma parte del sistema operativo. La rutina de entrada se asegura de que el programa de usuario no ha leído accidentalmente los datos JCL; si esto sucede, se produce un error y el control se transfiere al monitor.

Los sistemas sencillos por lotes también poseen las siguientes características:

- **Protección de memoria:** mientras el programa del usuario esté ejecutándose, no debe modificarse la zona de memoria en la que está el monitor. Si se hace, el hardware del procesador deberá detectar el error y transferir el control al monitor, quien abortará el trabajo, mostrará el mensaje de error y cargará el siguiente trabajo.
- **Uso de temporizador:** impide que un solo trabajo monopolice el sistema. El temporizador se carga al comenzar cada trabajo y, si expira el tiempo, se producirá una interrupción y el control volverá al monitor.
- **Instrucciones privilegiadas:** ciertas instrucciones son designadas como privilegiadas y pueden ser ejecutadas solo por el monitor. Si el procesador, al ejecutar el programa del usuario, encuentra una instrucción, se producirá una interrupción de error.
- El **tiempo de máquina:** se reparte entre la ejecución de programas de usuario y la ejecución del monitor. Así se tienen dos pérdidas; se entrega al monitor cierta cantidad de memoria principal y éste consume cierto tiempo de máquina.

#### 2.2.1.3 Sistemas por lotes con multiprogramación

Aún con el secuenciamiento automático de los trabajos ofrecido por un sistema operativo por lotes, el procesador está desocupado a menudo. El problema es que los dispositivos de E/S son lentos comparados con el procesador.

Ahora bien, si hay memoria principal suficiente para el sistema operativo y dos programas de usuario, cuando un trabajo necesite esperar una E/S, el procesador puede cambiar al otro trabajo que probablemente no estará esperando a la E/S. Además, se podría ampliar la memoria para almacenar tres, cuatro o más programas y commutar entre todos ellos. Este proceso es conocido como **multiprogramación** o **multitarea**. Éste es el punto central de los sistemas operativos modernos.

#### 2.2.1.4 Sistemas de tiempo compartido

Con el uso de la multiprogramación, el tratamiento por lotes puede llegar a ser bastante eficiente. Sin embargo, para muchas tareas, es conveniente suministrar un modo en que el usuario interactúe directamente con el computador.

Al igual que la multiprogramación permite al procesador manejar varias tareas por lotes al mismo tiempo, la multiprogramación puede también utilizarse para manejar varias tareas interactivas. En este último caso, la técnica se conoce como **tiempo compartido**, porque refleja el hecho de que el tiempo del procesador está compartido entre los diversos usuarios. La técnica básica de un sistema de tiempo compartido es tener a varios usuarios utilizando simultáneamente el sistema mediante terminales, mientras que el sistema operativo intercala la ejecución de cada programa de usuario en ráfagas cortas de cómputo (*quantum*). De esta manera, si hay  $n$  usuarios que solicitan el servicio a la vez, cada usuario solo dispondrá, en promedio, de  $1/n$  de la atención efectiva del computador, sin contar con la sobrecarga del sistema operativo.

#### 2.2.1.5 Sistemas distribuidos

Un sistema de procesamiento de datos en que todas las funciones están centralizadas en una CPU y en un sistema operativo se llama **procesamiento centralizado**.

En cambio, un **procesamiento distribuido** es aquel en el que se ejecutan los datos en distintos nodos, dispersos geográficamente, interconectados mediante una red.

Las características principales de un sistema distribuido son:

- Fragmentación de los elementos que componen una aplicación, en dos o más sistemas interconectados, de igual o diferente arquitectura operativa.
  - Los recursos de los sistemas se controlan y administran en forma independiente
  - La relación entre ambos sistemas puede tener diferentes formas: arquitectura cliente/servidor o punto a punto (ambos nodos ofrecen los mismos servicios).
- 

## 2.3 FUNCIONES DE UN SISTEMA OPERATIVO

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

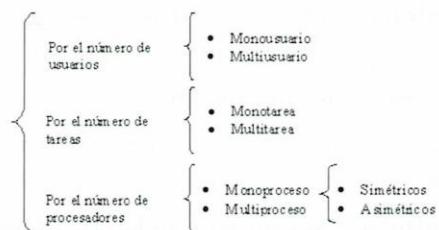
- **Control de la ejecución de los programas:** para ello, acepta los trabajos, administra la manera en que se realizan, les asigna los recursos y los conserva hasta su finalización.
  - **Administración de periféricos:** coordinando y manipulando los dispositivos conectados al ordenador.
  - **Gestión 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.
- 

## 2.4 CLASIFICACIONES DE LOS SISTEMAS OPERATIVOS

Según la perspectiva con la que se observen los sistemas operativos, pueden realizarse múltiples clasificaciones. Entre ellas se pueden incluir las siguientes:

## 2.4.1 POR LOS SERVICIOS OFRECIDOS

En esta clasificación se tiene en cuenta la visión del usuario final y puede ser la siguiente:



Los sistemas operativos **monousuario** son aquellos que únicamente soportan un usuario a la vez, sin importar las características de la máquina sobre la que está montado el sistema.

Los sistemas operativos **multiusuario** son capaces de dar servicio a más de un usuario a la vez, también independientemente de la plataforma hardware sobre la que esté montado el sistema.

Los sistemas **monotarea** son aquellos que solo permiten una tarea a la vez por usuario. Puede darse el caso de un sistema multiusuario y monotarea, en el cual se admiten varios usuarios al mismo tiempo, pero cada uno de ellos puede estar haciendo solo una tarea a la vez.

Un sistema operativo **multitarea** es aquel que permite al usuario estar realizando varios trabajos al mismo tiempo. Es común encontrar en ellos interfaces gráficas orientadas al uso de menús y al ratón, lo que permite un rápido intercambio entre las tareas para el usuario, mejorando su productividad.

Los sistemas **monoproceso** son los que únicamente permiten realizar un proceso a la vez. Sin embargo, permiten simular la multitarea haciendo que el sistema realice una tarea rotatoria con intercambio muy rápido.

Los sistemas operativos **multiproceso** son los que permiten realizar varios procesos simultáneamente y, por tanto, son capaces de ejecutar varias tareas al mismo tiempo.

Dentro de los sistemas multiproceso, se encuentran los sistemas **simétricos**, que son los que distribuyen la carga de procesamiento por igual entre todos los procesadores existentes. Sin embargo, los sistemas multiproceso **asimétricos**, como Windows NT, asignan una tarea por procesador existente, según su prioridad, y el resto de tareas (de baja prioridad) se ejecutan en un único procesador. Por ejemplo, un sistema biprocesador asimétrico ejecutaría una sola tarea en un procesador y el resto en el otro.

## ACTIVIDADES 2.1



- Indique dos sistemas operativos monousuario y otros dos multiusuario.
- Indique dos sistemas operativos monotarea y otros dos multitarea.
- Indique dos sistemas operativos monoproceso y otros dos multiproceso.

## 2.4.2 POR LA FORMA DE OFRECER LOS SERVICIOS

En esta clasificación se encuentran:

- **Sistemas centralizados:** hasta que los computadores personales no tuvieron un precio accesible y suficiente potencia, la mayoría de los sistemas utilizaban el modelo de proceso centralizado. Con este tipo de modelo los computadores *mainframe* se encargaban de todo el procesamiento y los usuarios manejaban únicamente terminales *tontos*



## ¿SABÍAS QUE...?

Un terminal tonto era aquel que no disponía de memoria, ni procesador.

Actualmente se siguen utilizando los sistemas centralizados (como los *Terminal Services* de Microsoft) pero los terminales dejan de ser tontos y pueden realizar otras muchas tareas por sí mismos.

- **Sistemas distribuidos:** los sistemas operativos distribuidos son sistemas quasi-independientes que permiten distribuir los trabajos, tareas o procesos entre un conjunto de procesadores. Puede ocurrir que este conjunto de procesadores se encuentren en el mismo equipo o en equipos distintos (siendo en este último caso transparente para el usuario).

Los sistemas operativos distribuidos más extendidos son los siguientes: *Sprite*, *Solaris-MC*, *Mach*, *Chorus*, *Spring*, *Amoeba*, *Taos*, etc.

- **Sistemas operativos en red:** estos sistemas operativos son aquellos que mantienen a dos o más computadoras unidas a través de algún medio de comunicación (físico o no), con el objetivo primordial de poder compartir los diferentes recursos y la información del sistema. En este entorno, cada computador mantiene su propio sistema operativo y su propio sistema de archivos local.

Los sistemas operativos de red usados más ampliamente son: *Novell NetWare*, *Windows Server*, *Linux Server*, etc.

- **Sistemas operativos de escritorio:** estos sistemas operativos son los que se utilizan en los equipos de sobremesa, estaciones de trabajo o portátiles. También se les puede denominar como **sistemas operativos cliente**. Entre ellos se encuentran: *Windows XP Professional*, *Windows Vista*, *Windows 7* y *Linux*.

### 2.4.3 POR SU DISPONIBILIDAD

En esta clasificación se encuentran:

- **Sistemas operativos propietarios:** son aquellos que son propiedad intelectual de alguna empresa. Esto implica que se necesitan **licencias** de uso para que el usuario ejecute el software y no se dispone de acceso a su código fuente o, aun teniendo acceso a él, no se tiene derecho a modificarlo ni distribuirlo. En este grupo se encuentra Windows.
- **Sistemas operativos libres:** son aquellos que garantizan las cuatro libertades del software (según Richard M. Stallman):
  1. La libertad de usar el programa con cualquier propósito.
  2. La libertad de estudiar cómo funciona el programa y modificarlo, adaptándolo a las necesidades que tuviera el usuario.

3. La libertad de distribuir copias del programa, con lo que se puede ayudar a otros usuarios.
4. La libertad de mejorar el programa y hacer públicas dichas mejoras a otros usuarios, de modo que toda la comunidad se beneficie de ello.

Las libertades 1 y 3 requieren acceso al código fuente para estudiar y modificar dicho software, por lo que al final el software libre es también **software de código abierto**.

El **software libre** suele estar disponible gratuitamente o al precio de coste de la distribución a través de otros medios; sin embargo no es obligatorio que sea así, por lo tanto, no hay que asociar software libre a **software gratuito**, ya que, conservando su carácter de libre, podrá ser distribuido comercialmente (**software comercial**).

De la misma manera, el software gratuito puede incluir el código fuente, pero eso no quiere decir que se pueda considerar como “libre” a no ser que se garanticen los derechos de modificación y redistribución de las versiones modificadas del programa.

Tampoco debe confundirse software libre con **software de dominio público**. Este último es aquel que no requiere de licencia pues sus derechos de explotación pertenecen a todos por igual y cualquiera puede hacer uso de él, siempre con fines legales y consignando su autoría original.

## ACTIVIDADES 2.2



- Indique de qué tipo es el sistema operativo de su propio ordenador, según todas las clasificaciones.
- Indique el nombre de un sistema operativo libre y propietario distinto al que dispone en su ordenador.

## 2.5 TIPOS DE SOFTWARE

Se define el 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.

- **Nivel 0 - Lógica digital:** se corresponde con el hardware real de la máquina (dispositivos y circuitos electrónicos).
- **Nivel 1 - Microporgramación:** son los comandos que intercambian entre sí los circuitos electrónicos. En este nivel se encuentran los microporogramas, 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 interaccione 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.

### 2.5.1 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.



#### EJEMPLO 2.1

Instrucción lenguaje máquina JMP 03FH.

El primer campo indica salto (JMP); el segundo campo indica la dirección de salto, en este caso, 03F en hexadecimal.

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. Esta 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 “ficheros ejecutables”.

## 2.5.2 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 (Microsoft Access, Oracle).
- ✓ Hojas de cálculo (Microsoft Excel, Lotus 1-2-3).
- ✓ Procesadores de texto (Microsoft Word, OpenOffice Writer).
- ✓ Correo electrónico (Microsoft Outlook, Eudora).
- ✓ Creación de presentaciones (Microsoft Power Point, Corel Presentations).
- ✓ Diseño (AutoCAD, Corel Draw).
- ✓ Tratamiento fotográfico (Corel Photoshop, Google Picasa).
- ✓ 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.

## 2.6 TIPOS DE APLICACIONES

En función del tipo de software, las aplicaciones pueden ser:

- **Gratuitas (freeware) o comerciales:** esta clasificación es interesante en el momento de la planificación, del análisis del entorno y de la adquisición de las aplicaciones, y es fundamental para evaluar los costes. Además tiene fuertes repercusiones en la etapa de mantenimiento del software.
- **Libres o propietarias:**

Las **licencias de software libre** se basan en la distribución del código fuente junto con el programa, así como en las cuatro premisas indicadas en el epígrafe anterior.

Hay que dejar claro de nuevo que el que un determinado programa sea libre no implica en ningún momento que sea o deba ser gratuito (*freeware*). Es perfectamente compatible el que se trate de un software libre y a su vez sea un programa comercial, en el que se pida un pago por licencia.

Las **licencias de software propietario** son aquellas en las que los usuarios tienen limitadas las posibilidades de usarlo, modificarlo o redistribuirlo (con o sin modificaciones).

- Otra clasificación a reseñar es la que divide las aplicaciones en **opensource** (código abierto al usuario) o **privativas** (cuyo código fuente no está disponible o el acceso a él se encuentra restringido).

También aquí la repercusión en las fases de la implantación es importante, ya que una aplicación *opensource* va a permitir un afinamiento más adaptado, barato y simple que una *privativa*. La consecuencia es una mejora en el ciclo de vida de la implantación a un menor coste. Aunque debido a contingencias en la compatibilidad e interconexión de aplicaciones, es necesario habitualmente instalar *software privativo*.

## 2.7 TIPOS DE LICENCIA

En función de las licencias de distribución, el software comercial puede ser:

- **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. Aunque 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 la 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).

- **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 2.3



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

## 2.8 LOS GESTORES DE ARRANQUE

En caso de que haya instalados varios sistemas operativos en un mismo ordenador, hay que utilizar un sistema para poder seleccionar qué sistema operativo se desea iniciar.

El **gestor de arranque** es un pequeño programa que se ejecuta una vez completado el inicio normal de la BIOS y que permite seleccionar el sistema operativo en caso de disponer de arranque múltiple.

Entre los gestores de arranque se encuentran los siguientes:

- **NTLDR (NT Loader)**: es el archivo encargado del arranque de los sistemas operativos: Windows NT, Windows 2000, Windows XP y Windows Server 2003. Se encuentra, normalmente, en el directorio raíz del disco del sistema.

Requiere, como mínimo, que se encuentre en dicho directorio raíz: el archivo **boot.ini**, que contiene el menú de opciones de inicio, y el archivo **NTDETECT.COM**, que es el que se encarga de iniciar el sistema seleccionado.

- El **Administrador de arranque de Windows (Bootmgr)**: es el archivo encargado del arranque de los sistemas operativos: Windows Vista, Windows 7 y Windows Server 2008. Se encuentra, normalmente, en el directorio raíz del disco del sistema.

Controla el proceso de arranque mostrando el menú multiarranque (si hubiera más de un sistema operativo instalado en el disco). Después, llama al archivo **WinLoad.exe** que es el cargador del sistema operativo Windows y dará paso al archivo **ntoskrnl.exe**, que se encargará del resto del arranque del sistema.



### ¿SABÍAS QUE...?

El archivo WinLoad.exe se encuentra en el directorio **\Windows\system32**.

- **Lilo (Linux Loader)**: es un gestor de arranque de Linux que permite iniciar este sistema operativo junto con otras plataformas (como Windows) que haya en el mismo ordenador. Funciona en una variedad de sistemas de archivos y puede arrancar un sistema operativo desde el disco duro o desde un disco flexible externo.
- **Grub**: es un gestor de arranque más moderno y flexible que Lilo, ya que permite que el administrador ejecute cualquier comando desde la línea de comando de Grub. Entre todas sus características hay que destacar la posibilidad de incluir múltiples formatos de ejecutables, el arranque de sistemas operativos no-multiarranque, una agradable interfaz de usuario y una interfaz de línea de comando muy flexible.

### ACTIVIDADES 2.4

- ▶ Averigüe qué gestor de arranque utiliza el sistema operativo de su equipo.
- ▶ Busque información en Internet sobre los cuatro gestores de arranque descritos anteriormente.

## 2.9 GESTIÓN DE VARIOS SISTEMAS OPERATIVOS EN UN ORDENADOR

Hasta hace poco tiempo, la posibilidad de tener en un mismo ordenador dos sistemas operativos gestionados por un gestor de arranque permitía utilizar uno u otro y conseguir un ahorro en el hardware necesario. Sin embargo, esta utilización no era completa ya que no permitía disponer de ambos los sistemas operativos simultáneamente.

Actualmente, con las máquinas virtuales ha desaparecido esa limitación y es posible disponer de varios sistemas operativos instalados en máquinas virtuales de forma simultáneamente e, incluso, interactuando entre ellos de forma que un usuario no es capaz de distinguir entre un sistema operativo instalado en un ordenador y el que está instalado en una máquina virtual.

Para obtener más información sobre las máquinas virtuales, vaya al Capítulo 3.



### RESUMEN DEL CAPÍTULO



En este capítulo se ha indicado las estructuras disponibles de un sistema informático: monolítica, jerárquica, en anillos, máquina virtual y cliente-servidor.

También se ha descrito la arquitectura de un sistema operativo y la evolución que ha sufrido a lo largo de los años: proceso en serie, sistemas por lotes, sistemas por lotes con multiprogramación, sistemas de tiempo compartido y sistemas distribuidos.

Se han desarrollado las funciones principales que realiza un sistema operativo: control de la ejecución de los programas, administración de periféricos, gestión de permisos y de usuarios, control de concurrencia, control de errores, administración de memoria. Asigna memoria a los procesos y gestiona su uso y control de seguridad.

Así mismo, se ha tratado sobre las distintas posibilidades para clasificar los sistemas operativos en función de los servicios ofrecidos, la forma de ofrecerlos y su disponibilidad.

Además, se ha hablado sobre la relación entre el hardware y el software, los tipos de software y los lenguajes de programación.

También se han indicado los tipos de aplicaciones diferenciando entre las gratuitas y las comerciales y las libres y las propietarias, los tipos de licencia y los gestores de arranque, así como la posibilidad de tener dos sistemas operativos en el mismo equipo para utilizarlo de forma alternativa.

Así mismo, se ha hablado sobre la gestión de varios sistemas operativos en un ordenador.



# EJERCICIOS PROPUESTOS

- 1. Comente lo que sepa sobre las distintas estructuras de un sistema informático.
- 2. Comente brevemente las funciones de un sistema operativo.
- 3. Indique las diferencias entre un sistema operativo monousuario y otro multiusuario.
- 4. Indique las diferencias que hay entre una licencia OEM y una *Retail*.
- 5. Indique lo que son los gestores de arranque.



# TEST DE CONOCIMIENTOS

**1** Indique qué afirmación es verdadera:

- a) Todo sistema informático está compuesto por dos elementos básicos (hardware y software).
- b) Hay dos tipos de software (de operación y de aplicación).
- c) El sistema operativo es el conjunto de programas que proporcionan facilidades a los niveles superiores en la gestión de los recursos del sistema.
- d) Todas las anteriores son correctas.

**2** Indique qué afirmación es verdadera:

- a) El sistema operativo se corresponde con el hardware.
- b) Los lenguajes de alto nivel se encuentran en un nivel superior que los lenguajes de aplicación.
- c) El lenguaje máquina es un lenguaje de alto nivel.
- d) Todas las anteriores son falsas.

**3** Indique qué afirmación es falsa:

- a) Los sistemas operativos multiusuario son capaces de dar servicio a más de un usuario a la vez, independientemente de la plataforma hardware sobre la que esté montado.
- b) Un sistema operativo multitarea es aquel que permite al usuario estar realizando varios trabajos al mismo tiempo.
- c) Un sistema operativo en red es aquel que mantiene a dos o más computadoras unidas a través de algún

medio de comunicación (físico o no), con el objetivo primordial de poder compartir los diferentes recursos y la información del sistema.

- d) Los sistemas monoproceso son los que únicamente permiten realizar un proceso a la vez sin permitir simular la multitarea.

**4** Indique qué afirmación es falsa:

- a) Los sistemas operativos de red usados más ampliamente son: *Windows Server* y *Linux Server*.
- b) Los sistemas operativos de escritorio usados más ampliamente son: *Windows XP*, *Windows Vista*, *Windows 7* y *Linux Server*.
- c) *Windows 7* es un sistema operativo multiusuario.
- d) *Linux* es un sistema operativo libre.

**5** Indique qué afirmación es falsa:

- a) Un software de dominio público es aquel que no requiere de licencia pues sus derechos de explotación pertenecen a todos por igual y cualquiera puede hacer uso de él, siempre con fines legales y consignando su autoría original.
- b) El software libre siempre es gratuito.
- c) NTLDR es el archivo encargado del arranque de los sistemas operativos: *Windows NT*, *Windows 2000*, *Windows XP* y *Windows Server 2003*.
- d) Grub es un gestor de arranque más moderno y flexible que Lilo, ya que permite que el administrador ejecute cualquier comando desde la línea de comando de Grub.