

1. Introducción a los sistemas informáticos

1.3 Seguridad de la información



• DVD (Digital Video Disk, Disco de vídeo digital)

Se aplican las mismas características que al CD-ROM, con la diferencia de que ofrecen mucha más capacidad de almacenamiento (entre 4,7 Gb y 9 Gb, dependiendo de si la unidad de DVD permite grabar en una sola capa o en doble capa). Actualmente han desbancado en popularidad a los CD-ROM.

• FMD-ROM (Fluorescent Multilayer Disc, Disco multi-capa fluorescente)

Los discos FMD-ROM contienen material fluorescente incrustado en surcos en sus más de 10 capas. En contacto con un láser, el

material fluorescente se estimula para generar luz coherente e incoherente; los datos se almacenan en la luz incoherente.

Los discos FMD-ROM pueden contener hasta 140 Gb, que son 215 CD-ROM de tamaño estándar. Las unidades FMD-ROM son compatibles con los discos CD-ROM y DVD, pero no a la inversa.

Muchos de los periféricos de entrada y salida que se han comentado necesitan programas de software para ser configurados y utilizados sin problemas; es decir, deben instalarse. Esto se hace mediante unos programas específicos que permiten al sistema operativo reconocer el periférico y utilizarlo de forma correcta. Estos programas, denominados *drivers* o *controladores*, se describen posteriormente.

1.3 Seguridad de la información

En determinados ordenadores, la integridad de la información que se procesa y almacena es muy importante, máxime si estos equipos son los denominados *servidores*.

La seguridad de la información, que es responsabilidad del administrador de red, puede centrarse en varios aspectos:

- Seguridad física.
- Seguridad frente a virus.
- Seguridad de datos.

A. Seguridad física

La seguridad física es un aspecto básico para los ordenadores que contienen información esencial para el funcionamiento de una organización o empresa. Muchas empresas utilizan redes de ordenadores, por lo que la seguridad debe ampliarse tanto a los ordenadores personales como a los de red.

El término «seguridad física» hace referencia a cómo es posible mantener seguro un equipo doméstico o de la organización ante posibles contingencias. En la última Unidad del libro se describen con todo detalle los aspectos de la seguridad informática, por lo que aquí sólo los veremos como una breve introducción.

Una **red** es un conjunto de ordenadores interconectados entre sí de tal forma que los recursos hardware y software de los que se dispone se pueden optimizar y centralizar mediante una administración adecuada.

Las redes centralizadas tienen un ordenador principal o **servidor de red**. Este ordenador tiene las siguientes características: alma-

cena la mayoría del software que utiliza la empresa, así como los datos de toda la organización, a él se conectan las distintas impresoras, permite acceso a Internet, se puede conectar con otros servidores, etcétera.

Más adelante se describen los diferentes tipos de redes, los recursos que gestionan, la forma de interconectar a los clientes con los servidores, la gestión de impresoras en red, etcétera.

Los servidores de red son ordenadores que suministran recursos hardware y software a gran cantidad de usuarios; por eso la integridad de la información que contienen, y con la que trabajan los usuarios, ha de estar protegida ante eventuales errores físicos: interrupción del suministro eléctrico, calor, polvo, campos electromagnéticos, virus, etcétera.

Para prevenir los errores físicos de tipo eléctrico, se utilizan los dispositivos denominados **SAI** (Sistema de Alimentación Inin-

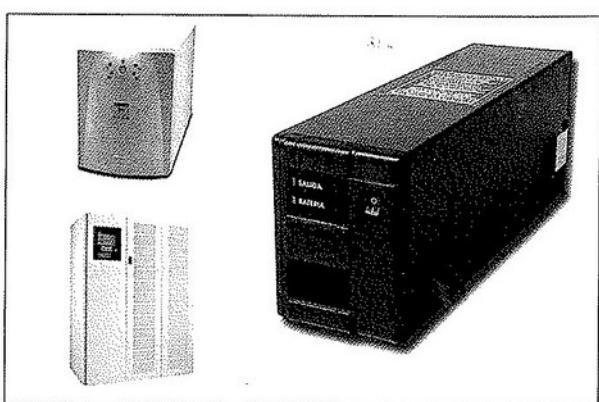
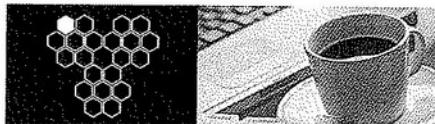


Fig. 1.15. Diferentes tipos de SAI.



1. Introducción a los sistemas informáticos

1.3 Seguridad de la información

terrumpida), que son acumuladores de corriente o baterías. Están conectados entre el ordenador y la red eléctrica, y permiten que, ante una interrupción del suministro eléctrico, el equipo siga funcionando durante unos minutos. De esta forma, el usuario puede finalizar correctamente los procesos que en ese momento se estén ejecutando sin que se pierdan datos, además de apagar el equipo de forma correcta.

En función de la potencia del SAI, puede conectarse individualmente a cada ordenador o bien conectar todos los ordenadores a la red eléctrica interna de la organización, cuya corriente pasa a través de un SAI. De esta forma, ante una interrupción del suministro eléctrico, el administrador de la red puede apagar correctamente el servidor de archivos, así como cada uno de los usuarios su ordenador personal.

B. Seguridad frente a virus

Se puede definir un **virus** como un programa de ordenador que puede infectar otros programas modificándolos para incluir una copia de sí mismo. El virus es un *huésped* que, normalmente, se ubica al principio o al final del conjunto de instrucciones que componen el programa en el que se aloja o al cual infecta.

Un virus está formado por un conjunto de instrucciones, dado que es un programa. Los virus informáticos tienen, básicamente, la función de propagarse por duplicación, pero algunos contienen además la carga viral con distintos objetivos, desde una simple broma hasta provocar daños importantes en los sistemas, o bloquear las redes informáticas al generar tráfico inútil.

Los programas a los que afecta son normalmente programas o archivos ejecutables (todos aquellos que tienen las extensiones .com y .exe). También pueden afectar a las **macros**, que son programas que se incluyen en otro tipo de archivos. Los hay que infectan el **boot** o sector de arranque, que es la parte del disco duro en la que se encuentran las instrucciones para inicializar el sistema operativo.

Por ello, se recomienda instalar en nuestro sistema los correspondientes antivirus para prevenir, en primer lugar, la entrada de éstos y, en segundo lugar, para poder eliminarlos en caso de que hayan entrado.

El mantenimiento e instalación, así como la configuración y la auditoría de los antivirus, es una de las funciones esenciales del administrador de un sistema informático, ya que de él depende la integridad del sistema.

Los medios más habituales de entrada de virus pueden ser los siguientes: unidades de disquete o CD-ROM, redes de ordenadores, Internet, etcétera.

El sistema informático debe estar protegido en diferentes puntos, ya que los virus, según su tipología y características, se suelen ubicar en: memoria RAM, documentos de texto con macros, sec-

tor de arranque de los discos duros u otros soportes de información, ficheros adjuntos de correo electrónico, etcétera.

Fundamentalmente existen dos tipos de virus: aquellos que infectan archivos (virus de acción directa y virus residentes) y los que infectan el sector de arranque.

Las técnicas más usuales para prevenir, o eliminar, virus son las siguientes: búsqueda de cadenas, excepciones, análisis heurístico, protección permanente, vacunación.

C. Seguridad de los datos

Los errores puede provocarlos el usuario de forma involuntaria. Puede ocurrir que copie en el disco duro del ordenador un archivo con el mismo nombre de otro que ya existía anteriormente. En este caso, lo más probable es que la información del archivo antiguo desaparezca y provoque problemas en el sistema.

También puede ocurrir que cambie la configuración inicial del equipo y, a continuación, se dé cuenta que no es correcta. En este caso, no habrá forma de saber cuál era la configuración original.

Así mismo, puede darse el caso de que el usuario dé formato a su disco duro de forma accidental, con lo que perderá toda la información.

Estos y otros problemas similares pueden solucionarse mediante la utilización de los *discos reflejados o espejo* o las *copias de seguridad*. Como su nombre indica, un **disco espejo** es un soporte de información que contiene exactamente la misma información y configuración que el disco duro en los que está instalado el sistema, el software de aplicación y los datos que se procesan. Este disco espejo puede encontrarse físicamente dentro del mismo equipo o fuera de él. Así, en casos accidentales de pérdida de información original, ésta se puede recuperar a partir del disco reflejado.

Es posible crear un disco espejo en el mismo disco duro físico que se desea copiar. En este caso, una parte del disco duro está destinada a realizar de forma continua una copia de lo que el usuario realiza y modifica en todo momento en el ordenador. Así, si desaparece lo que se está haciendo, se tendrá un duplicado.

Esta forma de gestionar los discos espejo no es la más conveniente, ya que solamente serán útiles ante errores lógicos, pero no ante errores físicos, como cuando se raya un disco duro, por ejemplo.

La mayoría de los sistemas operativos multiusuario, como Windows NT Server, Windows 2000 Server, Unix, Novell NetWare, etc., incorporan la posibilidad de mantener una segunda máquina u ordenador como dispositivo adicional de seguridad. En Windows NT Server, 2000 Server y Server 2003 son los denominados **BDC** (*Backup Domain Controller*, Controlador de dominio de reserva).

1. Introducción a los sistemas informáticos

1.4 Componentes lógicos. Tipos de datos y tipos de software



El sistema de tolerancia a errores III de Novell NetWare realiza copias periódicas de información en diferentes discos duros para mantener la integridad de la misma. Cualquiera de estos discos duros permite almacenar una copia de todo el sistema para reparar eventuales errores ante un fallo del ordenador principal.

Otra medida preventiva recomendable consiste en realizar copias de seguridad periódicas de los datos y de la información que se está procesando en el sistema informático. Las copias de seguridad son fundamentales, sobre todo para los administradores de sistemas informáticos, ya que permiten recuperar en el momento deseado cualquier información perdida. Estas copias de seguridad deben realizarse con la frecuencia oportuna. Así, en sistemas informáticos en los que se realizan muy pocas operaciones, bastará con que se hagan una vez cada dos semanas. Por otro lado, en sistemas que actualizan permanentemente la informa-

ción sería conveniente realizar una copia de seguridad con mayor frecuencia.

Para hacer estas copias existen soportes y dispositivos especiales. Estos periféricos suelen ser de tipo secuencial en la mayoría de los casos. Ejemplos claros pueden ser las cintas **DAT** o **DLT**, o los **streamer**. Son soportes bastante lentos, pero de gran capacidad de almacenamiento y de elevada seguridad. También suelen utilizarse CD-ROM o DVD, tanto grabables como regrabables; son más rápidos que los dispositivos anteriores, pero de menor capacidad.

La realización de copias de seguridad es tarea del administrador del sistema. De él depende que la información esencial esté siempre actualizada, protegida y en perfecto estado de uso para solucionar posibles pérdidas provocadas por cualquier error físico o humano.

1.4 Componentes lógicos. Tipos de datos y tipos de software

Hasta el momento, se han descrito básicamente los componentes físicos del ordenador. Es obvio que, para que el ordenador funcione, necesita información con la que trabajar. Esta información puede ser de varios tipos, dependiendo de su función.

En un primer lugar, el ordenador sirve para procesar información en forma de datos.

En segundo lugar, también maneja información que servirá para procesar esos datos. En este caso, se hace referencia a programas o aplicaciones informáticas como, por ejemplo, procesadores de textos, que se utilizan para procesar datos en formato texto, herramientas de diseño gráfico que se utilizan para procesar datos en formato gráfico, etcétera.

Por último, un sistema informático necesita otro tipo de software fundamental: programas y datos que hacen funcionar las aplicaciones informáticas que, a su vez, procesan sus propios datos. En este caso, se hace referencia al **sistema operativo**, o componente software. Sirve para que la información pueda ser procesada por las aplicaciones informáticas, mediante la utilización de todos los componentes hardware del sistema informático. El sistema operativo consta de programas propios que sirven para realizar otras funciones.

En este punto, es necesario diferenciar claramente los tres tipos de software que puede incluir un sistema informático. En primer lugar, aplicaciones informáticas que no forman parte del sistema operativo y que se instalan después de instalar el propio sistema operativo. En segundo lugar, los datos que procesan esas aplicaciones informáticas y que normalmente son introducidos por el usuario. En tercer lugar, y como software fundamental, el sistema operativo, que consta de sus propios programas de control, de pro-

ceso, de gestión de datos, de gestión del sistema, de gestión de periféricos, etc. Actualmente, el sistema operativo incorpora, además de las instrucciones básicas para hacer funcionar el hardware y las aplicaciones informáticas que no forman parte de él, procesadores de textos, accesorios, juegos, programas de diseño gráfico, etcétera.

Es evidente que si el usuario adquiere un ordenador únicamente con el sistema operativo, estará muy limitado en cuanto a las tareas que pueda realizar hasta que no instale alguna aplicación informática o programa específico según sus necesidades. Por ello, los fabricantes de sistemas operativos incluyen, cada vez más, en el software base del sistema operativo, aplicaciones específicas que puedan cubrir la mayoría de las necesidades básicas de cualquier usuario.

A. Datos. Tipos de datos

Una primera clasificación de los datos puede ser la siguiente:

- **Datos de entrada.** Son los que se suministran al ordenador desde los periféricos de entrada (teclado, ratón, módem, escáner, etc.) o desde los diferentes soportes de información (discos duros, CD-ROM, etc.). Forman la primera fase del tratamiento automático de la información: **entrada**.
- **Datos intermedios.** Son los que se obtienen en la segunda fase del tratamiento automático o de la información: **procesamiento**.
- **Datos de salida.** También llamados resultados, completan el proceso del tratamiento automático de la información: **salida**. Pueden mostrarse mediante los distintos periféricos de salida



1. Introducción a los sistemas informáticos

1.4 Componentes lógicos. Tipos de datos y tipos de software

(monitor, impresora, trazador, etc.), con su posterior distribución y análisis, completan el proceso.

Otra posible clasificación de los datos, según varíen o no durante su procesamiento, es la siguiente:

- **Datos fijos.** Son aquellos que permanecerán constantes durante el procesamiento o programa que se les aplique. Los datos fijos reciben el nombre de **constantes**. Un ejemplo podría ser un programa que emita facturas en euros y pesetas; es evidente que el cambio del euro será el mismo en todo el proceso.
- **Datos variables.** Son aquellos que sí se modifican a lo largo del procesamiento, según se produzcan determinadas condiciones o acciones realizadas por los programas.

En función de cómo utiliza el ordenador los datos, éstos se clasifican en:

- **Datos numéricos.** Son los dígitos del 0 al 9.
- **Datos alfabéticos.** Son las letras mayúsculas y minúsculas de la a la Z.
- **Datos alfanuméricos.** Son una combinación de los anteriores, más una serie de caracteres especiales.

B. Sistemas de codificación

Los sistemas de codificación se utilizan para procesar la información que el usuario comprende y el ordenador no. Es evidente que el usuario y el sistema informático trabajan en lenguajes diferentes.

Como hemos visto anteriormente, la memoria, y por extensión todos los componentes del ordenador, funcionan mediante pequeños biestables o componentes electrónicos que pueden adoptar dos estados diferentes: con corriente o sin ella.

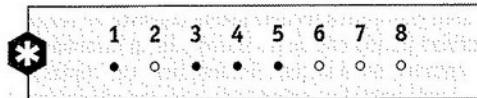
La memoria del ordenador no puede almacenar la letra «A» o el carácter «*». Ésta y, por extensión, el resto de componentes internos, no comprende las letras o números, sino sólo los impulsos de corriente eléctrica.

Por tanto, cuando queremos almacenar una letra en memoria, como la primera letra de su documento de texto, el ordenador, mediante el sistema operativo y componentes hardware, transforma la letra que se desea almacenar en un conjunto de impulsos eléctricos (normalmente 8). Recordemos el ejemplo en el que almacenábamos el carácter «/» (barra de dividir) en memoria.

Las celdas de memoria pueden adoptar los dos estados siguientes:

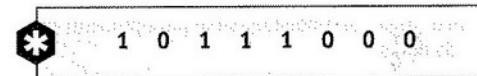
- Indica presencia de corriente eléctrica.
- Indica ausencia de corriente eléctrica.

El sistema operativo y el resto de componentes hardware tienen que transformar ese carácter en alguna combinación válida de impulsos eléctricos para almacenarlo. En este caso las ocho celdas de memoria correspondientes se magnetizarán o no de la siguiente forma:



Pero ¿en qué se basa el sistema informático para saber qué celda tiene que magnetizar y cuál no? La respuesta es que, el ordenador, se basa en un **código**. Cuando se escribe cualquier carácter, se busca dentro de una tabla la correspondencia adecuada (código ASCII, que se describe posteriormente). Esta tabla de códigos la introduce el fabricante del sistema operativo dentro del conjunto de instrucciones y datos que lo forman. Es un estándar internacional y todos los fabricantes de software y hardware la conocen y la utilizan. De esta forma todos ellos tienen las mismas equivalencias y a cada uno de ellos les resulta fácil interpretar la información que procesan otros programas o componentes hardware.

En el ejemplo anterior, el conjunto de ocho celdas de memoria que se han magnetizado al escribir el carácter «/» lo han hecho debido a que el sistema operativo ha leído en su tabla de códigos el siguiente valor.



Lo ha interpretado y ha magnetizado (1) o no (0) la celda de memoria, dependiendo de la combinación de esos ocho valores, dígitos binarios o bits.

Si, por el contrario, se lee una posición de memoria, primero se analizan las celdas correspondientes. Una vez analizadas ocho de ellas, se sabe, por diseño del propio sistema operativo y gracias a la equivalencia del código, que se ha leído un byte o carácter como conjunto de ocho bits. Cada posición magnetizada se convierte en un uno, y cada posición no magnetizada, en un cero. Se busca en la tabla de códigos y se compara la combinación de esos ocho bits, con lo que se obtiene la equivalencia con el carácter determinado. A continuación, se muestra el carácter equivalente al byte leído, no los ocho bits.

En este apartado se describen los códigos numéricos y alfanuméricos. Los primeros se utilizan para representar sólo números, mientras que los segundos se usan para representar números, letras y símbolos especiales.

Antes de explicar los sistemas de codificación, es necesario introducir algunos conceptos. Se define un **sistema de numeración** como el conjunto de símbolos y reglas que se utilizan para representar cantidades o datos numéricos.

1. Introducción a los sistemas informáticos

1.4 Componentes lógicos. Tipos de datos y tipos de software



Estos sistemas se caracterizan por la **base** a la que referencian y que determina el diferente número de símbolos que lo componen. El ser humano utiliza el sistema de numeración en base 10, compuesto por diez números diferentes (del 0 al 9).

Los sistemas de numeración que se utilizan son sistemas posicionales; es decir, el valor relativo que cada símbolo representa quedará determinado por su valor absoluto y la posición que ocupe dicho símbolo en un conjunto.

$$* \quad 283 = 2 \cdot 10^2 + 8 \cdot 10^1 + 3 \cdot 10^0$$

Todos los sistemas posicionales están basados en el **Teorema fundamental de la numeración** (TFN), que sirve para relacionar una cantidad expresada en cualquier sistema de numeración con la misma cantidad expresada en el sistema decimal. Viene dado por la fórmula siguiente:

$$* \quad \sum X_i \cdot B^i$$

donde X es el valor absoluto del dígito en cuestión, i es la posición que ocupa el dígito con respecto al punto decimal y B es la base.

Esta misma fórmula también se puede expresar de la siguiente forma:

$$* \quad \dots X_2 \cdot 10^2 + X_1 \cdot 10^1 + X_0 \cdot 10^0 + \\ + X_{-1} \cdot 10^{-1} + X_{-2} \cdot 10^{-2} \dots$$

Sistemas de codificación numérica

- **Binario.** Este sistema utiliza dos símbolos diferentes: el cero y el uno (0, 1). Es el sistema que maneja el ordenador internamente, ya que lo utilizan sus componentes electrónicos.

Cada uno de estos símbolos recibe el nombre de **bit**, o mínima unidad de información posible.

Los símbolos del sistema decimal pueden representarse (codificarse) en binario mediante el TFN, con una combinación de cuatro bits.

- **Octal.** Sistema en base 8 que utiliza los números del 0 al 7 para representar las cantidades, las cuales quedan representadas posicionalmente por potencias de 8. El sistema de numeración en base 8 tiene una correspondencia directa con el binario, ya que cada símbolo en base 8 puede representarse mediante una combinación de 3 bits.

- **Hexadecimal.** Sistema de numeración en base 16. Utiliza 16 símbolos diferentes: los números del 0 al 9 y las letras A, B, C, D, E y F. Estas letras representan, respectivamente, los dígitos 10, 11, 12, 13, 14 y 15 del sistema decimal. Este sistema también tiene una correspondencia directa con el sistema binario, ya que cada símbolo en base 16 se puede representar mediante una combinación de 4 bits. El sistema que maneja internamente un ordenador es el binario, pero, en ocasiones, por comodidad a la hora de manejar los datos, se suele utilizar el octal y el hexadecimal, ya que buena parte de la información que muestra el sistema operativo, como direcciones de memoria, está expresada en hexadecimal.

El sistema de numeración binario tiene una gran importancia en el funcionamiento del ordenador. Ya se ha señalado que la memoria del ordenador es un conjunto de biestables que pueden contener o no corriente eléctrica.

La Figura 1.16 muestra los primeros 20 símbolos decimales y sus correspondencias en binario, base 8 y base 16.

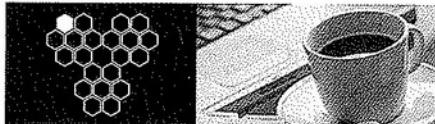
En el Caso práctico 3, se explica cómo pasar números de una base de numeración a otra.

Si queremos pasar de un número entero del sistema decimal al sistema binario, será necesario que dividamos sucesivamente entre dos hasta que resulte un cociente 0. El número en binario se obtiene uniendo todos los restos en orden inverso al último obtenido.

Si pasáramos el mismo número a base 16, el primer resto sería 10. Pero este símbolo en hexadecimal no existe; se utiliza la A como símbolo décimo de la base.

Decimal	Binario	Base 8	Base 16
0	00000	0	0
1	00001	1	1
2	00010	2	2
3	00011	3	3
4	00100	4	4
5	00101	5	5
6	00110	6	6
7	00111	7	7
8	01000	10	8
9	01001	11	9
10	01010	12	A
11	01011	13	B
12	01100	14	C
13	01101	15	D
14	01110	16	E
15	01111	17	F
16	10000	20	10
17	10001	21	11
18	10010	22	12
19	10011	23	13

Fig. 1.16. Sistema decimal, binario, octal y hexadecimal.



1. Introducción a los sistemas informáticos

1.4 Componentes lógicos. Tipos de datos y tipos de software

Cada número en base 8 y en base 16 tiene una correspondencia directa con el número en binario mediante un conjunto de 3 y 4 bits, respectivamente. Es decir, si se desea transformar un número en base 8 o en base 16 a un número en base 2, bastará con formar grupos de 3 o 4 bits, respectivamente. Esta transformación se llama **directa**.

Por ejemplo, consideremos el número 132 en base 8.

* $132_{(8)} = 001\ 011\ 010_{(2)}$

Como se puede apreciar, hemos transformado directamente el dígito 1 en 001, el 3 en 011 y el 2 en 010. Cada dígito en base 8 tiene su correspondencia con tres dígitos binarios (véase la Figura 1.16). De esta forma es posible transformar un número en base n a otro en base m . Si las bases n y m son potencias de dos, la transformación es directa, como se ha indicado.

Supongamos que queremos pasar el número 132 de base 8 a base 16. A continuación, se describe cómo pasar directamente el número 132 de octal a hexadecimal sin necesidad de recurrir a pasar el número primero a base 10 o decimal.

En primer lugar, descomponemos cada dígito octal en sus correspondientes agrupaciones de dígitos binarios.

* $132_{(8)} = 001\ 011\ 010_{(2)}$

Después tomamos el valor binario y, de derecha a izquierda, hacemos bloques de 4 bits. Si el último bloque no tiene dígitos suficientes, se rellenan con ceros, ya que, como en cualquier sistema de numeración, los ceros a la izquierda no son significativos. *En este ejemplo aparecen en cursiva*.

* $0000\ 0101\ 1010_{(2)} = 0000\ 0101\ 1010_{(16)}$

El conjunto más a la derecha tiene su equivalencia con el dígito 10 en hexadecimal, pero como este símbolo no existe en este sistema de numeración, se corresponde con su símbolo correspondiente, que es la letra A. El bloque del centro se corresponde con el dígito 5 y, evidentemente, los cuatro ceros de la izquierda representan un cero, y como tal no tiene valor precisamente por estar a la izquierda; en este caso el resultado sería el siguiente:

* $0000\ 0101\ 1010_{(2)} = 05A_{(16)} = 5A_{(16)}$

El mismo caso sería el paso de base 16 a base 8. Para ello bastaría con formar bloques de tres bits.

También podemos realizar este cambio de base mediante el **método indirecto**, que consiste en pasar el número de base n a base 10, y posteriormente a base m . Este método hay que utilizarlo siempre que las bases de numeración no tengan correspondencia posicional.

Así, si se desea transformar un número de base 6 al correspondiente en base 5, es evidente que siempre será necesario pasar por base 10. Ahora bien, si las bases son binario, octal y hexadecimal, al ser potencias de dos y ser equivalentes, el paso puede ser directo.

Partimos del mismo ejemplo. Se pasa el número 132 octal a base 16, pero pasando primero por base 10. La forma de convertir un número de base n a base 10 consiste en utilizar el Teorema fundamental de la numeración.

Casos prácticos

3 Pasar a base 2 el número 90 en base 10:

$$\begin{aligned} 90 : 2 &= 45. \text{ Resto } 0. \\ 45 : 2 &= 22. \text{ Resto } 1. \\ 22 : 2 &= 11. \text{ Resto } 0. \\ 11 : 2 &= 5. \text{ Resto } 1. \\ 5 : 2 &= 2. \text{ Resto } 1. \\ 2 : 2 &= 1. \text{ Resto } 0. \\ 1 : 2 &= 0. \text{ Resto } 1. \end{aligned}$$

Resultado: $90_{(10)} = 1011010_{(2)}$

Si se desea pasar el mismo número a base 8 y 16, la forma de proceder será la misma, si bien se tendrá en cuenta que ahora el divisor es el 8 o el 16, respectivamente.

Paso a base 8:

$$\begin{aligned} 90 : 8 &= 11. \text{ Resto } 2. \\ 11 : 8 &= 1. \text{ Resto } 3. \\ 1 : 8 &= 0. \text{ Resto } 1. \end{aligned}$$

Resultado: $90_{(10)} = 132_{(8)}$

Paso a base 16:

$$\begin{aligned} 90 : 16 &= 5. \text{ Resto } 10. \\ 5 : 16 &= 0. \text{ Resto } 5. \end{aligned}$$

Resultado: $90_{(10)} = 5A_{(16)}$

1. Introducción a los sistemas informáticos

1.4 Componentes lógicos. Tipos de datos y tipos de software



La operación sería la siguiente:

$$\begin{aligned} * & 1 \cdot 8^2 + 3 \cdot 8^1 + 2 \cdot 8^0 = \\ & = 1 \cdot 64 + 3 \cdot 8 + 2 \cdot 1 = 64 + 24 + 2 = 90 \end{aligned}$$

A partir de aquí, procederemos como se indicó anteriormente para transformar el número 90 en base 10 al correspondiente en base 16.

Para finalizar, se ofrece otro pequeño ejemplo. En este caso, pasaremos el número binario 1 001 a base 10:

$$\begin{array}{r} * \quad 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = \\ \qquad\qquad\qquad = 8 + 0 + 0 + 1 = 9 \end{array}$$

- **Coma o punto fijo.** El punto fijo se usa para la representación de números enteros. Hay tres formas de representar los números en punto fijo: binario puro, decimal desempaquetado y decimal empaquetado.

Para el binario puro se utiliza una combinación de 32 bits en la que el bit de la izquierda sirve para representar el signo: 0 para el signo + y 1 para el signo -. Los restantes 31 bits sirven para representar el valor del número.

Por ejemplo, el número -10 se escribiría como:

El decimal desempaquetado representa cada número decimal, de forma que cada una de sus cifras ocupa un byte u octeto.

En primer lugar, para hablar de la codificación en decimal desempaquetado, es necesario conocer cómo se representan los números decimales en **BCD** (*Binary Coded Decimal*, Decimal codificado en binario).

En este sistema, cada dígito decimal se representa con una combinación de 4 bits. Las cifras decimales del 0 al 9 quedarían representadas en BCD (Figura 1.17).

Cada número en decimal desempaquetado incluye en los cuatro bits de la izquierda los denominados *bits de zona*. El cuarteto de la derecha se utiliza para codificar el número en BCD. El signo se representa en el cuarteto de bits de la izquierda correspondiente al último número: 1100 para el signo positivo y 1101 para el signo negativo.

Si se desea representar el número 2371 decimal en decimal desempaquetado, el resultado sería el siguiente:

Decimal	DCB
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Fig. 1.17. Código BCD.

*** 1111 0010 1111 0011 1111 0111 1100 0001
signo +**

Y si se trata del -2371:

* 1111 0010 1111 0011 1111 0111 1101 0001
signo -

El decimal empaquetado representa cada cifra con un conjunto de 4 bits. El conjunto de 4 bits de la derecha se usa para representar el signo con la misma combinación que en el caso anterior.

El número 2 371 en decimal empaquetado se representaría de la siguiente forma:

* 0010 0011 0111 0001 1100
signo +

- **Coma flotante.** Se utiliza para representar números reales y enteros con un rango de representación mayor que el que ofrece el punto fijo. De esta forma, se consigue que el ordenador pueda tratar números muy grandes o muy pequeños.

Para representar así los números, se utiliza la **notificación científica**, que tiene el siguiente formato:

N1 = mantisa · base de exponente^{exponente}

- El exponente de la anterior fórmula también se denomina **características**.

- La **mantisa** es un número real con el punto decimal implícito a la izquierda de los bits que lo representan.



1. Introducción a los sistemas informáticos

1.4 Componentes lógicos. Tipos de datos y tipos de software

- La **base de exponenciación** es una potencia de 2 que dependerá del fabricante del componente.

La representación de números en coma flotante se puede hacer de dos formas:

- *Simple precisión*. Se utilizan 32 bits para representar una cifra.
- *Doble precisión*. Se utilizan 64 bits para representar una cifra.

Codificación alfanumérica

Como se ha indicado, los datos pueden ser numéricos, alfabéticos o alfanuméricos. Normalmente, con los datos alfanuméricos podemos construir instrucciones y programas.

Por otro lado, es lógico pensar que el ordenador no solamente procesará datos numéricos, sino también datos alfabéticos y combinaciones de los anteriores, como datos alfanuméricos.

Los sistemas de codificación alfanumérica sirven para representar una cantidad determinada de símbolos en binario. A cada símbolo le corresponderá una combinación de un número de bits.

Los sistemas de codificación alfanumérica más importantes son:

- **ASCII (American Standard Code for Information Interchange, Código estándar estadounidense para el intercambio de información)**. Este sistema utiliza una combinación de 7 u 8 bits (dependiendo del fabricante) para representar cada símbolo. Es el más utilizado y admite hasta 2^8 símbolos diferentes.

Con este código es posible representar los dígitos del 0 al 9, las letras mayúsculas y minúsculas de la A a la Z, caracteres especiales y algunos otros caracteres denominados de control.

La Figura 1.18 muestra los primeros 128 símbolos diferentes que se pueden representar con el código ASCII de 8 bits.

Caracteres no imprimibles				Caracteres imprimibles								
Nombre	Dec.	Hex.	Car.	Dec.	Hex.	Car.	Dec.	Hex.	Car.	Dec.	Hex.	Car.
Nulo	0	00	NUL	32	20	Espacio	64	40	@	96	60	'
Inicio de cabecera	1	01	SOH	33	21	!	65	41	A	97	61	a
Inicio de texto	2	02	STX	34	22	"	66	42	B	98	62	b
Fin de texto	3	03	ETX	35	23	#	67	43	C	99	63	c
Fin de transmisión	4	04	EOT	36	24	\$	68	44	D	100	64	d
Consulta	5	05	ENQ	37	25	%	69	45	E	101	65	e
Confirmación	6	06	ACK	38	26	&	70	46	F	102	66	f
Sonido (beep)	7	07	BEL	39	27	'	71	47	G	103	67	g
Retroceso	8	08	BS	40	28	(72	48	H	104	68	h
Tabulación horizontal	9	09	HT	41	29)	73	49	I	105	69	i
Salto de línea	10	0A	LF	42	2A	*	74	4A	J	106	6A	j
Tabulación vertical	11	0B	VT	43	2B	+	75	4B	K	107	6B	k
Salto de página	12	0C	FF	44	2C	,	76	4C	L	108	6C	l
Retorno de carro	13	0D	CR	45	2D	-	77	4D	M	109	6D	m
Mayús fuera	14	0E	SO	46	2E	.	78	4E	N	110	6E	n
Mayús dentro	15	0F	SI	47	2F	/	79	4F	O	111	6F	o
Escape línea de datos	16	10	DLE	48	30	0	80	50	P	112	70	p
Control dispositivo 1	17	11	DC1	49	31	1	81	51	Q	113	71	q
Control dispositivo 2	18	12	DC2	50	32	2	82	52	R	114	72	r
Control dispositivo 3	19	13	DC3	51	33	3	83	53	S	115	73	s
Control dispositivo 4	20	14	DC4	52	34	4	84	54	T	116	74	t
Configuración negativa	21	15	NAK	53	35	5	85	55	U	117	75	u
Sincronismo	22	16	SYN	54	36	6	86	56	V	118	76	v
Fin de bloque transmitido	23	17	ETB	55	37	7	87	57	W	119	77	w
Cancelar	24	18	CAN	56	38	8	88	58	X	120	78	x
Fun medio	25	19	EM	57	39	9	89	59	Y	121	79	y
Sustituto	26	1A	SUB	58	3A	:	90	5A	Z	122	7A	z
Escape	27	1B	ESC	59	3B	;	91	5B	[123	7B	{
Separador de archivos	28	1C	FS	60	3C	<	92	5C	\	124	7C	
Separador de grupos	29	1D	GS	61	3D	=	93	5D]	125	7D	}
Separador de registros	30	1E	RS	62	3E	>	94	5E	^	126	7E	~
Separador de unidades	31	1F		63	3F		95	5F		127	7F	

Fig. 1.18. Código ASCII de 8 bits (hasta el carácter 127).

1. Introducción a los sistemas informáticos

1.4 Componentes lógicos. Tipos de datos y tipos de software



- **EBCDIC (Extended BCD Interchange Code)**, Código BCD extendido para el intercambio de información). Cada símbolo se representa mediante una combinación de 8 bits agrupados en dos bloques de cuatro. Es el formato extendido del BCD. La Figura 1.19 muestra la representación de cada uno de sus símbolos.
- **FIELDATA**. Utiliza bloques de 6 bits para representar los diferentes símbolos. Su implantación es limitada, dado que se usa en ordenadores que procesan la información en bloques de 36 bits.
- **UNICODE**. Código estándar internacional que se utiliza en la mayoría de los sistemas operativos. Permite que un producto software o página web específica esté disponible para varias plataformas, idiomas o países, sin necesidad de modificar su diseño. El código ASCII, por ejemplo, tiene una tabla específica para cada país, ya que los diferentes símbolos de todos los países no cabrían en una tabla única.

Es una norma, desarrollada por **Unicode Consortium**, que regula la codificación de caracteres y ofrece un sistema de

codificación de caracteres internacional extensible a 16 bits (ASCII trabaja con 8 bits) para procesar la información que abarca la mayor parte de los idiomas del mundo.

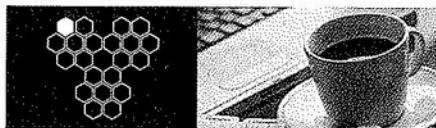
Define la codificación de caracteres, así como las propiedades y los algoritmos que se utilizan en su aplicación. Proporciona un número único para cada carácter, independientemente de la plataforma, (hardware) el programa (software) o el idioma.

La mayoría de líderes del mercado informático, como Apple, HP, IBM, Microsoft, Oracle, Sun, Unisys y otros, han adoptado el estándar Unicode, lo que permite crear aplicaciones y hardware estándar con XML, Java, etcétera.

Es compatible con muchos sistemas operativos actuales, así como con la mayoría de los exploradores de Internet, y permite que un producto software se oriente a varias plataformas o idiomas sin necesidad de volverlo a diseñar. Pobres de aquellos que vivimos en **España** y nos apellidamos **Muñoz**, ya que, hasta que se estandarice Unicode, el nombre de nuestra nación o nuestro apellido nunca será como tiene que ser.

Carácter	EBCDIC	Hexadecimal	Carácter	EBCDIC	Hexadecimal
A	11000001	C1	2	11110010	F2
B	11000010	C2	3	11110011	F3
C	11000011	C3	4	11110100	F4
D	11000100	C4	5	11110101	F5
E	11000101	C5	6	11110110	F6
F	11000110	C6	7	11110111	F7
G	11000111	C7	8	11111000	F8
H	11001000	C8	9	11111001	F9
I	11001001	C9	ESPACIO	01000000	40
J	11010001	D1	.	01001011	4B
K	11010010	D2	<	01001100	4C
L	11010011	D3	(01001101	4D
M	11010100	D4	+	01001110	4E
N	11010101	D5	&	01010000	50
O	11010110	D6		01011010	5A
P	11010111	D7	*	01011100	5C
Q	11011000	D8)	01011101	5D
R	11011001	D9	:	01011110	5E
S	11000010	E2	-	01100000	60
T	11100011	E3	/	01100001	61
U	11100100	E4	,	01101011	68
V	11100101	E5	%	01101100	6C
W	11100110	E6	>	01101110	6E
X	11101000	E7	{	01101111	6F
Y	11101001	E8	:	01110100	7A
Z	11101011	E9	'	01111101	7D
0	11110000	F0	=	01111110	7E
1	11110001	F1	"	01111111	7F

Fig. 1.19. Código EBCDIC.



1. Introducción a los sistemas informáticos

1.4 Componentes lógicos. Tipos de datos y tipos de software

C. Medida de la información

Como hemos indicado, el bit es la mínima unidad de información. Éste queda representado por un 0 o un 1.

En este sentido, se puede establecer una equivalencia de medidas en múltiplos de bits utilizados para designar cada una de éstas:

- **Nibble o cuarteto.** Conjunto de 4 bits.
- **Byte u octeto.** Conjunto de 8 bits.
- **Kilobyte (Kb).** Conjunto de 1 024 bytes.
- **Megabyte (Mb).** Conjunto de 1 024 Kb.
- **Gigabyte (Gb).** Conjunto de 1 024 Mb.
- **Terabyte (Tb).** Conjunto de 1 024 Gb.
- **Petabyte (Pb).** Conjunto de 1 024 Tb.
- **Exabyte (Eb).** Conjunto de 1 024 Pb.
- **Zettabyte (Zb).** Conjunto de 1 024 Eb.
- **Yottabyte (Yb).** Conjunto de 1 024 Zb.

El número 1 024 es una potencia de 2 (2^{10}). Su uso está justificado, ya que el ordenador utiliza internamente el sistema de codificación binario para todas sus operaciones.

El byte se suele utilizar para representar un carácter alfanumérico. Anteriormente se comentó cómo las codificaciones ASCII y EBCDIC utilizan 8 bits para representar cada símbolo.

Actualmente, la capacidad de la memoria RAM se mide en Mb o Gb, y la capacidad de los discos duros, en Gb o Tb.

D. Componentes software. Sistema operativo y aplicaciones

El software se compone de dos partes fundamentales:

- **Software básico.**
- **Software de aplicaciones.**

Se define como software básico aquella parte del software sin la cual el ordenador no puede funcionar. También se le denomina **sistema operativo** y se describe con detalle en la siguiente unidad. El sistema operativo es el alma del ordenador. Sirve de comunicación entre el usuario y el hardware de la máquina. Controla los

recursos hardware de la máquina según las necesidades, los programas de aplicación, el lugar donde se almacenan los datos, el momento en que hay que imprimir, cuándo se pulsa un botón del ratón, etcétera.

La Unidad 2 está dedicada al sistema operativo y sus tipos, que depende de varios parámetros: número de tareas, procesadores, tiempo de respuesta, etcétera.

El **software de aplicaciones** es la parte del software que sirve para procesar la información. Lo integran los programas y los datos. Los programas permiten editar textos, extraer información, editar gráficos, realizar cálculos numéricos, etcétera.

Los programas están formados por un conjunto de órdenes o instrucciones, y se utilizan para procesar los datos que se introducen como información.

Los datos componen la información que los programas pueden procesar utilizando las diferentes partes del hardware que intervienen en un sistema informático.

Otra clasificación del software de aplicación atiende a que éste sea **estándar o personalizado**. El software estándar es el que el usuario puede adquirir en el mercado con unas características predeterminadas. Para el empleo de este software, el usuario se adapta a su forma de trabajo y a las características del propio software.

Por el contrario, el software personalizado lo diseñan e implementan programadores (lo codifican en un lenguaje de programación) en función de las necesidades particulares de cada usuario; es decir, el software se personaliza para el usuario.

E. Lenguajes de programación

El software, tanto el básico como el de aplicaciones, experimenta actualmente un auge enorme, que ha sido provocado, en primer lugar, por las crecientes necesidades de los usuarios de procesar más información y de forma más rápida. Por otro lado, el desarrollo del hardware permite que los programas sean cada vez más eficaces y complejos, y que procesen más información en menos tiempo.

Como consecuencia, los *lenguajes de programación* han sufrido cambios espectaculares para adaptarse a tales necesidades. Un **lenguaje de programación** es una notación que se utiliza para escribir programas mediante un conjunto de instrucciones. Los programas permiten utilizar el hardware del ordenador con las órdenes adecuadas.

Como todo lenguaje, el de programación está definido por una gramática o conjunto de reglas que se aplican a su alfabeto.

Los lenguajes de programación, según se aproximen más o menos al lenguaje máquina (que es el lenguaje que comprende

1. Introducción a los sistemas informáticos

1.4 Componentes lógicos. Tipos de datos y tipos de software



el ordenador; es decir, 0 y 1), se pueden clasificar de la siguiente forma:

- **Lenguaje máquina** o de bajo nivel.
- **Lenguaje intermedio** o lenguaje ensamblador.
- **Lenguaje de alto nivel.**

Normalmente, los programas se crean en lenguaje de alto nivel, e incorporan instrucciones u órdenes que permiten realizar cálculos muy complejos. Con un lenguaje de bajo nivel, estos mismos cálculos requerirán muchas instrucciones. El programador incluye las instrucciones, pero se despreocupa de cómo éstas se ejecutan internamente.

Los lenguajes de alto nivel son muy variados, y cada uno de ellos se utiliza según la aplicación informática que se desee desarrollar. Actualmente, la mayoría de los lenguajes de alto nivel se desarrollan en entornos gráficos y no en entornos de texto, como los lenguajes de bajo nivel.

Los **lenguajes imperativos** son los que utilizan instrucciones para realizar el proceso. Pueden estar orientados a la gestión (Cobol), a procesos matemáticos (Pascal) o a procesos complejos (C, Ada, etcétera).

Actualmente, los lenguajes de programación más extendidos son los que se usan en la **programación orientada a objetos**, como, por ejemplo C++, Visual Basic, Smalltalk, etc. Cada uno de ellos está diseñado con una finalidad específica, aunque se puede hacer casi de todo con cualquiera. C++ es un lenguaje orientado a objetos. Visual Basic está orientado a eventos. Los gestores de bases de datos trabajan también orientando sus programas a objetos, como Oracle.

Para crear programas utilizando los lenguajes de programación se siguen varios pasos:

1. **Confección de la estructura lógica**, es decir, lo que queremos hacer. A esta estructura se le llama pseudocódigo, árboles, diagramas programáticos, etcétera.
2. **Implementación o codificación** de la estructura en lenguaje fuente, que es el propio lenguaje de programación.
3. **Depuración de errores sintácticos y lógicos**, mediante el uso de compiladores y depuradores.
4. **Transformación a lenguaje máquina**, que consiste en que el ordenador comprenda, mediante el uso de los montadores o editores de vínculos.

Dependiendo del tratamiento y análisis de las instrucciones, los lenguajes de programación se clasifican en dos grandes bloques: lenguajes intérpretes y lenguajes compiladores.

- **Lenguajes intérpretes.** Las instrucciones se introducen a través de un editor propio del lenguaje de programación. Una vez finalizada la introducción de instrucciones, el programa puede ejecutarse directamente.

En este caso, por cada instrucción que se ejecuta, se realiza previamente un **análisis sintáctico** de la misma. Si la instrucción tiene algún error sintáctico, el programa se interrumpe y permite al programador corregirlo.

El inconveniente que tienen los lenguajes de este tipo es que si en la fase de prueba existe alguna instrucción mal construida sintácticamente pero el programador no la ha ejecutado, el programa no advierte tal error. Sólo aparece el error cuando se ejecuta una instrucción mal construida.

Por otro lado, pueden existir errores lógicos del propio programa; cuando se producen, el programa también se detiene y permite la posibilidad de corregirlos.

- **Lenguajes compiladores.** Este tipo de lenguajes es más complejo que los anteriores.

En primer lugar, se escribe el **código fuente** mediante cualquier editor, gráfico o de texto, incluido en el propio lenguaje de programación o independiente de éste.

Terminada esta fase, se realiza un proceso de **compilación** en el que se depura todo el programa de los posibles errores sintácticos. Los errores lógicos no se pueden eliminar en esta fase.

Así, hasta que el programa no esté totalmente corregido de errores sintácticos no podremos pasar a la siguiente fase.

Una vez corregidos los errores sintácticos, se obtiene un programa o módulo llamado **objeto**. Este módulo depurado se encadena mediante los **editores de vínculos** para obtener el programa **ejecutable**.

En este punto, al ejecutar el programa pueden aparecer los errores lógicos, pero nunca errores sintácticos. Si es así, cuando se corrija el error lógico tendremos que empezar desde el principio; es decir, pasar el programa fuente a objeto, y éste a ejecutable.

Existen diferencias entre lenguajes compiladores e intérpretes. Los primeros, una vez puestos a punto, son más rápidos que los intérpretes, ya que estos últimos siempre tendrán que analizar previamente la instrucción que se va a ejecutar.

La puesta a punto de un programa que utilice un lenguaje compilador es más lenta, pero más fiable, pues no habrá errores sintácticos.

Los errores de ejecución se corrigen mejor en lenguajes intérpretes que en compiladores; al menos, se ven más rápido y en la instrucción precisa en la que se han producido.



1. Introducción a los sistemas informáticos

1.5 Almacenamiento externo

Normativa legal sobre el uso del software

Debido al auge alcanzado por las aplicaciones informáticas, se han promulgado leyes que sirven para proteger a los autores del software de la denominada **piratería informática**.

En España, en 1992, se publicó la Ley orgánica de Regulación del Tratamiento Automático de Datos (LORTAD), que incluye más de cuarenta artículos acerca de distribución, diseño y protección del software. En la actualidad, la LORTAD ha quedado derogada por la LOPD (Ley Orgánica de Protección de Datos).

1.5 Almacenamiento externo

El almacenamiento externo de datos se realiza sobre los soportes de información, que pueden ser disquetes, discos duros, CD-ROM, DVD, cintas, etcétera.

Los soportes de información se definen como cualquier medio físico capaz de registrar información de forma magnética, óptica o mediante otro método. Una disquetera (unidad de disquetes) no es un soporte. La disquetera sirve para grabar o leer información sobre el soporte, que es el disquete.

Los soportes de información se clasifican según el modo de acceso a la información:

- **Secuenciales.** Se accede a la información deseada pasando previamente por la anterior. Algunos ejemplos son las cintas DAT, los streamer y cualquier otro dispositivo de cinta (DLT).
- **Directos.** Se accede a la información de forma directa, sin tener que pasar por otra información anterior. Son los disquetes, discos duros, CD-ROM, etcétera.

Los disquetes, como soportes básicos, pueden ser de varias medidas y capacidades. Los disquetes de 3½ pulgadas son capaces de almacenar 720 Kb en formato de baja densidad y 1 440 Kb (1,44 Mb) en formato de alta densidad. Existen disquetes de densidad extra, con una capacidad de 2,88 Mb, pero están poco extendidos, ya que necesitan unidades periféricas específicas para ser grabados y leídos. Los de 1,44 Mb son los más utilizados.

Los discos duros son soportes que, con una estructura similar a la de los disquetes, permiten almacenar más información que éstos. Posteriormente en este libro se describen con más detalle.

Tanto en los disquetes como en los discos duros, la información se puede leer y escribir, es decir, son soportes que permiten que se modifique la información que contienen. Esto se debe a que son magnéticos, como cualquier casete o cinta de video.

Los discos ópticos, CD-ROM o DVD, almacenan la información mediante técnicas ópticas. Como ya se ha comentado, su capacidad se mide en múltiplos de bytes. Estos soportes, una vez grabados, solamente pueden ser leídos; es decir, no son reescribibles como los discos duros o disquetes.

No obstante, existen CD-ROM y DVD grabables y regrabables; para ello son necesarias las correspondientes unidades periféricas.

Los discos magnetoópticos utilizan una técnica mixta entre la magnética y la óptica para registrar la información. Son de elevada capacidad y permiten reescribir información sobre ellos; o sea, son soportes reutilizables.

Las cintas DAT son como cintas de casete, pero de tamaño inferior, que permiten almacenar de forma secuencial la información. Estas cintas se utilizan normalmente para realizar copias de seguridad, ya que son mucho más lentas que cualquier soporte directo, pero de mayor capacidad.