

# Fundamentos tecnológicos de la sociedad de la información

David Bañeres Besora (coordinador)

Antoni Martínez Ballesté

Gregorio Robles Martínez

PID\_00150198

Material docente de la UOC



Universitat Oberta  
de Catalunya

[www.uoc.edu](http://www.uoc.edu)



FONDO SOCIAL  
EUROPEO



plan  
avanza...



**David Bañeres Besora**

Ingeniero de Informática por la UPC. Doctor en Informática por la UPC. Profesor de los Estudios de Informática, Telecomunicaciones y Multimedia de la UOC.

**Antoni Martínez Ballesté**

Ingeniero de Informática por la Universidad Rovira i Virgili. Doctor en Telemática por la Universidad Politécnica de Cataluña. Profesor del Departamento de Ingeniería Informática y Matemáticas de la Universidad Rovira i Virgili. Consultor de la Universitat Oberta de Catalunya.

**Gregorio Robles Martínez**

Ingeniero de Telecomunicación por la UPM. Doctor en Ingeniería de Telecomunicación por la URJC. Profesor de Ingeniería telemática de la URJC. Consultor de la Universitat Oberta de Catalunya.

*El proyecto E-ALQUIMIA ha sido apoyado por el Ministerio de Industria, Turismo y Comercio en el marco de las ayudas para la realización de actuaciones sobre contenidos digitales en el marco del Plan Avanza, y por la Unión Europea a través de los Fondos Comunitarios.*

Primera edición: febrero 2010  
© Antoni Martínez Ballesté, Gregorio Robles Martínez  
Todos los derechos reservados  
© de esta edición, FUOC, 2010  
Av. Tibidabo, 39-43, 08035 Barcelona  
Diseño: Manel Andreu  
Realización editorial: Eureka Media, SL  
ISBN: 978-84-692-9411-6  
Depósito legal: B-1.852-2010



Los textos e imágenes publicados en esta obra están sujetos –excepto que se indique lo contrario– a una licencia de Reconocimiento-NoComercial-SinObraDerivada (BY-NC-ND) v.3.0 España de Creative Commons. Podéis copiarlos, distribuirlos y transmitirlos públicamente siempre que citéis el autor y la fuente (FUOC. Fundació per la Universitat Oberta de Catalunya), no hagáis de ellos un uso comercial y ni obra derivada. La licencia completa se puede consultar en <http://creativecommons.org/licenses/by-nc-nd/3.0/es/legalcode.es>

## Contenidos

### Módulo didáctico 1

#### **Aspectos tecnológicos de los sistemas informáticos**

Antoni Martínez Ballesté y Gregorio Robles Martínez

1. Perspectiva histórica de la informática
2. Tratamiento de la información
3. Componentes del sistema informático
4. Tipo de ordenador
5. Sistemas de información

### Módulo didáctico 2

#### **Aspectos tecnológicos de las redes e Internet**

Antoni Martínez Ballesté

1. Las redes de computadores
2. La transmisión de información
3. Las redes de área local
4. Internet
5. El correo electrónico

### Módulo didáctico 3

#### **Seguridad en la sociedad de la información**

Antoni Martínez Ballesté

1. La seguridad en las redes y en Internet
2. Seguridad en los equipos informáticos
3. Seguridad en la información
4. Identidad digital
5. Privacidad e Internet
6. Seguridad en la gestión audiovisual

### Módulo didáctico 4

#### **La World Wide Web**

Gregorio Robles Martínez

1. Historia
2. HTTP: cliente y servidor
3. El protocolo HTTP
4. El lenguaje HTML. Nociones básicas
5. La web dinámica o interactiva y aplicaciones web
6. Web 2.0
7. La web multimedia

### Módulo didáctico 5

#### **Cultura libre**

Gregorio Robles Martínez

1. Definición de obra libre y sus consecuencias

2. Fundamentos de Propiedad Intelectual
3. Tipos de obras según su licencia
4. Historia y definición del software libre
5. Licencias de software libre
6. Categorías de licencias
7. Licencias para otros tipos de contenidos



# Aspectos tecnológicos de los sistemas informáticos

Antoni Martínez Ballesté  
Gregorio Robles Martínez

PID\_00150272



Universitat Oberta  
de Catalunya

[www.uoc.edu](http://www.uoc.edu)



Los textos e imágenes publicados en esta obra están sujetos –excepto que se indique lo contrario– a una licencia de Reconocimiento-NoComercial-SinObraDerivada (BY-NC-ND) v.3.0 España de Creative Commons. Podéis copiarlos, distribuirlos y transmitirlos públicamente siempre que citéis el autor y la fuente (FUOC. Fundació per la Universitat Oberta de Catalunya), no hagáis de ellos un uso comercial y ni obra derivada. La licencia completa se puede consultar en <http://creativecommons.org/licenses/by-nc-nd/3.0/es/legalcode.es>

# Índice

|                                                                 |           |
|-----------------------------------------------------------------|-----------|
| <b>Introducción.....</b>                                        | <b>5</b>  |
| <b>Objetivos.....</b>                                           | <b>7</b>  |
| <b>1. Perspectiva histórica de la informática.....</b>          | <b>9</b>  |
| 1.1. La electrónica y los primeros computadores .....           | 10        |
| 1.2. El ordenador personal .....                                | 12        |
| 1.3. La informática, en todas partes y para todo el mundo ..... | 12        |
| <b>2. Tratamiento de la información.....</b>                    | <b>14</b> |
| 2.1. El mundo binario .....                                     | 14        |
| 2.1.1. Los números en binario .....                             | 15        |
| 2.1.2. Los caracteres .....                                     | 17        |
| 2.1.3. Los múltiplos del byte y del bit .....                   | 17        |
| 2.2. El tratamiento lógico de la información .....              | 18        |
| <b>3. Componentes del sistema informático.....</b>              | <b>20</b> |
| 3.1. La estructura del hardware .....                           | 20        |
| 3.1.1. El microprocesador .....                                 | 21        |
| 3.1.2. La memoria .....                                         | 22        |
| 3.1.3. La placa base .....                                      | 24        |
| 3.1.4. Dispositivos de entrada y salida de información .....    | 24        |
| 3.1.5. Dispositivos de almacenamiento de información .....      | 25        |
| 3.2. El software .....                                          | 26        |
| 3.2.1. Algoritmos .....                                         | 28        |
| 3.2.2. Diseño del software .....                                | 29        |
| 3.2.3. Evolución de los lenguajes de programación .....         | 32        |
| 3.3. Tipos de software .....                                    | 35        |
| 3.3.1. El software de sistemas: el sistema operativo .....      | 36        |
| 3.3.2. Software para usuario final .....                        | 39        |
| <b>4. Tipo de ordenador.....</b>                                | <b>42</b> |
| 4.1. Los ordenadores principales .....                          | 42        |
| 4.2. Los microcontroladores .....                               | 43        |
| 4.3. Los superordenadores .....                                 | 43        |
| 4.4. Multicomputadores .....                                    | 43        |
| 4.5. Los ordenadores portátiles .....                           | 44        |
| <b>5. Sistemas de información.....</b>                          | <b>45</b> |
| <b>Resumen.....</b>                                             | <b>47</b> |

|                                          |           |
|------------------------------------------|-----------|
| <b>Actividades.....</b>                  | <b>49</b> |
| <b>Ejercicios de autoevaluación.....</b> | <b>49</b> |
| <b>Solucionario.....</b>                 | <b>51</b> |
| <b>Glosario.....</b>                     | <b>52</b> |
| <b>Bibliografía.....</b>                 | <b>54</b> |

## Introducción

En este módulo estudiaremos los aspectos tecnológicos de los ordenadores y los sistemas informáticos. El ordenador se ha convertido en el elemento que vehicula el conocimiento dentro de la sociedad de la información. Los sistemas informáticos están formados por uno o varios ordenadores, de diferentes tipos y con diferentes propósitos (ordenadores personales para trabajos de oficina, computadores que forman parte de entornos industriales, etc.). Sobre estos ordenadores se ejecuta una gran diversidad de programas. En muchos sistemas informáticos, el software incluye sistemas de información para el almacenamiento y gestión de datos.

Los términos *computación* e *informática* definen el propósito inicial de los ordenadores: el cálculo y el tratamiento automático de la información. A pesar de ser términos sinónimos, la palabra *informática* tiene hoy día un sentido más generalista que *computación*, ya que la informática ha permitido el uso al público en general de toda una serie de potentes herramientas que facilitan multitud de tareas, y también ha facilitado una comunicación sin barreras y el acceso a cantidades ingentes de información.

Las palabras *computador* y *ordenador* también tienen significados ligeramente distintos. Un computador tendría una utilidad no tan generalista como un ordenador. Normalmente, los computadores designan los primeros ordenadores (destinados a hacer cálculos y cálculos), mientras que el término *ordenador* se aplica a los computadores actuales. Sin embargo, en inglés el término *computer* sirve para definir tanto computadores como ordenadores.

Los ordenadores empezaron como herramientas de cálculo para facilitar tareas rutinarias como operaciones aritméticas o grandes procesos de cálculo, por ejemplo, la gestión de censos. En el apartado 1 del módulo, conoceremos los hitos más importantes en la historia de la informática y los ordenadores. En el apartado 2 del módulo, veremos cómo tratan los ordenadores la información en el mundo digital.

En el apartado 3 del módulo, distinguiremos las diferentes partes que forman un sistema informático. Básicamente, veremos que hay una parte tangible, llamada hardware, que sirve de soporte a la parte intangible de la informática: los programas y la información. También definiremos este componente flexible de los ordenadores: el software. Veremos qué es el software y aprenderemos cómo se crea. Así, conoceremos cómo se transforma un programa desde la especificación hasta un programa que se pueda ejecutar en un ordenador.

Finalmente, en el apartado 4 del módulo, presentaremos los tipos de ordenadores actuales que podemos encontrar, en el último apartado, y los sistemas de información, que se utilizan de manera generalizada en el mundo empresarial y gubernamental.

## Objetivos

Los objetivos que el estudiante habrá alcanzado al finalizar este módulo son:

- 1.** Tener una perspectiva histórica de la informática.
- 2.** Comprender las bases del tratamiento de la información por parte de los ordenadores.
- 3.** Distinguir las partes fundamentales de un sistema informático y entender cuáles son sus objetivos.
- 4.** Entender los conceptos más importantes relacionados con el software, incluidas todas sus etapas de creación desde la especificación en algoritmos hasta su implementación con un lenguaje de programación de alto nivel.
- 5.** Conocer la evolución de los lenguajes de programación.
- 6.** Conocer los diferentes tipos de software: el software de sistemas (del que el más significativo es el sistema operativo) y el software de usuario final, con especial énfasis en los sistemas de información.





## 1. Perspectiva histórica de la informática

Los inicios de la informática son más bien difusos. Lo que hoy conocemos como ordenadores e informática es el resultado del avance de varias disciplinas, en especial las matemáticas, la electrónica y la física. Sin embargo, los esfuerzos de la humanidad por tener artilugios que ayudaran con los cálculos se remontan a épocas antiguas.

Los inicios de la informática van ligados a la invención de diferentes utensilios y artilugios dedicados a facilitar el cálculo matemático.

Unos de los más antiguos es el ábaco, originario de la China del siglo V a. C. Se trata de un marco de madera con ejes sobre los que se deslizan bolas. Los ábacos facilitan la realización de sumas, restas, multiplicaciones y divisiones. Este utensilio fue adaptado y utilizado por muchas otras culturas.

Durante los siglos XVII y XVIII, el avance de la mecánica propicia el diseño e incluso la producción de diferentes máquinas de calcular. Una de las más célebres es la Pascalina, creada en el año 1645 por Blaise Pascal y de la que se produjeron una cincuentena de copias.



La Pascalina, una de las primeras calculadoras. Expuesta en el Musée des Arts et Métiers de París.

En el siglo XIX tiene lugar el primer diseño de lo que se puede considerar un computador programable y no una mera máquina de calcular: la máquina analítica, de Charles Babbage.

La máquina analítica de Babbage estaba ideada para poder implementar pequeños programas de cálculo, mostrar la salida y guardar datos en una memoria.

Desgraciadamente, su inventor no la pudo ver realizada por una serie de problemas, sobre todo porque la ingeniería de la época aún no estaba preparada para llevarla a la práctica. Ada Lovelace, hija de Lord Byron, estudió el funcionamiento de esta máquina e incluso propuso métodos para definir procedimientos a partir de las operaciones que la máquina permitiría realizar. Ada Lovelace es considerada la primera persona que trabajó en los conceptos de programación de ordenadores.

Aquí empieza lo que se considera la historia de los ordenadores. Acto seguido, veremos sus hitos más importantes. Pero antes describiremos una división clásica de las etapas de la historia de los ordenadores, que pasa por diferentes generaciones:

- La primera generación comprende los grandes ordenadores electromecánicos, en los que la electrónica se controlaba mediante válvulas y los circuitos eran cableados.
- La segunda generación se inicia con la introducción del transistor y la consecuente reducción de tamaño de los equipos. Un módulo basado en componentes electrónicos del tamaño de una bombilla ahora se podría diseñar con componentes no mayores que una mosca.
- La tercera generación supone unos ordenadores aún más pequeños debido a la aparición de los microchips. Así pues, un circuito cableado de dimensiones considerables comparables al tamaño de una mesa de comedor se podía reducir al tamaño de un sello de correos.
- La cuarta generación de los ordenadores comprende los ordenadores personales. La miniaturización iniciada con la tercera generación aún es más evidente, ya que los circuitos integrados pueden incluir diferentes circuitos, incluso sobrepuestos en diferentes capas, para poder realizar varias tareas.

Se considera que la generación actual de ordenadores es la quinta. En esta generación se introducen conceptos como el multiprocesador y la memoria caché.

### 1.1. La electrónica y los primeros computadores

La llegada de la electrónica a mediados del siglo XX propició la materialización de diseños que, con tecnologías puramente mecánicas, no eran realizables. Así pues, durante la década de los años cuarenta vieron la luz los primeros computadores electromecánicos. Estos ordenadores, grandes logros de la ingeniería de la época, eran muy diferentes de los ordenadores que conocemos hoy en día: eran máquinas de gran tamaño, alimentadas por una potencia eléctrica considerable y generadoras de ruido y altas temperaturas. Para controlar los voltajes e intensidades que regulaban los módulos de procesamiento de información se utilizaban válvulas, como las que había dentro de las radios antiguas.

En el año 1941, se construyó el ordenador Z3, que se presentó en Berlín. Desgraciadamente, este equipo quedó destruido en un bombardeo durante la Segunda Guerra Mundial. Por ello, la popularidad la ganó el Mark I, diseñado y construido en la Universidad de Harvard. Fue presentado en el año 1944. Le siguió el ENIAC<sup>(1)</sup>, presentado en el año 1946 en la Universidad de Pennsylvania. Ambas máquinas tenían grandes dimensiones y ocupaban una sala entera. Elevaban la temperatura ambiente y necesitaban una gran cantidad de energía para arrancar. La diferencia entre el ENIAC y el Mark I es que con el primero se interactuaba por medio de cables y tableros de conmutación, mientras que

<sup>(1)</sup>ENIAC es el acrónimo de *electronic numerical integrator and calculator*, en español, calculadora e integrador numérico electrónico.

el segundo disponía de lectores de cinta perforada. Esta técnica de almacenamiento ya se utilizaba para las partituras de las pianolas automáticas y las primeras calculadoras como la máquina analítica.

### Las máquinas de calcular analógicas

Aunque la gran mayoría de ordenadores de estas épocas se basan en el tratamiento digital de la información, durante el siglo XX también se desarrollaron máquinas de calcular analógicas. En éstas, por ejemplo, para realizar una operación de suma se sumaban dos voltajes diferentes. El resultado se mostraba en un voltímetro.

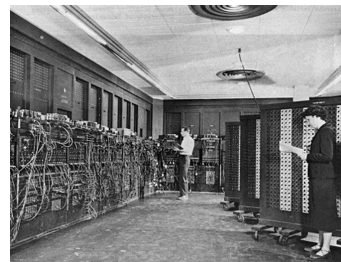
Durante los años cincuenta, diferentes empresas fabricaron grandes ordenadores como producto comercial. El primer computador comercial fue el UNIVAC<sup>2</sup> I. De estas máquinas, que ocupaban unos 30 m<sup>2</sup> cada una, se vendieron unas 40 copias a un millón de dólares. IBM entró en el mercado de los ordenadores haciendo máquinas considerablemente más pequeñas y, en consecuencia, más económicas. Uno de los primeros modelos populares a partir del año 1954 fue el IBM 650, que pesaba unos 900 kilogramos de peso y necesitaba un espacio de unos dos metros cúbicos.

A finales de esta década, la aparición del transistor y los circuitos integrados facilitaron la reducción de tamaño de los computadores, a la vez que disminuyeron el consumo energético. Paralelamente, van apareciendo equipos con entrada y salida de datos, ya que como hemos mencionado anteriormente los primeros computadores había que programarlos mediante cables y enchufes (como es el caso del ENIAC) y la salida se representaba en un panel de bombillas. Las pantallas de televisión y las máquinas de escribir se adaptaron para facilitar la interacción de los operadores con las máquinas.

En los inicios de la informática, cada fabricante utilizaba sus sistemas de representación de información y sus sistemas de programación de ordenadores. Durante los años sesenta y setenta del siglo XX, se hicieron esfuerzos a fin de que los ordenadores se miniaturizaran y utilizaran estándares tanto en software como en la representación de la información e incluso desde el punto de vista de la comunicación.

En este sentido, ven la luz los sistemas Unix, los lenguajes de programación generalistas para crear programas, e incluso aparecen los primeros intentos de interconexión de ordenadores en una incipiente Internet.

A finales de los años sesenta, aparece la ingeniería informática, especializada en el diseño de sistemas informáticos para gestionar los procesos de información. En la misma época, la creciente complejidad de la gestión de la información provoca que se ideen herramientas de ingeniería para el diseño de pro-



El ENIAC, uno de los primeros grandes ordenadores

<sup>(2)</sup>En inglés, *universal automatic compute* (computador automático universal).

### Ved también

Las redes de ordenadores se introducirán en el módulo "Aspectos tecnológicos de las redes e Internet".

gramas, empezando lo que llamamos ingeniería del software. Los conceptos teóricos sobre los modelos matemáticos y de procesamiento de información relacionados con los computadores dieron lugar a las Ciencias de la Computación.

## 1.2. El ordenador personal

Los años ochenta tienen como estrella la irrupción de la informática personal, gracias a la aparición de la cuarta generación de ordenadores. Hasta entonces, las empresas grandes y medianas ya contaban con ordenadores *mainframes* a los que varios trabajadores podían acceder mediante sus terminales. Los ordenadores personales IBM PC<sup>3</sup> y los Apple hicieron posible que muchas pequeñas empresas y particulares se adentraran en el mundo de la informática, ya sea por afición o para mejorar su productividad.

<sup>(3)</sup>Del inglés *Personal Computer*.

Los IBM PC utilizaban chips fabricados por Intel y un sistema operativo (lo que permite la ejecución de los programas) creado por Microsoft. Ambas empresas se convirtieron en las más importantes en el mundo de la informática. La compañía Apple se hizo popular con el modelo Macintosh, que aprovechaba unas ideas que tenían su origen en investigaciones de la empresa Xerox en los años setenta: el ratón y las ventanas. Estos conceptos fueron llevados al mundo de los IBM PC por Microsoft mediante el sistema operativo Windows.

La evolución de los ordenadores durante los años ochenta y noventa se produjo a una velocidad vertiginosa: por una parte, aparecen microordenadores como el ZX Spectrum, que popularizaron la informática a coste asequible. Estas máquinas no eran compatibles con los PC de IBM (es decir, los programas creados para estos ordenadores no podían funcionar en los otros). Muchas compañías diseñaron sus microordenadores sin tener en cuenta, en general, la compatibilidad entre productos de diferentes marcas.



Microordenador de la casa Spectrum. Como muchos ordenadores de su época, había que conectarlos al televisor de casa.

Por otra parte, Intel permitió que otras compañías utilizaran sus chips, con lo que centenares de empresas fabricaron ordenadores personales compatibles con el de IBM y sensiblemente más económicos que el original. Así pues, después de unos años en los que multitud de marcas y modelos ocupaban parte importante del mercado de la informática doméstica, los ordenadores compatibles con los IBM PC pasaron a ser los más vendidos.

## 1.3. La informática, en todas partes y para todo el mundo

La evolución de los ordenadores PC en cuanto a velocidad y capacidad, la aparición de los ordenadores portátiles, las impresoras de calidad fotográfica, las cámaras digitales, etc. son historias bastante conocidas por nosotros, dado que son recientes y podemos hacer memoria de ellas.

Los ordenadores personales han adquirido una popularidad notable, en parte gracias a su abaratamiento y en parte debido a las posibilidades que ofrece Internet como pieza integrante de cualquier sistema informático. El ordenador ya no es una herramienta exclusivamente dedicada a mejorar la productividad: el ocio personal, la gestión de contenidos multimedia, el comercio electrónico, etc., son hechos del día a día vehiculados por los ordenadores personales.

**One Laptop Per Child Association**

Varias asociaciones, como One Laptop Per Child Association, promueven la fabricación y distribución de equipos informáticos para países en vías de desarrollo.

Los ordenadores portátiles se han convertido en un complemento al ordenador de sobremesa, sustituyéndolo en muchos casos. La miniaturización de los ordenadores ha permitido la fabricación y popularización de teléfonos móviles (que en el fondo son pequeños ordenadores), aparatos digitales de grabación de vídeo (que también lo son), coches controlados y gestionados por un sistema informático de a bordo, etc.

Personas de cualquier nivel cultural tienen acceso a las herramientas informáticas que les permiten formar parte de la sociedad de la información. La adquisición de información y su tratamiento automático se han convertido en universales en todos los ámbitos, al menos en lo que llamamos primer mundo.

## 2. Tratamiento de la información

Los ordenadores, y la gran mayoría de aparatos electrónicos, trabajan en lo que se llama **mundo digital**. En contraposición, el mundo real es analógico. En el mundo analógico, las fuerzas y señales que encontramos (como la gravedad, la electricidad, el sonido, los colores, etc.) son continuas, en contraposición al mundo digital, en el que las medidas son discretas.

Eludiendo una descripción basada en las matemáticas, pondremos un ejemplo. Supongamos que queremos medir la longitud de una mesa. Si utilizamos un sistema continuo de medida, utilizaremos toda la precisión necesaria para calcular el tamaño exacto de la mesa.

En el mundo analógico, todas las mesas tienen longitudes diferentes, aunque a veces éstas difieran en millonésimas de milímetro.

En un mundo digital, las medidas y las señales son discretas, es decir, tienen una determinada precisión que especifica un conjunto de valores. Por lo tanto, no puede haber infinitos valores diferentes aparte de los que defina una determinada precisión.

En un mundo digital con una precisión de centímetros, todas las mesas que en el "mundo analógico" tienen 1.200, 1.201, 1.202, 1.203 y hasta 1.209 milímetros se considerarán de 120 centímetros.

En este apartado veremos cómo se trata la información en el mundo digital.

### 2.1. El mundo binario

Los circuitos electrónicos en los que se basan los computadores funcionan a partir de intensidades de corriente eléctrica. Así pues, un transistor puede estar emitiendo corriente o no hacerlo. De la misma manera, los sistemas primitivos de almacenamiento de información en tarjetas perforadas utilizaban agujeros para guardar los datos: o hay agujero o no lo hay.

Todo esto responde a lo que se llama *Álgebra de Boole*, ideada por el matemático George Boole en el siglo XIX. Las operaciones complejas de cálculo se pueden reducir a **operaciones lógicas binarias**, que trabajan sobre dos valores posibles (por ejemplo, el cero o el uno; el agujero o la ausencia de agujero). En los primeros ordenadores, estas operaciones se llevaban a cabo con diferentes módulos de cálculo que se basaban, en general, en las sencillas operaciones lógicas del álgebra de Boole. La representación de números en el sistema binario,

#### Los ordenadores cuánticos

Los ordenadores cuánticos, actualmente en fase de investigación, permitirían trabajar con más valores que los dos del Álgebra de Boole.

es decir, en ceros y unos, hizo posible que se pudiera tratar automáticamente la información con estos módulos de cálculo. Hoy en día, los ordenadores tienen la misma base de funcionamiento.

Los ordenadores tratan la información utilizando el sistema binario, formado por unos y ceros. Un **bit** es un dígito binario, es decir, uno de estos dos valores. En consecuencia, un bit es la cantidad mínima de información que puede tratar un ordenador.

Sin embargo, los módulos de cálculo y los actuales chips no realizan los cálculos sólo con un bit. Normalmente se utilizan grupos de bits, por ejemplo, 4, 8, 16, 32 o 64 bits. Lo más habitual históricamente fue trabajar con grupos de 8 bits, llamados *bytes*.

### 2.1.1. Los números en binario

Cualquier número se puede representar en binario. Por ejemplo, aquí tenéis todos los números naturales que se pueden representar con tres bits:

| Nº binario | Nº natural |
|------------|------------|
| 000        | 0          |
| 001        | 1          |
| 010        | 2          |
| 011        | 3          |
| 100        | 4          |
| 101        | 5          |
| 110        | 6          |
| 111        | 7          |

La cantidad de números naturales que se puede representar dependerá del número de bits. En concreto, con  $n$  bits se pueden representar  $2^n$  números diferentes, del 0 al  $2^n - 1$ .

#### El byte

Como los módulos de cálculo podían trabajar con la información a base de "comérsela a mordiscos de 8 bits", el grupo de 8 bits se llama *byte*, que suena igual que *bite* (mordisco en inglés).

### El número decimal que representa un número en binario

Para obtener el número decimal que representa un número en binario hay que multiplicar el dígito (cero o uno) por un valor que corresponde a la base (en este caso, dos para ser el sistema binario) elevada al número que indica la posición del dígito, entendiendo que la posición de más a la derecha es la posición cero. Para entenderlo mejor, veamos un ejemplo. En el caso de los tres bits, los valores de cada uno de los dígitos de izquierda a derecha son:  $4(2^2)$ ,  $2(2^1)$  y  $1(2^0)$ . Por lo tanto, el número binario 100 corresponde al número decimal 4, ya que  $1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 = 4$ . De la misma manera, el número 111 corresponde al número 7, ya que  $1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 7$ .

Fijémonos en que eso también sucede con el sistema decimal que utilizamos los humanos, en el que la base es 10. Por ejemplo, el número 2.009 se puede descomponer en varias potencias:  $2 \times 10^3 + 0 \times 10^2 + 0 \times 10^1 + 9 \times 10^0$ .

#### Origen del sistema decimal

Los antropólogos afirman que el origen del sistema decimal está en los 10 dedos que siempre nos han servido de base para contar.

En el caso de los números enteros (es decir, los que pueden ser positivos o negativos), existen diferentes alternativas para representarlos en binario. Una de ellas sería utilizar uno de los bits para decir si el número es positivo o negativo. En el caso anterior, se necesitaría un bit de más. ¡Fijaos en que aparece el caso extraño de que hay dos representaciones posibles del cero!

| Números positivos |            | Números negativos |            |
|-------------------|------------|-------------------|------------|
| Nº binario        | Nº decimal | Nº binario        | Nº decimal |
| 0000              | 0          | 1000              | -0         |
| 0001              | 1          | 1001              | -1         |
| 0010              | 2          | 1010              | -2         |
| 0011              | 3          | 1011              | -3         |
| 0100              | 4          | 1100              | -4         |
| 0101              | 5          | 1101              | -5         |
| 0110              | 6          | 1110              | -6         |
| 0111              | 7          | 1111              | -7         |

Gracias a la representación de números en el sistema binario fue posible crear máquinas digitales de cálculo, desde los primeros grandes ordenadores hasta las calculadoras de bolsillo.

El sistema binario no es el único que se utiliza en informática. También es habitual el llamado sistema hexadecimal, en el que la base es 16. El objetivo de este sistema es representar números binarios de manera más compacta, utilizando menos dígitos que si los representamos con ceros y unos.

| Número decimal | Número binario | Número hexadecimal | Número decimal | Número binario | Número hexadecimal |
|----------------|----------------|--------------------|----------------|----------------|--------------------|
| 0              | 0000           | 0                  | 8              | 1000           | 8                  |
| 1              | 0001           | 1                  | 9              | 1001           | 9                  |



| Número decimal | Número binario | Número hexadecimal | Número decimal | Número binario | Número hexadecimal |
|----------------|----------------|--------------------|----------------|----------------|--------------------|
| 2              | 0010           | 2                  | 10             | 1010           | A                  |
| 3              | 0011           | 3                  | 11             | 1011           | B                  |
| 4              | 0100           | 4                  | 12             | 1100           | C                  |
| 5              | 0101           | 5                  | 13             | 1101           | D                  |
| 6              | 0110           | 6                  | 14             | 1110           | E                  |
| 7              | 0111           | 7                  | 15             | 1111           | F                  |

### 2.1.2. Los caracteres

Los primeros ordenadores estaban dedicados a realizar grandes cálculos, pero la transmisión y el almacenamiento de información digital precisaron de la necesidad de codificar esta información. Por ejemplo, los textos enviados mediante un sistema de transmisión se debían codificar con bits.

Uno de los sistemas de codificación de caracteres, y el más importante en la actualidad, es el ASCII<sup>4</sup>. Un código ASCII de 8 bits puede representar 256 símbolos diferentes. Por ejemplo, la letra "A" tiene asignado el código 65 (que en binario es 01000001), la cifra "1" tiene el código 49, el interrogante tiene el código 63, etc.

#### La codificación de caracteres

La frase "la vida es bella" ocupa en el ordenador un total de 16 bytes (13 letras más los tres espacios en blanco). El número 2009, si se trata como una cadena de caracteres, ocupará 4 bytes (32 bits), uno para cada cifra, pero si el número se trata como un número entero, ocupará 11 bits, ya que se utilizará su valor en binario: 11111011001.

Con el aumento de la transmisión de información con diferentes lenguas con el código ASCII aparece un problema: no se pueden representar todos los sistemas de escritura. Actualmente, se utiliza el código UNICODE (del inglés *universal code*) de 16 bits, donde se pueden representar incluso los símbolos de las lenguas asiáticas.

### 2.1.3. Los múltiplos del byte y del bit

Los sistemas informáticos de tercera y cuarta generación son capaces de tratar y almacenar millones de bytes de información. De esta manera, se han definido múltiplos del byte:

- Un **kilobyte** o KB son 1.024 bytes. También se llama "ka".
- Un **megabyte** o MB son 1.024 KB o 1.048.576 bytes. También se llama "mega".
- Un **gigabyte** o GB son 1.024 MB o 1.073.741.824 bytes. También se llama "giga".

#### Código Morse

El uso de la codificación es anterior a los propios ordenadores. Un ejemplo lo encontramos en el código Morse, en el que diferentes símbolos (letras y cifras, básicamente) se representan en un código de puntos y rayas.

<sup>(4)</sup>Del inglés *american standard code for information interchange*, código estándar americano para el intercambio de información.

- Un **terabyte** o TB son 1.024 GB o 1.099.511.627.776 bytes. También se llama "tera".

### Ejemplos sobre cuánto pueden ocupar algunos elementos habituales

Una carta hecha con un procesador de texto puede ocupar 25 KB. No sólo se guardan los caracteres que la forman, sino también información adicional como el tipo de letra, el color del texto, posibles imágenes integradas, etc. Una foto de alta calidad puede ocupar 2 MB. Una canción para ser escuchada en un reproductor MP3 ocupa unos 4 MB. Una película en DVD puede ocupar unos 6 GB.

#### Ved también

Los formatos multimedia (imagen, vídeo y sonido) se explicarán en el módulo "La World Wide Web".

¿Por qué múltiplos de 1.024 y no de 1.000? Pues por razones estrictamente relacionadas con el sistema binario y la construcción de los ordenadores. Sin embargo, la especificación de capacidad de soportes de almacenamiento, como los discos duros y los DVD, utilizan múltiplos de 1.000. Así pues, uno esperaría poder guardar 5.046.586.572 bytes en un DVD de 4,7 GB, cuando realmente caben 4.700.000.000 bytes.

Cuando se habla de **velocidades de transferencia de información**, hablamos de múltiplos de bit por segundo. Por ejemplo, tendremos los kilobits por segundo (kbit/s) o los megabits por segundo (Mbit/s). En estos casos, se utilizan múltiplos de 1.000.

#### Ved también

Trataremos las velocidades de transferencia de información en el módulo "Aspectos tecnológicos de las redes e Internet".

## 2.2. El tratamiento lógico de la información

Una vez que hemos visto cómo se representan los valores numéricos y los caracteres en los sistemas digitales y/o informáticos, veremos algunos conceptos básicos sobre cómo se diseñan los módulos de cálculo que ya se encontraban en los primeros computadores.

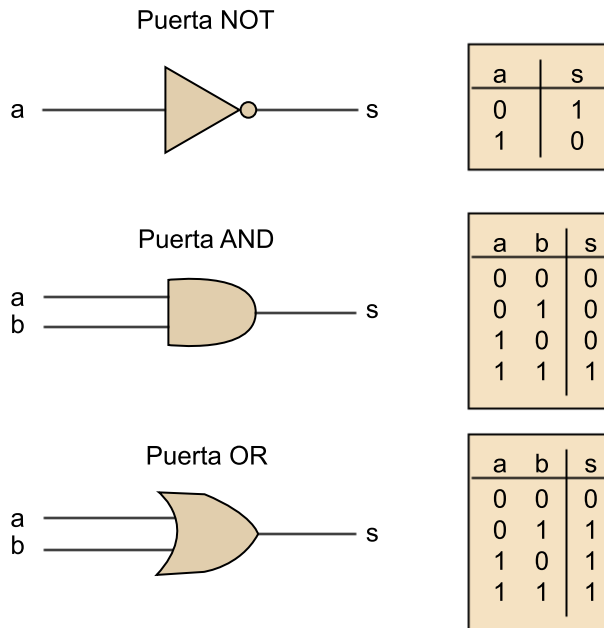
Hemos visto que el Álgebra de Boole se fundamenta en una serie de funciones lógicas que trabajan con bits. Usando estas funciones, se pueden hacer módulos sumadores, restadores, multiplicativos, etc. Las tres funciones lógicas más importantes son:

- La función **no**, llamada NOT, cambia el valor del bit que le entra. Por ejemplo, el valor 0 se convierte en 1 y el 1 en 0.
- La función **y**, llamada AND, tiene el valor 1 de salida cuando todos los valores de entrada son 1, y 0, en caso contrario.
- La función **o**, llamada OR, tiene valor 1 cuando alguna de las entradas vale 1, y 0, si todas las entradas valen 0.

Estas funciones tienen un símbolo que las representa y una **tabla de verdad** asociada. En la figura siguiente, se muestra el símbolo para las tres funciones lógicas que se acaban de definir, así como sus tablas de verdad.

#### Tabla de verdad

Una tabla de verdad específica, por las diferentes combinaciones de los bits de entrada, cuál es la salida de una función lógica o de un circuito de cálculo.



Arriba, la puerta lógica NOT; en el centro, la puerta AND, y abajo la tiene la puerta OR.

Otras funciones utilizadas son la XOR, la NAND o la NOR. La función XOR tiene 1 como salida sólo si una de las dos entradas es 1. La NAND y la NOR son los resultados de "negar" las salidas de las funciones AND y OR, respectivamente. Todas estas funciones se pueden implementar físicamente en un circuito mediante la combinación de dispositivos electrónicos, como transistores. Al encadenar estos dispositivos, se consiguen funciones más complejas diseñadas a partir de las básicas y, en consecuencia, se pueden hacer módulos para cálculos complejos.

Los módulos de cálculo están diseñados para trabajar con un número concreto de bits, por ejemplo, 8 bits. Al encadenar módulos de éstos, se consiguen módulos capaces de procesar 16, 32 o 64 bits.

Para realizar operaciones más complejas que la suma o la resta, por ejemplo, una división o una raíz cuadrada, existen dos alternativas:

- Diseñar bloques específicos para llevar a cabo estas operaciones. Esto implicaría tener muchos módulos que a menudo no se estarían utilizando. Algunos de los primeros grandes ordenadores seguían esta alternativa.
- Hacer un único módulo capaz de realizar operaciones "complejas" a partir de la definición de procedimientos que utilicen un conjunto reducido de módulos de cálculo sencillos. Por ejemplo, hacer la resta a partir del módulo suma con el segundo operando cambiado de signo, hacer una división mediante restas, etc. Ésta es la tendencia que ha marcado el diseño de los ordenadores modernos. Este módulo de cálculo se llama *procesador*.

### 3. Componentes del sistema informático

Una vez hemos visto cómo los ordenadores y los sistemas digitales tratan la información, estudiaremos los componentes del sistema informático. Un sistema informático está formado por dos tipos de componentes:

- **Componentes de hardware.** Se trata de todos los componentes físicos y electrónicos que configuran los sistemas informáticos. Dentro de estos componentes encontramos, entre otros, el procesador, el teclado, el lector de DVD, etc.
- **Componentes de software.** Forman parte de él todos los programas que funcionan en el ordenador y los datos que éstos manejan. Se trata, pues, de la parte "intangible" del sistema informático.

Un elemento importante del software es el **sistema operativo**, que crea una "capa" intermedia entre el hardware y el software con el objetivo de facilitar la interacción del software y el usuario con el hardware. Un caso concreto de software es el que se encuentra registrado en los propios chips del hardware, por ejemplo, en los dispositivos móviles como teléfonos o GPS, que recibe el nombre de **firmware**.

#### 3.1. La estructura del hardware

La estructura del hardware del ordenador se basa en la llamada arquitectura Von Neumann. John Von Neumann, un matemático prolífico, propuso que los ordenadores deberían estar gobernados por una **unidad de control**, que se encargaría de transportar información desde la memoria a una **unidad de cálculo** (llamada unidad aritmético-lógica) y devolverla a la **memoria** una vez acabadas las operaciones. Definió, además, unos **módulos para la entrada y salida** de información, así como tres caminos (o **buses**) por los que circularían diferentes tipos de información: los datos con los que trabajar, el nombre de la instrucción que debía ejecutar la unidad **aritmético-lógica** y la posición en la memoria en la que se ubica la información y se guardarán los resultados. Las instrucciones que conforman el procedimiento de operaciones que se deben realizar también se almacena en la memoria.

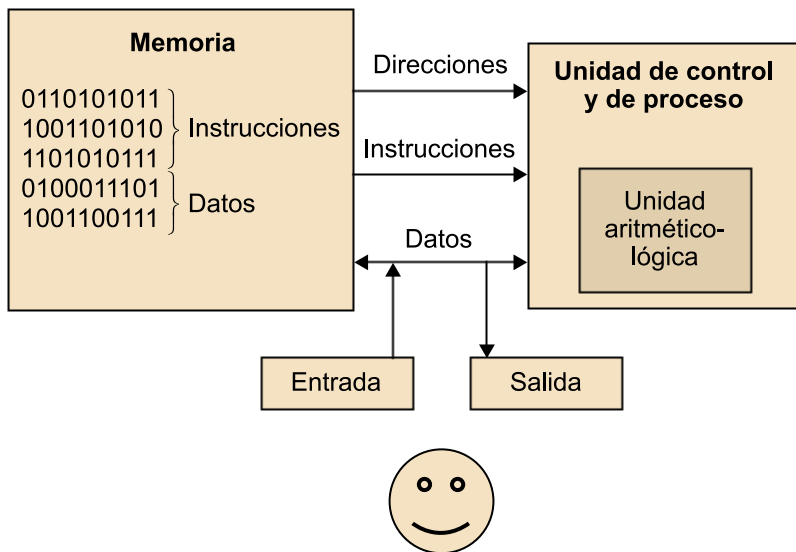
#### Ved también

Se definirá con más detalle el software en el subapartado 3.2 de este módulo.

Mediante las instrucciones, los programadores definen lo que deben hacer los ordenadores durante la ejecución de los programas.

La siguiente figura muestra un esquema de esta arquitectura.

## Arquitectura Von Neumann



Incluso los ordenadores actuales se basan en esta propuesta, a pesar de haber evolucionado y haberse añadido nuevos elementos.

### 3.1.1. El microprocesador

La unidad de control y de proceso<sup>5</sup> es el elemento central de los **microprocesadores**. El microprocesador es el encargado de la preparación del hardware para la ejecución de tareas, así como de controlar la ejecución y ubicar los resultados en el lugar especificado en las instrucciones. Uno de los factores a la hora de valorar el rendimiento de un ordenador es medir las prestaciones del microprocesador.

<sup>(5)</sup>En inglés, *central processing unit* (CPU).

Entre los factores que permiten evaluar el **rendimiento de un microprocesador** destacamos la velocidad, el número de bits con los que es capaz de trabajar o el número de operaciones sobre números decimales capaz de hacer en un segundo.

A continuación, haremos una descripción más detallada. La **velocidad del microprocesador** depende de la frecuencia de funcionamiento del reloj que éste incorpora. Esta velocidad se mide en hercios y durante decenios no se llegó a superar el megahercio (o MHz, es decir, el millón de hercios). En la última década del siglo XX, se superó la barrera del gigahercio (o GHz) y las frecuencias actuales tienen dificultades tecnológicas para ser superadas.

El **número de bits de trabajo** también es un factor determinante del rendimiento.

### Ejemplo sobre el número de bits de trabajo

Un procesador de 8 bits podrá hacer operaciones con números de 8 bits (por ejemplo, números enteros entre -127 y 128). Si se quiere llevar a cabo una operación con un número mayor, habrá que dividir éste en dos números de 8 bits, y en consecuencia el microprocesador empleará el doble de tiempo que al realizar la operación con un único número de 8 bits. Los procesadores actuales trabajan con grupos de hasta 64 bits.

Finalmente, el **número de operaciones con números decimales por segundo**<sup>6</sup> también es un factor importante. Hoy en día, los microprocesadores llegan a rebasar los gigaflops (miles de millones de operaciones por segundo).

<sup>(6)</sup>En inglés, *floating point operations per second* (flops).

La siguiente tabla muestra algunos de los microprocesadores de la casa Intel usados en los ordenadores personales, con el año de introducción en el mercado. También se muestra el número de transistores que lo forman, para dar idea de cómo se ha incrementado la complejidad en el diseño de los microprocesadores.

| Año  | Nombre      | Bits de trabajo | Megahercio <sup>7</sup> | Transistores |
|------|-------------|-----------------|-------------------------|--------------|
| 1978 | 8086        | 16              | 8                       | 29.000       |
| 1982 | 80286       | 16              | 12,5                    | 134.000      |
| 1988 | 80386       | 32              | 25                      | 275.000      |
| 1989 | 80486       | 32              | 50                      | 1.200.000    |
| 1993 | Pentium     | 32              | 66                      | 3.100.000    |
| 1997 | Pentium II  | 32              | 300                     | 7.500.000    |
| 1999 | Pentium III | 32              | 800                     | 28.000.000   |
| 2003 | Pentium IV  | 32              | 2400                    | 42.000.000   |

<sup>(7)</sup>De cada procesador ha habido versiones con diferentes velocidades; mostramos un valor medio.

A partir del 2003, al tener dificultades tecnológicas para aumentar la velocidad de reloj, se pensó en diseñar microprocesadores que incluyen, en realidad, más de un microprocesador.

### 3.1.2. La memoria

Si el microprocesador es un elemento importante a tener en cuenta en los ordenadores, también lo es la memoria. Acto seguido, se detallan los tipos de memoria que se pueden encontrar en un ordenador.

La **memoria ROM**<sup>8</sup> es una memoria que no se borra. Este tipo de memorias sirven, por ejemplo, para almacenar procedimientos y datos fundamentales para el funcionamiento de determinados componentes del ordenador o aparatos digitales. Notad que en este tipo de memoria no se puede modificar su contenido, únicamente se puede consultar.

<sup>(8)</sup>En inglés, *Read-Only Memory*, es decir, memoria sólo de lectura.

La **memoria RAM**<sup>9</sup> es la más importante. Se trata del conjunto de chips que sirven para que el ordenador almacene los procesos que se ejecutan (es decir, las instrucciones que conforman los programas que se están ejecutando) y los datos que utilizan estos programas. Cuanta más memoria RAM haya y más rápido sea el acceso a ésta, mejor podrá ser el rendimiento del ordenador.

<sup>(9)</sup>Del inglés, *Random Access Memory*, memoria de acceso aleatorio.

Esta memoria RAM se vacía cuando se apaga el ordenador. Por lo tanto, es necesario disponer de memorias capaces de retener la información que contienen. El **disco duro** es el dispositivo de almacenamiento de memoria en el que se almacena el sistema operativo, el software que se utiliza, y los documentos con los que trabajaremos. Cuando se inicia el ordenador, el sistema operativo se carga desde el disco duro a la memoria RAM. Sin embargo, algunos equipos utilizan memorias tipo ROM para guardar este software (por ejemplo, los teléfonos móviles o los ordenadores ligeros).

#### Memoria principal y memoria secundaria

La memoria principal del ordenador está normalmente formada por módulos de memoria RAM. Los discos duros forman la denominada memoria secundaria.

La velocidad del disco duro también es un factor importante para evaluar el rendimiento de un sistema informático: cuanto más rápido se pueda ir a leer o escribir la información, mejor.

En general, la memoria RAM tiene mucha menos capacidad que el propio disco duro. Por ejemplo, el disco duro ofrece capacidades un centenar de veces superior a la memoria RAM. Uno podría pensar por qué no utilizar discos duros en vez de memorias RAM si aquéllos tienen más capacidad. La razón es el tiempo de acceso. La tecnología utilizada en los discos duros provoca que su velocidad de acceso sea mucho menor que la velocidad de la memoria RAM.

Una solución utilizada para aumentar la memoria principal del ordenador es emplear parte del espacio del disco duro como **memoria virtual** y guardar aquella información que se está utilizando pero no cabe en la RAM.

#### Ejemplo de uso de la memoria virtual

Si tenemos un gestor de correo abierto pero estamos retocando una fotografía, lo más probable es que el programa de retoque y la imagen estén en la RAM, mientras que el programa de correo está "descansando" en el disco duro. Una vez activamos el gestor de correo, éste pasará automáticamente a la memoria RAM (y, en consecuencia, quizás la imagen que estamos retocando se guardará en la memoria virtual del disco duro).

Finalmente, la **memoria caché** permite tener más próxima al microprocesador aquella información que se utiliza más a menudo. Se encuentra ubicada dentro de la propia CPU, al lado del procesador, de manera que acceder a información de la memoria caché lleva menos tiempo que acceder a la información que se encuentra en la RAM (y, evidentemente, ¡menos tiempo que para acceder a la memoria virtual del disco duro!).

### Ejemplo de uso de la memoria caché

Siguiendo con el ejemplo anterior, la porción de la imagen que estamos retocando podría estar en la memoria caché, mientras que el resto de la imagen se encuentra en la RAM.

También existen otros dispositivos de almacenamiento como los DVD o las memorias flash.

### 3.1.3. La placa base

El microprocesador y la memoria se encuentran acoplados a la placa base<sup>10</sup>. Esta placa contiene una ubicación para cada uno de los elementos básicos del sistema informático: bancos para ubicar chips de memoria RAM, conexiones para el disco duro, puertos para los dispositivos de entrada y salida, etc.

### 3.1.4. Dispositivos de entrada y salida de información

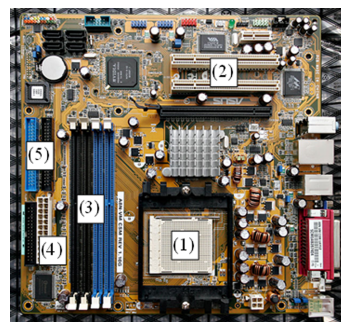
Nombramos ahora los dispositivos más importantes de entrada y salida de información. Gracias a ellos, podemos introducir información en el sistema informático (en el caso de los dispositivos de entrada) o ver la información (en el caso de los de salida).

El **teclado** es un dispositivo de entrada que permite introducir texto, por ejemplo, en un documento, en el navegador, etc. Dispone de teclas distribuidas en áreas (área numérica, alfanumérica, teclas de navegación, etc.), y algunos teclados más modernos disponen de otros botones y teclas para hacer diferentes tareas (subir o bajar el volumen del sonido, apagar el ordenador, etc.). Aunque lo más habitual es disponer de un teclado que se conecta a la unidad central con un cable, podemos optar por un teclado sin hilos. La idea es que lo que va conectado al ordenador no es el propio teclado, sino un receptor.

El **ratón** es un dispositivo de entrada que se inventó a finales de 1960 y fue olvidado en un rincón, hasta que la casa Apple lo popularizó en sus ordenadores. El ratón permite desplazar un puntero por la pantalla al ser movido por una superficie plana. Podemos activar acciones haciendo clic con sus botones o moviendo una ruedecita de desplazamiento que contiene. Podemos encontrar dos tipos de ratón, como en el caso del teclado: ratones con cable y ratones inalámbricos. Por otra parte, según la tecnología que utilizan para detectar el movimiento, tenemos los ratones mecánicos (de bola) y los ratones ópticos. La ventaja de éstos respecto a los primeros es que no se ensucia su mecanismo interno y siempre disfrutaremos de un movimiento preciso con ellos, mientras que con los ratones de bola a menudo deberemos sacar la bola y limpiar los ejes internos.

Un **escáner** es un dispositivo de entrada que permite digitalizar (pasar al ordenador) cualquier documento en papel, como una fotografía, un dibujo, una diapositiva, etc.

(10) También conocida como placa madre, según el nombre en inglés *motherboard*.



Esquema de una placa base habitual. 1) Espacio para el microprocesador. 2) Ranuras para placas de extensión para conectar dispositivos. 3) Espacio para chips de memoria RAM. 4) Conector para la fuente de alimentación eléctrica. 5) Conectores para el disco duro.



El último dispositivo de entrada que describimos es la webcam o **cámara web**, que va conectada al ordenador, con lo que su campo de alcance de toma de imágenes es bastante limitado. Se utiliza básicamente en comunicaciones de videoconferencia.

La **pantalla** o monitor es el dispositivo de salida más básico e importante del ordenador. Su dimensión se mide, como todos los televisores, con el número de pulgadas en diagonal. Así pues, encontramos monitores de 15", 17", etc. La pantalla es una parrilla que contiene miles de píxeles para representar imágenes y textos. El número de píxeles que pueden representar es un factor a tener en cuenta: cuantas más filas y columnas tenga, más elementos al mismo tiempo se podrán visualizar y más cómodo será trabajar con determinadas tareas.

La **impresora** es un dispositivo de salida que permite sacar en papel todo aquello que hacemos con el ordenador. Existen diferentes tecnologías de impresión, entre las que conviene destacar la de inyección de tinta (la más habitual en el mercado doméstico), la láser (para grandes volúmenes de impresión) y la matricial (ruidosa, pero utilizada en casos específicos).

#### Ved también

Entraremos más en detalle sobre cómo representar las imágenes en formato digital en el módulo "La World Wide Web".

#### Los píxeles

Una imagen digital se dibuja en la pantalla a base de píxeles, del inglés *picture element*, en español, elemento de imagen.

### 3.1.5. Dispositivos de almacenamiento de información

Finalmente, describiremos algunos de los dispositivos dedicados al almacenaje de la información.

El **disco duro**, generalmente ubicado en el interior del ordenador, permite el acceso rápido a la información que se almacena. Está formado por varios discos cerrados en una carcasa que los protege.

Los discos duros externos suelen utilizarse como extensión del disco duro, o bien para guardar los documentos que no nos caben, o bien como soporte para hacer copias de seguridad.

Las **unidades ópticas**, como los lectores/grabadores de CD-ROM y DVD, permiten la lectura y grabación de este tipo de discos. Los CD-ROM<sup>(11)</sup> tienen capacidades de grabación de unos 600 MB, mientras que con los DVD llegamos a los varios GB de espacio de almacenamiento. Inicialmente, estos dispositivos se han utilizado, aparte de como soporte de audio y vídeo digital, como soporte para las copias de seguridad de los documentos y programas. Sin embargo, la proliferación y el abaratamiento de los discos duros externos han provocado que los CD y los DVD se releguen a la multimedia y a la distribución de software.



Interior de un disco duro, en el que se muestra el brazo lector/escritor.

#### Los disquetes

Los **disquetes** son unos dispositivos muy utilizados en los años ochenta y noventa del siglo XX. Aunque algunos ordenadores aún incluyen de serie una disquetera para cubrir las funcionalidades de los disquetes, ésta ha quedado relegada con las memorias flash e incluso el envío de información en Internet.

<sup>(11)</sup>Del inglés *Compact Disc Read Only Memory*, memoria sólo de lectura en un disco compacto.

Finalmente, las **memorias flash** son un dispositivo utilizado para transportar información de un equipo a otro. Permiten el almacenamiento de una cantidad considerable de información (actualmente del orden de los GB), pero no se suelen utilizar como soporte de copia de seguridad por su baja fiabilidad a largo plazo y porque la cantidad de información que se puede llegar a almacenar en el disco duro es considerablemente más elevada.

### 3.2. El software

El software es un **conjunto de instrucciones** que indican al procesador lo que debe hacer. Estas instrucciones se utilizan para manejar los datos, codificados en binario como hemos visto, de manera conveniente para resolver problemas.

Es interesante hacer notar que el software también se encuentra codificado en binario y almacenado en memoria junto a los datos.

En realidad, no puede ser de otra manera, ya que como hemos explicado, los ordenadores sólo entienden conceptualmente de unos y ceros. Por lo tanto, al igual que en código ASCII se codifica un carácter en una serie de unos y ceros para almacenarlo en memoria, se hace lo propio con una instrucción que se debe ejecutar (por ejemplo, la instrucción sumar).

Si todo es binario y todo está almacenado en memoria, ¿cómo sabe un ordenador qué es una instrucción y qué es un dato? Se trata de una buena pregunta, porque en realidad no lo sabe. En un dispositivo de almacenamiento, por ejemplo un disco duro, sólo se puede identificar unos y ceros y no se puede distinguir cuáles corresponden a instrucciones y cuáles a datos. La clave está en el orden en el que se le pasan los bits al procesador.

El procesador se construye de manera que cada instrucción se representa con un conjunto de bits. Además, cada instrucción acepta un conjunto determinado de argumentos después de la instrucción que identifican los datos.

Imaginemos una situación en la que queramos sumar dos números con un procesador que tiene la siguiente instrucción:

**SUMAR DATO1, DATO2, RESULTADO**

Siguiendo el orden predeterminado, el procesador espera primero el código binario de la instrucción y, a continuación, la dirección de memoria en el que está almacenada la primera cifra seguida de la segunda. Finalmente, recibe la dirección en la que se guardará el resultado. De esta manera, el procesador, dependiendo del orden en el que llegan los bits, algunas veces los interpretará de una manera (instrucciones) y otras veces, de otra (datos).

Si, por la razón que fuera, no se le envían al procesador las cosas en el orden correcto, dará un comportamiento erróneo o un error. Nótese que cada procesador normalmente tiene su conjunto de instrucciones codificadas de una manera concreta. Por ello, el software creado para una arquitectura específica

(de las que el procesador es una parte importante) no funciona en otra arquitectura. Un software para la arquitectura i386 de la familia Intel (la de los IBM PC tradicionales) no funcionará en una arquitectura Motorola, usada en los antiguos Macintosh de Apple. La solución consiste en adaptar el software para que siga las pautas de cada arquitectura.

La característica más importante que diferencia el software del hardware es su **flexibilidad**. El hardware, y no sólo el de ordenador, es específico para cada tarea. Por ejemplo, la Pascalina que ya conocemos permitía hacer (algunos) cálculos, pero nada más, al igual que una máquina exprimidora de fruta sólo permite exprimir fruta. Se puede tratar de hacer otras cosas con la máquina exprimidora, como freír un huevo, pero el resultado no será satisfactorio.

El software es muy versátil, debido a que su tarea es manejar (tomar, transformar, guardar) datos binarios. En otras palabras, con un mismo ordenador y diferente software podemos resolver un gran conjunto de problemas. Y es que una vez que un problema se ha conseguido transformar en datos binarios, podemos resolver el problema aplicando una serie de instrucciones sobre ese conjunto de datos.

### **Los edificios inteligentes**

Los edificios inteligentes cuentan con ordenadores que toman datos del edificio (personas que entran y salen del edificio, temperatura, humedad, luminosidad, etc.) mediante sensores y otros dispositivos. Estos datos son convertidos en binarios, y posteriormente almacenados y tratados. Así, se puede conocer fácilmente el número de personas que han entrado y salido, por ejemplo, en las horas punta o ajustar la temperatura o la luminosidad que debe haber en cada espacio para ahorrar energía. Y esto es sólo el principio, ya que se nos podrían ocurrir muchas tareas más, incluso cuando los diseñadores originales del software ni las pudieran imaginar. Por ejemplo, se podría añadir funcionalidad para que el ordenador central, diseñado en la década de los ochenta, cuando todavía no existía la tecnología de la telefonía móvil, permitiera reservar salas de reuniones mediante el envío de un mensaje SMS.

Pero no todo son ventajas. La flexibilidad también conlleva que el software sea cada vez más complejo, ya que cada vez se quieren programas que realicen tareas más complicadas.

### **La complejidad del software**

En el caso de los edificios inteligentes, si al principio sólo se quería conocer el número de personas que entraban y salían de un edificio, más adelante se desarrolló un sistema para ajustar de manera inteligente la temperatura según la ocupación del edificio, para terminar queriendo un sistema que acepte reservas de salas mediante mensajes de telefonía móvil.

Todo esto provoca que el software se vuelva cada vez más complejo: el número de factores que hay que tener en cuenta crece, así como las situaciones y las reglas... Con tantos factores que considerar, el hecho de crear software se ha convertido en una de las actividades que requieren mayor ingenio humano. Si no fuera así, probablemente lo crearían máquinas de manera automática.

### 3.2.1. Algoritmos

¿Cómo se crea el software? Ya hemos indicado que el software está pensado principalmente para resolver problemas. La manera en la que podemos resolver estos problemas es mediante el uso de **algoritmos**.

Un **algoritmo** es un conjunto finito de pasos que sirven para resolver un problema o llevar a cabo una tarea.

Un algoritmo es como una especie de receta, como las de la cocina, pero en este caso informáticas. Así, imaginemos que quisiéramos cocinar unas sabrosas torrijas para degustarlas a la hora de la merienda. Al ir a un libro de recetas de cocina cualquiera, nos encontraríamos con una receta como la que sigue:

- 1) Mezclar 2 huevos levemente batidos con una cucharadita de vainilla, media de canela y una taza de leche.
- 2) Mojar 6 rebanadas de pan en la mezcla.
- 3) Freír en un poco de mantequilla hasta que se doren.
- 4) Servir el pan con un poco de sirope de caramelo.



En la imagen, se pueden ver unas torrijas, listas para ser degustadas, también conocidas como tostadas francesas. Fuente: Wikipedia. Licencia: Creative Commons Atribución 2.0.

Los algoritmos comparten con las recetas de cocina el hecho de que están escritas con un **lenguaje natural**. Esto es una ventaja, porque una vez que se ha aprendido a leer y se tienen los conocimientos básicos de los ingredientes que se suelen encontrar en la cocina tradicional, se puede entender una receta sin mayores problemas. Pero, a su vez, el uso del lenguaje natural resulta problemático. Es un problema, porque el lenguaje natural es **ambiguo**. Así, cuando se pide batir levemente dos huevos, ¿a qué se refiere con "levemente"? ¿A qué velocidad significa "levemente"? También están esos detalles relacionados con las cantidades que todos conocemos de nuestras experiencias culinarias, como el "agréguese una cucharada de sal...". ¿Se trata de una cucharada sopera o de una de café?

En general, los humanos tenemos cierta capacidad (aunque limitada) para disminuir estos problemas asociados a la ambigüedad del lenguaje. Y esta capacidad se ve aumentada en el caso de la cocina, porque se suele degustar el producto mientras se cocina o justo al terminar. Pero las máquinas no tienen esta capacidad. Las máquinas no entienden bien *levemente* o *un poco*, salvo que se sea más concreto. Y ahí es donde, en el software, entran los programas. Los programas deben verse como una **descripción formal de un algoritmo**. Para

tal fin, en vez de valernos del lenguaje natural, utilizaremos una serie de lenguajes pseudomatemáticos llamados **lenguajes de programación**, que explicaremos más adelante.

### 3.2.2. Diseño del software

Cuando se crea un software, el primer paso suele corresponder a la descripción del algoritmo en lenguaje natural. Generalmente, las personas que quieren utilizar el software no saben de lenguajes de programación. Por otro lado, los ingenieros de software saben programar, pero muchas veces no entienden el problema al que se enfrentan.

Para aclarar este punto, imaginemos a un chef que quiere una máquina de hacer torrijas. Es evidente que el chef sabrá hacer torrijas, pero es improbable que tenga conocimientos avanzados en programación de software. Para tener un programa que haga torrijas llamaría a alguien con conocimientos de programación. Sin embargo, es muy probable que esta persona experta en programación no haya hecho una torrija en su vida.

La manera que el chef y el ingeniero de software tienen para entenderse será mediante un algoritmo como el que hemos visto con anterioridad, especificado mediante lenguaje natural. El chef y el ingeniero de software quedarán para indicar las especificaciones y requisitos del programa. El nombre específico que recibe el ingeniero de software en este caso suele ser el de **analista de software**, ya que su tarea es realizar un análisis de los requisitos y de cómo afrontarlos para codificarlos convenientemente en el programa que quiere el chef.

En general, la siguiente etapa suele consistir en tomar los requisitos del análisis del software (plasmados en un documento) y crear un documento de diseño del programa. Básicamente, esto consiste en tomar los algoritmos de la etapa de análisis y organizarlos de manera que el software sea sencillo de realizar, sin perder a su vez flexibilidad para que en el futuro se puedan realizar mejoras. Y es que cuanto mejor diseñado, menos errores tendrá el programa y más fácil será modificarlo en el futuro, incluyendo nuevas funcionalidades. Como consecuencia de este trabajo, se creará otro documento en el que los algoritmos estén organizados convenientemente.

Será tarea de un programador transformar estos algoritmos en un programa de software, utilizando para ello un lenguaje de programación. Si la tarea de elaborar un algoritmo no es sencilla, tampoco lo es la transformación a un lenguaje de programación. Se debe eliminar la ambigüedad, ya que los programas deben ser precisos. En el caso de que no se haga correctamente, el software contendrá errores. Imaginemos las trágicas consecuencias de haber indicado mal el tamaño de las cucharadas de azúcar para nuestras torrijas. ¡Sería un desastre!

## Errores del software

Errores informáticos se pueden encontrar en todos los sitios, pero los aeroespaciales son de los más conocidos. Por ejemplo, en 1972, la sonda Mariner 1 se desvió del curso establecido y tuvo que ser destruida debido a que una fórmula escrita a mano fue trasladada erróneamente al lenguaje de programación. Otro caso, éste en 1999, tuvo como protagonista a la Mars Climate, que se estrelló contra el suelo de Marte debido a un error en la conversión de millas a kilómetros.

Como vemos, es común que el software tenga errores, y para aplicaciones algo complejas –casi todas las que utilizamos a día de hoy– es imposible asegurar que un software esté libre de ellos. Por este motivo, la creación del programa también consiste en **probarlo** –comprobar si tiene errores– y **validarlo** –asegurar que al final del proceso se obtiene lo que queríamos al principio y no otra cosa diferente–. Antes de que un software se publique definitivamente, suele pasar por un período en el que se comprueba que hace lo que debe hacer. En este sentido, la programación se parece mucho al quehacer habitual de la cocina, ya que toda receta se prueba una vez elaborada, y generalmente las recetas más innovadoras se suelen probar en círculos íntimos antes de ser presentadas en grandes ocasiones.

Y aun así, es común que con el paso del tiempo aparezcan errores que no se habían detectado o se quiera que el software tenga nuevas funcionalidades que no se habían incluido anteriormente. Es por ello que la última etapa de la vida de un programa es la del **mantenimiento**. Durante toda esta fase, los ingenieros de software gestionan el programa eliminando errores y añadiendo nuevas funcionalidades. Se trata de la etapa más larga, ya que mientras las anteriores se suelen llevar a cabo en cuestión de meses o de unos pocos años, un software puede estar en fase de mantenimiento durante años, en algunos casos, incluso décadas.

## Ejemplo de software longevo

El caso más conocido de software longevo se puede encontrar en el sector de la banca, en la que existe una gran cantidad de software de la década de los setenta. Un ejemplo de funcionalidad que se debió añadir a este software es la conversión de las antiguas pesetas al euro, la nueva moneda única europea. Para tal fin, los programadores encargados del mantenimiento tuvieron que modificar los programas, y probar otra vez las nuevas versiones del programa para constatar que no tuvieran errores.

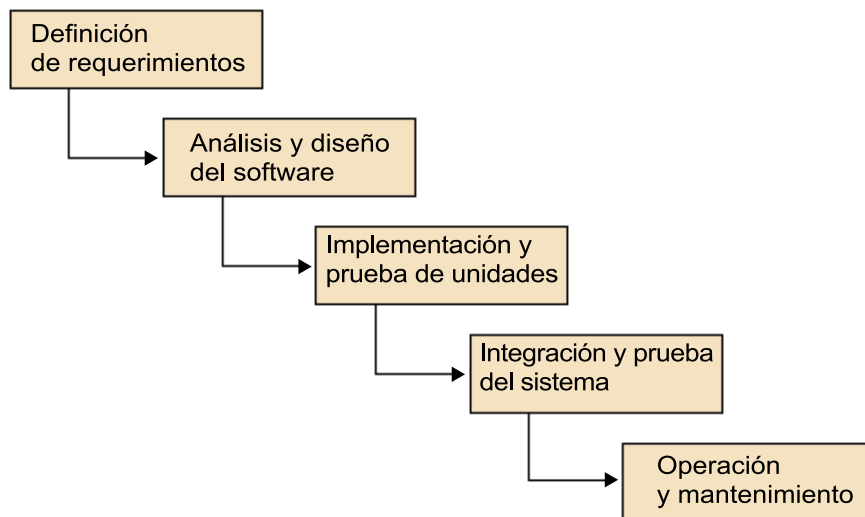
El modelo de desarrollo del software recibe el nombre de **modelo de desarrollo en cascada**, porque las fases se realizan una detrás de otra sin solapamientos. Éste es el modelo que se utiliza de manera tradicional en la gran industria del software.

- La "definición de requerimientos" es lo que en nuestro caso ha hecho el analista de software con el chef en un algoritmo.
- Tomando los requerimientos como punto de partida, un diseñador de software crea un diseño sobre papel, como un arquitecto dibuja unos planos. Generalmente, es una buena práctica dividir el problema general en sub-

problemas, más manejables y abordables. Estos subproblemas reciben el nombre de unidades.

- En la tercera etapa, se tomará este diseño y se implementará mediante el uso de un lenguaje de programación. La división en unidades permite probar cada unidad de manera aislada y comprobar que cada una de ellas hace lo que queremos.
- En la cuarta etapa, se toman todas las unidades y se integran en un único programa. Es ahora cuando hay que probar el sistema completo para comprobar si es correcto.
- Después de probarlo, se entrega al usuario (entra en operación) y se corrigen los errores que éste pueda notificar.

#### **Modelo cascada puro o secuencial para el ciclo de vida del software**



Fuente: Wikipedia. Licencia: Creative Commons Atribución CompartirIgual.

El modelo en cascada es sencillo de llevar a cabo y de entender, pero tiene también sus aspectos negativos. El primero es que desde la primera fase hasta que se obtiene el programa, puede pasar mucho tiempo. Y sólo entonces es cuando le podemos entregar al cliente el programa definitivo. Si se da la circunstancia de que en cualquier eslabón de la cadena la información no ha fluído de manera conveniente o que se ha desviado del propósito inicial, se podría tener un software que funciona pero que no hace lo que el cliente quería. Para evitar esta situación, existen modelos de desarrollo más flexibles que obtienen resultados intermedios con anterioridad, de manera que el cliente pueda ver la evolución y dictaminar si ésta es de su agrado o no.

### 3.2.3. Evolución de los lenguajes de programación

En un principio, el trabajo de implementación de los programadores era muy arduo, porque las máquinas sólo entienden unos y ceros. Así, su trabajo consistía en tomar el algoritmo deseado y transformarlo en **código máquina**, que es el nombre que recibe la secuencia de unos y ceros de los programas.

#### Ejemplo de código máquina

El código máquina de dos números que se suman podría ser el siguiente: 1100 0100 0010  
1001 0100 0100.

Trabajar de esta manera con unos y ceros directamente es, como os podéis imaginar, una tarea tediosa. Es difícil ver qué es lo que se está haciendo, y es difícil identificar si hay algún error. En programas escritos directamente en código máquina, la tarea de encontrar un error (conocida como **depuración**) se convierte en una tarea titánica. Al hecho de trabajar con unos y ceros se lo conoce como trabajar al nivel de máquina; ya que es como si estuviéramos conversando a su misma altura.

Para no tener que trabajar al nivel de máquina, se inventó un **nivel de abstracción** superior. Este nivel de abstracción permite que la tarea de programación sea más sencilla: en vez de unos y ceros se trabaja con representaciones más abstractas de las instrucciones. Así, surgió el **lenguaje ensamblador**.

#### Ejemplo de lenguaje ensamblador

La instrucción para el código máquina de dos números que se suman (1100 0100 0010 1001 0100 0100) podría ser la siguiente: *add \$1 \$2*.

Habréis podido ver que se trata de un pseudolenguaje mitad matemático, mitad inglés, que parece indicar que se trata de una operación de suma (*add*) de dos números. En vez de indicar los dos números directamente, se indican las posiciones en las que están almacenados esos dos números en memoria (en 1 y 2). Cabe notar que no se indica la dirección en la que se debe guardar el resultado. Algunos lenguajes en ensamblador guardan el resultado de la operación en el primer operando. También habrá otras instrucciones, como una operación para introducir en una posición de memoria un número *store \$1, valor*. Mediante esta instrucción, por ejemplo, se podría introducir en la dirección \$1, el valor -4.

Aun así, el lenguaje ensamblador tiene una filosofía similar al código máquina; simplemente es más legible. Los informáticos dirían que se trata de un lenguaje muy cercano a la máquina, ya que la transformación en binario es inmediata. Esto tiene sus puntos positivos, pero también sus aspectos negativos. En el lado positivo cabe mencionar el hecho de que es muy eficiente, ya que se está usando directamente el conjunto de instrucciones que utiliza el procesador. Al trabajar en un nivel de abstracción tan cercano a la máquina, se pueden optimizar las operaciones, obviar las redundancias y aprovechar el conocimiento de los pormenores del procesador. Sin embargo, aunque sea mejor que trabajar al nivel de unos y ceros, no deja de ser una manera muy tediosa de indicar secuencias de instrucciones; sigue siendo muy fácil cometer errores. Además, diferentes procesadores –por ejemplo, de diferentes fabricantes– pueden tener un conjunto de instrucciones distintas, por lo que el programa en ensamblador puede que sólo sirva para un procesador específico.



## Programa en ensamblador

Para aclarar un poco más los aspectos relativos al lenguaje ensamblador y sin entrar en detalles, en la siguiente figura se muestra una parte de un programa en ensamblador, con su correspondiente código máquina. El código máquina se representa en hexadecimal para simplificar el ejemplo. Podéis observar que las primeras cinco líneas corresponden a instrucciones en lenguaje ensamblador, pero de la séptima a la undécima los unos y ceros no se corresponden con instrucciones, sino con datos. Se trata de un texto codificado en ASCII.

Lenguaje máquina y lenguaje ensamblador para la arquitectura Intel 8088

| Código<br>máquina                                         | Código<br>ensamblador |
|-----------------------------------------------------------|-----------------------|
| -u 100 1a                                                 |                       |
| OCFD:0100 8A0B01                                          | MOV DX,010B           |
| OCFD:0103 8409                                            | MOV AH,09             |
| OCFD:0105 CD21                                            | INT 21                |
| OCFD:0107 8400                                            | MOV AH,00             |
| OCFD:0109 CD21                                            | INT 21                |
| -d 10b 13f                                                |                       |
| OCFD:0100 20 65 73 74 65 20 65 73-20 75 48 6F 6C 61 2C    | Hola,                 |
| OCFD:0110 72 61 60 61 20 68 65 63-68 6F 20 65 72 6F 67    | este es un prog       |
| OCFD:0120 72 61 60 61 20 68 65 63-68 6F 20 65 72 6F 67    | rama hecho en as      |
| OCFD:0130 73 65 6D 62 6C 65 72 20-70 61 72 61 20 6C 61 20 | sembler para la       |
| OCFD:0140 57 69 68 69 70 65 64 69-61 24                   | Wikipedia\$           |
| Direcciones<br>de memoria                                 | Código<br>máquina     |

Fuente: Wikipedia. Licencia: GFDL y Creative Commons Attribution ShareAlike 3.0.

Debido a la laboriosidad del código ensamblador, empezaron a surgir con el paso del tiempo **lenguajes de programación** de mayor nivel de abstracción, es decir, más cercanos al humano (al programador). Notad que cuanto más se alejan de la máquina (de los unos y ceros) y más se acercan al lenguaje natural, mayor es el nivel de abstracción. Por un lado, se gana en facilidad de implementar los algoritmos; pero, por otro, se pierde la eficiencia que se ganaba al ajustar los algoritmos a la máquina. Hoy en día, casi todas las aplicaciones, salvo cuestiones muy concretas que requieran de mucha eficiencia, como las tareas del sistema operativo, se realizan en lenguajes de alto nivel.

### Ejemplos de lenguaje de programación

Existen muchos lenguajes de programación; los más populares a día de hoy son Java, C, C++, COBOL o Pascal, pero hay otros miles más.

Los lenguajes de programación permiten al programador centrarse en la implementación del algoritmo y olvidarse de las particularidades de la máquina.

Mediante estos lenguajes se crean programas menos complejos de leer y de escribir, y a su vez, menos propensos a errores, lo que permite tener aplicaciones más potentes –en el sentido de que pueden realizar tareas más complejas–. Sin embargo, son programas que no están optimizados tal y como podríamos hacer si se hubiera programado en lenguaje ensamblador. Pero la optimización no suele ser una prioridad en la mayoría de los casos para los programadores. Esto es debido a que en los últimos treinta años la potencia de los ordenadores (la velocidad de procesamiento, la capacidad de almacenamiento, etc.) se ha duplicado aproximadamente cada dieciocho meses. Eso significa que no tiene mucho sentido optimizar un programa utilizando el tedioso código ensamblador teniendo en cuenta que el coste de comprar una máquina más potente es mucho menor que el de tener a programadores optimizando los programas.

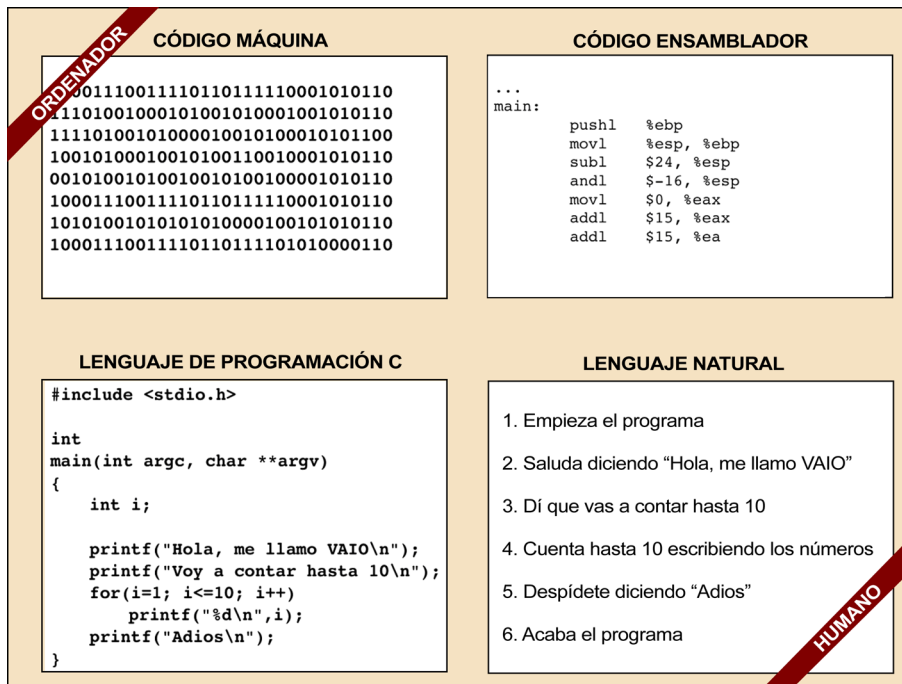
Para mostrar la conveniencia de los lenguajes de programación, se muestra a continuación un programa ejemplo, tomado de la Wikipedia, e implementado en el lenguaje de programación Pascal. Este programa realiza una suma entre dos números enteros dados por el usuario. Además de instrucciones y datos,

el programa incluye comentarios (se pueden identificar fácilmente, porque en Pascal los comentarios se encuentran entre llaves). Estos comentarios no son ni instrucciones ni datos; simplemente ayudan al programador actual (y a los futuros) a entender qué es lo que hace el programa.

```
program suma;
var x,s,r:integer; {declaración de las variables}
begin {comienzo del programa principal}
    writeln('Introduzca 2 números enteros'); {imprime el texto}
    readln(x,s); {lee 2 números y los coloca en las variables x y s}
    r:= x + s; {suma los 2 números y coloca el resultado en r}
    writeln('La suma es ',r); {imprime el resultado}
    readln;
end. {termina el programa principal}
```

En un lenguaje de programación de alto nivel, el texto que el programador genera se conoce como **código fuente**. Para poder estudiar un programa es indispensable contar con el código fuente. Esto es debido a que con la complejidad de los programas actuales, el código binario tiene una extensión tal que haría falta muchísimo tiempo y esfuerzo para descifrarlo. Sin embargo, si se ha utilizado un lenguaje de alto nivel, esto es al menos más sencillo.

En la figura siguiente exponemos la evolución que han sufrido los lenguajes de programación al crearse varios niveles de abstracción. Partiendo de nuestro algoritmo hasta llegar al código máquina que entienden los ordenadores, podemos ver los niveles intermedios (código fuente, en este caso en el lenguaje de programación C, y código ensamblador). Al proceso de transformar el algoritmo en código fuente se lo conoce como **implementación** y, como hemos comentado, lo realiza un humano. La transformación de código fuente en código máquina, sin embargo, es un proceso que se puede automatizar, incluso obviando el paso intermedio de transformar el código fuente en código ensamblador. Así, esta tarea es responsabilidad de un programa especial conocido como **compilador** y recibe el nombre de **compilación**. Cada lenguaje de programación tiene un compilador específico que transforma las instrucciones de la sintaxis del lenguaje de programación en el código máquina final.



Niveles de abstracción. (c) Enrique Soriano. Licencia: GFDL y Creative Commons Attribution ShareAlike 3.0.

### 3.3. Tipos de software

Llegados a este punto, tenemos unas nociones básicas suficientes de cómo desarrollar un programa nuevo. Así, si quisiéramos tener un juego propio de buscaminas, empezaríamos por aprender un lenguaje de programación de alto nivel, estudiaríamos el problema para obtener los algoritmos del juego y empezaríamos con la implementación.

Sin embargo, hay una serie de detalles que no se han abordado y que son de gran importancia, como la interacción con el usuario. Volviendo al ejemplo del buscaminas, además de los algoritmos del juego en sí, habría que incluir algoritmos para capturar las decisiones del usuario, en particular si éste ha pulsado ciertas teclas, si ha movido el ratón, etc.

Esto, en realidad, no debería ser mayor problema, ya que no dejan de ser más algoritmos que se pueden implementar con el mismo lenguaje de programación recién aprendido. Imaginemos que quisiéramos que nuestro buscaminas funcionara para muchos tipos de ratón diferentes, idealmente para todos. ¡Deberíamos implementar las instrucciones para todos y cada uno de los ratones existentes! O lo que es peor, mantener y modificar el juego para todos los ratones habidos y por haber.

Todo esto no tiene mucho sentido, sobre todo porque la idea es realizar un buen juego de buscaminas y no andar liados con el hardware de los usuarios. Es por esta razón, y por otras que veremos a continuación, que el software no se hace encargándose de todo, desde la funcionalidad que brinda al usuario hasta la interacción con los dispositivos conectados al ordenador. También

existen niveles de abstracción a la hora de crear funcionalidad y por ello el software se divide en dos grandes familias: el software de sistemas y el software para usuario final.

### 3.3.1. El software de sistemas: el sistema operativo

El **software de sistemas** tiene como objetivo interactuar con el hardware y proveer servicios y funcionalidad a otras aplicaciones. De esta manera, se trata de un programa que está más centrado en el ordenador que en el usuario. Aunque existen muchos software de sistemas, como los antivirus o los servidores, el software de sistemas más conocido es el sistema operativo.

Los **sistemas operativos** son programas que crean un nivel de abstracción superior sobre el hardware.

Dicho en otras palabras: el sistema operativo se encarga de comunicar los programas para usuarios y otro software de sistema con el hardware del ordenador. Siguiendo con el ejemplo del juego del buscaminas del subapartado anterior, el sistema operativo se encargará de la interacción con todos los posibles ratones que existan o puedan existir. Los programadores de aplicaciones para usuario final sólo deben tener en cuenta que se interactúa mediante un ratón genérico (y abstracto). Los programadores conocerán la posición del puntero y si se han pulsado los botones, independientemente del ratón usado.

Además, el sistema operativo es el encargado de gestionar un gran número de otras tareas y elementos, como veremos a continuación. Por esta razón, los sistemas operativos se han convertido desde hace un par de décadas en un componente esencial de todo ordenador. Entre sus responsabilidades podemos encontrar:

- La **comunicación con los periféricos**. Es el caso que hemos visto hasta ahora; el sistema operativo tiene como objetivo que los programas de usuario puedan trabajar de manera independiente a las especificidades del hardware que se tenga. Por ello, cuando se compra un nuevo dispositivo, se deben instalar los controladores<sup>12</sup>.
- La **gestión de la memoria**. Es tarea del sistema operativo gestionar las diferentes memorias que podemos encontrar en un ordenador, desde la memoria principal (RAM) hasta la secundaria (el disco duro), pasando por la memoria virtual y la memoria caché que hemos visto con anterioridad. La gestión de la memoria es una tarea muy importante si recordamos que la memoria principal es muy rápida, pero muy cara y volátil, mientras que la memoria secundaria es lenta, pero barata y persistente. Es precisamente por la velocidad, que nos interesa que tanto las aplicaciones que se ejecutan como los datos se encuentren en la memoria principal. Sin embargo,

#### El caso de las impresoras

Los creadores de los procesadores de textos como Microsoft Office u OpenOffice.org Writer no se preocupan de la impresora instalada y de cómo se envían los trabajos. Estas aplicaciones simplemente indican al sistema operativo el fichero que se debe imprimir, y será tarea del sistema operativo que eso suceda.

<sup>(12)</sup> En inglés, *drivers*.

#### Los controladores

Los **controladores** son los programas que permiten la comunicación del sistema operativo con los dispositivos.

debido a su coste, generalmente no se tiene memoria principal suficiente para almacenar todo en una memoria principal y es necesario que el sistema operativo use la memoria secundaria.

- La **conurrencia de procesos**. Aunque el procesador sólo puede ejecutar una instrucción cada vez, se puede conseguir concurrencia virtual mediante el sistema operativo. En otras palabras, el sistema operativo puede conseguir que el usuario tenga la impresión de que se están ejecutando varios programas a la vez. Imaginemos que se está escribiendo una carta con el procesador de textos mientras se descarga un documento con el navegador web y se calcula mediante un explorador de archivos cuánto espacio libre queda en el disco duro. En la pantalla, se ve la ventana del procesador de textos, por lo que aparentemente se podría pensar que ésta es la tarea que tiene entre manos el sistema operativo, dejando las otras para después. Y, sin embargo, se tiene la impresión de que todas ocurren a la vez. En realidad, esta impresión es falsa: el procesador sólo puede ejecutar una instrucción a la vez, así que no puede realizar múltiples tareas de manera simultánea. Lo que sucede es que el procesador ejecuta las instrucciones a una velocidad tan elevada (lo hace a la velocidad de miles de millones de instrucciones por segundo) que nos parece que las realiza todas a la vez. Cuando las tareas del usuario dejan ocioso al procesador, éste realiza otras tareas. En nuestro ejemplo, mientras no se escribe nada en el procesador de textos, el procesador puede calcular el espacio libre en el disco duro, seguir descargando un documento con el navegador web y probablemente otras tareas de las que ni siquiera se es consciente, como ver cuánta batería nos queda en el portátil. Cabe notar que aunque actualmente la mayoría de los ordenadores tienen más de un procesador, se sigue utilizando la misma técnica explicada. El sistema operativo expulsa las tareas ociosas de los procesadores para ejecutar otras tareas pendientes. En este segundo caso hay concurrencia real entre procesos, ya que se ejecutan múltiples tareas al mismo tiempo en procesadores diferentes.
- La **monitorización y seguridad**. Los sistemas operativos modernos son multiusuarios. Esto introduce un problema de seguridad. Hay documentos personales en los que se debe restringir el acceso a otros usuarios del ordenador. El sistema operativo se encarga de esta función añadiendo metadatos a los ficheros, como el del usuario propietario. Por otro lado, para saber qué es lo que ha pasado en la ejecución de alguna tarea en especial, existen ficheros en los que se registran los acontecimientos para su posterior estudio. Es lo que se conoce como registros<sup>13</sup> y permite conocer qué ha sucedido y las tareas ejecutadas en un ordenador. Esto es útil, por ejemplo, para detectar si una máquina ha sido comprometida (alguien ha entrado en ella sin tener permiso de acceso).
- La **gestión de almacenamiento**. Los ordenadores contienen programas y datos. Ambos se guardan (almacenan) como ficheros en el disco duro. Es

<sup>(13)</sup>En inglés, *logs*.

tarea del sistema operativo hacer esto de manera ordenada (por ejemplo, en directorios o carpetas).

- La **gestión de comunicaciones**. Los ordenadores rara vez se usan hoy en día de manera aislada; generalmente se conectan a una red, como podría ser Internet. Para ello, se comunican mediante una serie de reglas establecidas, normalmente conocidas públicamente, que reciben el nombre de protocolos. El sistema operativo es el que se encarga del envío y la recepción de información utilizando protocolos de red y permitiendo que las aplicaciones (en realidad, los que las programan) no deban ocuparse de los detalles de la Red.

Debido a todas estas capacidades que tienen los sistemas operativos modernos, se puede entender rápidamente que son de gran provecho para crear nuevas aplicaciones de manera más rápida y sencilla.

Todo esto tiene un precio, que es el de la compatibilidad. Hemos comentado que los programadores realizan su trabajo creando código fuente para que posteriormente un programa conocido como compilador lo transforme en código máquina. Bien, pues esta etapa de compilación depende del sistema operativo, ya que el código máquina que "entiende" un sistema operativo no tiene por qué ser "comprendido" por otro. ¿Cómo se consigue esto? Principalmente mediante dos cuestiones:

- Por un lado, el lenguaje de programación usado cuenta con varios compiladores diferentes, uno para cada sistema operativo soportado. Por ello, a la hora de compilar se elige el compilador para nuestro sistema operativo, de manera que obtenemos el código máquina correspondiente.
- Por otro lado, los programadores han tenido suficiente cuidado para, además de utilizar un lenguaje multiplataforma (esto es, que se pueda compilar en diferentes sistemas operativos), no utilizar secuencias de programación o métodos que son exclusivos de un sistema operativo. Cabe notar que hay algunas funcionalidades o métodos que son específicos de un sistema operativo. La inclusión de estos métodos en el programa produciría errores en la compilación en otros sistemas operativos.

### **Ejemplos de dependencia del sistema operativo**

El programa PADRE para la realización de la declaración del impuesto sobre la renta en España, la *suite* ofimática libre OpenOffice.org o la del navegador Mozilla Firefox tienen diferentes versiones dependiendo del sistema operativo usado.

Con anterioridad, habíamos visto que el compilador depende de la arquitectura del ordenador. Así, la transformación de código fuente a código máquina sería diferente para una máquina Intel o para una Mac. Ahora, además, vemos que el sistema operativo también es una cuestión que tener en cuenta en esa transformación.

En definitiva, los compiladores deben ser específicos para una arquitectura y para un sistema operativo.

### **Ejemplo de compiladores**

Para que un programa creado en el lenguaje de programación Pascal pueda ejecutarse en un ordenador de la arquitectura Intel y con el sistema operativo GNU/Linux, se necesitará el compilador para Pascal para la arquitectura Intel y el sistema operativo GNU/Linux. El código máquina que generará este compilador no funcionará si sólo coincide la arquitectura (por ejemplo, en una máquina Intel con sistema operativo Microsoft Windows) o el sistema operativo (por ejemplo, en una máquina con arquitectura PowerPC con sistema operativo GNU/Linux).

### **3.3.2. Software para usuario final**

El **software para usuario final** tiene como finalidad resolver problemas del usuario, ya sea dando soporte, ya sea mejorando el trabajo que se quiere realizar. Su funcionalidad está más relacionada con la lógica de la tarea a realizar por el usuario que con el hardware del sistema o de los dispositivos con los que el usuario está interactuando, lo que, como hemos visto, es gestionado por el sistema operativo.

#### **Ejemplo de software para usuario final**

Tomemos una analogía del mundo urbano para comprender la diferencia entre el software de sistemas y el de usuario final. En las ciudades, tenemos el servicio de limpieza y mantenimiento que realiza tareas más cercanas a la infraestructura, mientras que las tiendas y los supermercados ofrecen a los ciudadanos servicios finales. Así pues, el software para usuario final es como las tiendas o supermercados, ya que ofrecen servicios y productos directamente al ciudadano, mientras que el software de sistemas se asemeja a los servicios de limpieza y mantenimiento, que se encargan de otras tareas con las que el ciudadano no suele interactuar directamente, pero sin las que sería difícil vivir (al menos cómodamente) en una ciudad.

Generalmente, los programas de usuario final realizan un objetivo específico, como puede ser la manipulación de imágenes, la grabación de CD-ROM, la navegación por la web, etc. La lista es larga y ha habido múltiples intentos de crear una taxonomía de programas para usuario final. A continuación, ofrecemos una posible clasificación de este tipo de software:

- **Software de infraestructura empresarial:** se trata de software como bases de datos para almacenar grandes volúmenes de datos, software de flujos del proceso de negocio que ayudan a la gestión del producto o servicio que ofrece la empresa, o software de sistemas de información geográfica que permite enlazar datos con cartografía, por ejemplo para conocer la localización exacta de cada uno de los camiones que componen la flota de la empresa.
- **Software de acceso a contenidos:** mediante este software, se puede acceder a información, ya sea textual o multimedia. Tal es el caso de navegadores web para navegar por Internet o de programas de reproducción

multimedia. El software de entretenimiento como juegos o salvapantallas también se suelen incluir en esta categoría.

- **Software educativo:** como tal se conoce al software con finalidad educativa. En esta categoría, encontramos software de gestión de aulas, software de aprendizaje o de entrenamiento o software de consulta (como las antiguas enciclopedias que vendían en CD-ROM, hoy en día más en desuso debido a Internet y la Wikipedia).
- **Software de simulación:** en esta categoría, se encuadrarían los conocidos juegos de simulación (como los simuladores aéreos o los de motor), pero también hay una gran gama de simuladores científicos que se utilizan en investigación.
- **Software de desarrollo multimedia:** aquí tenemos el software de gestión de imágenes, y de creación y edición de contenidos multimedia (vídeo, imágenes, sonido, etc.). Este campo está en gran apogeo en los últimos tiempos y está consiguiendo introducir grandes cambios en la industria cinematográfica.
- **Software de ingeniería:** ayudan en la creación ingenieril e incluyen el diseño –los famosos programas CAD (*computer aided design* o diseño asistido por ordenador), utilizados por ingenieros mecánicos, ingenieros aeronáuticos, ingenieros industriales y arquitectos–. También hay que incluir dentro de esta categoría el software de creación de programas software, entre los que se encuentran algunos de los que ya hemos hablado con anterioridad en este módulo, como el compilador.
- **Software de sistemas de información.**

#### Edutainment

El *edutainment* es un software de entretenimiento diseñado para educar.

#### Ved también

El software de sistemas de información se verá con más detalle en el apartado 5.

Un subconjunto del software para usuario final es el software integrado, como los paquetes ofimáticos. Cuentan con una infraestructura propia y una manera de interactuar con el usuario homogénea. Esto permite que, una vez que se conozca un programa de la *suite*, podamos utilizar los otros de manera más sencilla, ya que los menús, la manera de ser usado, etc., es parecida. Un ejemplo sencillo es el uso de iconos similares, mientras que entre las posibilidades más avanzadas podemos encontrar la posibilidad de integrar lo que hacemos con una aplicación en otra. Así, podríamos incluir una tabla de nuestra hoja de cálculo en nuestro documento en el procesador de textos.

Finalmente, existe un tipo de software para usuario final que también podemos hallar con bastante frecuencia: se trata de las aplicaciones a medida. Estos programas los solemos encontrar en el mundo empresarial, como por ejemplo en programas de facturación, o en la aplicación que manejan los empleados de banca, incluso el que se utiliza en un hospital. Este software es diferente



al anterior, ya que se suele diseñar y crear con la vista puesta en un usuario específico, y no para el gran público. Podríamos decir que son programas *ad hoc* creados específicamente para el usuario.

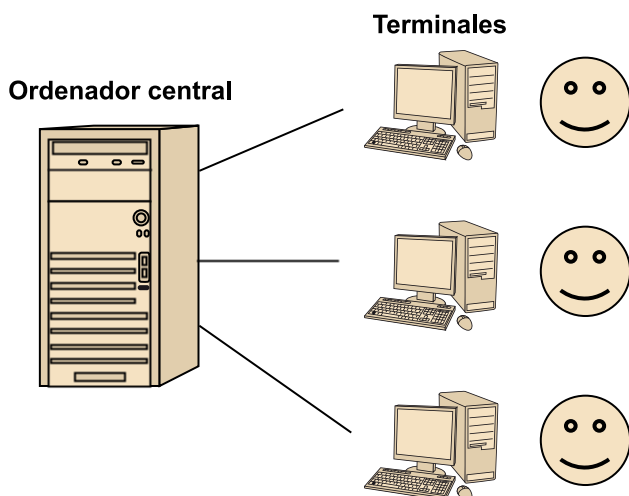
## 4. Tipo de ordenador

Los ordenadores personales no son extraños para nosotros. Cuando hablamos de ordenador personal nos viene a la mente la pantalla, el teclado, el ratón y una torre, grande o pequeña, que contiene el lector de DVD, las conexiones de dispositivos externos, los altavoces, etc. Sin embargo, durante la historia de la informática han ido variando los tipos de ordenador existentes. Algunos de ellos se consideran históricos y otros son muy populares en la actualidad. En este apartado, describiremos algunos de los sistemas informáticos más importantes.

### 4.1. Los ordenadores principales

Los primeros ordenadores tenían como objetivo la realización de cálculos sobre grandes volúmenes de datos. Durante décadas, ésta fue la principal actividad de los computadores. Llegó un momento en el que la tecnología permitió no sólo la miniaturización, sino también el hecho de poder ejecutar varias tareas a la vez mediante los sistemas operativos. Gracias a ello, un mismo microprocesador era capaz de ejecutar varios programas al mismo tiempo, simplemente repartiendo el tiempo de proceso entre los diferentes programas que se estaban ejecutando. De la misma manera, los sistemas multi-usuario permitieron que diferentes trabajadores accedieran a un mismo ordenador principal o *mainframe* para poder utilizarlo. Para trabajar con el ordenador había que utilizar un terminal que nos conectaría remotamente al equipo, como muestra la figura siguiente.

Un ordenador principal o *mainframe* con tres terminales conectados



Actualmente, este modelo continúa siendo válido, en cierta manera: muchos de los sistemas de información que utilizamos son grandes servidores accesibles en Internet. Para acceder, sólo es necesario utilizar el navegador web (que hace de terminal).

## 4.2. Los microcontroladores

Un microcontrolador es un chip que contiene todas las funcionalidades básicas descritas en la arquitectura Von Neumann, es decir, tiene una CPU, una memoria y una unidad capaz de gestionar la entrada y la salida de información. Estos chips pueden ejecutar programas. Los microcontroladores son ordenadores tan pequeños que es posible integrarlos en una gran variedad de aparatos: lavadoras, móviles, máquinas industriales, hornos microondas digitales, cámaras de vídeo, etc.

Hay microcontroladores genéricos que se pueden programar y se pueden utilizar en diferentes ámbitos. Por otra parte, existen microcontroladores específicamente diseñados para un determinado entorno.

## 4.3. Los superordenadores

Durante los años setenta y ochenta del siglo xx, varias compañías sacaron a la luz ordenadores muy potentes, si se comparaban con los ordenadores principales habituales en las grandes empresas, y evidentemente mucho más potentes que los microordenadores personales. Estos ordenadores, llamados supercomputadores, se utilizaban para aplicaciones específicas que requerían gran potencia de cálculo: desde simulaciones atómicas hasta participar contra humanos en competiciones de ajedrez. Sin embargo, hoy en día estos supercomputadores serían poco potentes si los comparáramos con los ordenadores portátiles actuales.

De esta manera, actualmente la idea superordenador se alcanza no con un único ordenador muy potente, sino con el trabajo colaborativo de centenares o miles de ordenadores "normales" conectados a redes de gran velocidad y rendimiento. Este concepto lo explicamos acto seguido.

## 4.4. Multicomputadores

Hemos visto anteriormente que se ha llegado a límites en cuanto a velocidad de reloj de los procesadores. Así pues, los nuevos ordenadores ya no aumentan su rendimiento a base de aumentar la velocidad de reloj, sino que se utilizan técnicas como la "multicomputación" para producir equipos más potentes.

Los **multicomputadores** propiamente dichos son los sistemas formados a partir de una **granja de ordenadores**, habitualmente con las mismas capacidades y características de rendimiento. De esta manera, las tareas se pueden distribuir entre los diferentes ordenadores que integran la granja y hacer el trabajo más

### Ved también

Recordad que hemos tratado los microordenadores personales en el apartado 1 de este módulo dedicado a la historia de los ordenadores.

### Deep Blue

Ha habido varios ordenadores diseñados específicamente para jugar al ajedrez. Uno de los más famosos es el Deep Blue, que en el año 1997 ganó al campeón de ajedrez Gari Kasparov.

rápido. Por ejemplo, en una productora de cine de animación por ordenador, una escena se podría dividir en pequeños trozos, y cada uno de ellos pasaría a ser tratado por un ordenador. Por otra parte, con sistemas multicomputador se pueden llegar a atender peticiones de miles de usuarios. Por ejemplo, los grandes servicios de búsqueda de Internet disponen de decenas de miles de ordenadores para atender los millones de peticiones que reciben cada minuto.

### Ejemplo de multicomputación

Los ordenadores personales con procesadores de varios núcleos son un ejemplo de multicomputación. Un microprocesador de doble núcleo contiene duplicados muchos de los elementos que forman un procesador, es decir, es como tener dos microprocesadores en uno. De esta manera, si se están realizando dos tareas al mismo tiempo (por ejemplo, trabajando con el campus virtual de la UOC y generando un pase de fotografías para ser registrado en un DVD), cada uno de los procesadores se dedicará a una de las tareas. En un sistema con un único procesador, al igual que los primeros ordenadores principales, se tardaría mucho más tiempo que con dos núcleos.

Finalmente, hay que hablar de los modelos de **computación grid**. El objetivo es también el procesamiento en un tiempo factible de tareas costosas por medio de la división del trabajo en tareas que realizarán los diferentes computadores que integran el grupo de trabajo. A diferencia del caso anterior, en la computación grid los equipos no son iguales, ni están conectados por una red de alta velocidad, ni se usan en exclusiva para la tarea: los equipos que se emplean suelen ser ordenadores personales de usuarios particulares que reciben las pequeñas tareas que realizar por medio de su conexión a Internet. Estas pequeñas tareas se van haciendo en períodos de inactividad del equipo.

#### Ejemplo de computación grid

Un ejemplo de computación grid es el *Search for Extra Terrestrial Intelligence (SETI)*, que se basa en el análisis de señales procedentes de radiotelescopios para descubrir si alguna de éstas puede haber sido generada por seres inteligentes.

## 4.5. Los ordenadores portátiles

La miniaturización de los equipos permitió la realización de equipos portátiles. Históricamente, los **ordenadores portátiles** o dispositivos electrónicos, como agendas digitales o **teléfonos móviles avanzados**, han sido equipos poco potentes si se los compara con los de sobremesa. Sin embargo, la tecnología ha permitido hacer ordenadores portátiles lo bastante potentes o lo bastante pequeños como para ser considerados ultraportátiles. Estos productos están siendo habituales, por lo que conviene citarlos en este material.

Un ordenador portátil es actualmente tan potente como un ordenador de sobremesa. De todas maneras, este tipo de ordenadores no es muy cómodo de llevar siempre encima, dado su peso. Los ordenadores **ultraportátiles** son equipos más cómodos de transportar. Incluso existe una variante que permite utilizarlos como si se tratara de una libreta para tomar nota (los llamados **tablet PC**): tienen una pantalla táctil para poder escribir directamente sobre la pantalla. Los ordenadores **netPC** son ordenadores aún más portátiles (normalmente tienen dimensiones más reducidas), a cambio de tener un rendimiento menor que los portátiles y los ultraportátiles. Sin embargo, son lo bastante potentes para permitir el acceso a Internet, la visualización de fotografías y vídeos y las tareas ofimáticas más habituales.

## 5. Sistemas de información

Un sistema de información, en términos generales, está constituido por una serie de elementos que se relacionan entre sí para apoyar en las actividades de una institución o empresa. Los sistemas de información se encuentran dentro del software para usuario final. Los elementos de los que está compuesto un sistema de información son:

- Las personas que interactúan con el sistema.
- Los datos.
- Las actividades para procesar la información y los datos de una organización.
- El componente informático, que incluye tanto hardware como software.

En un sentido más restringido del término, a veces muchos se refieren como sistema de información a la aplicación software que realiza las tareas de almacenamiento y algunas de manipulación y procesamiento de datos e información.

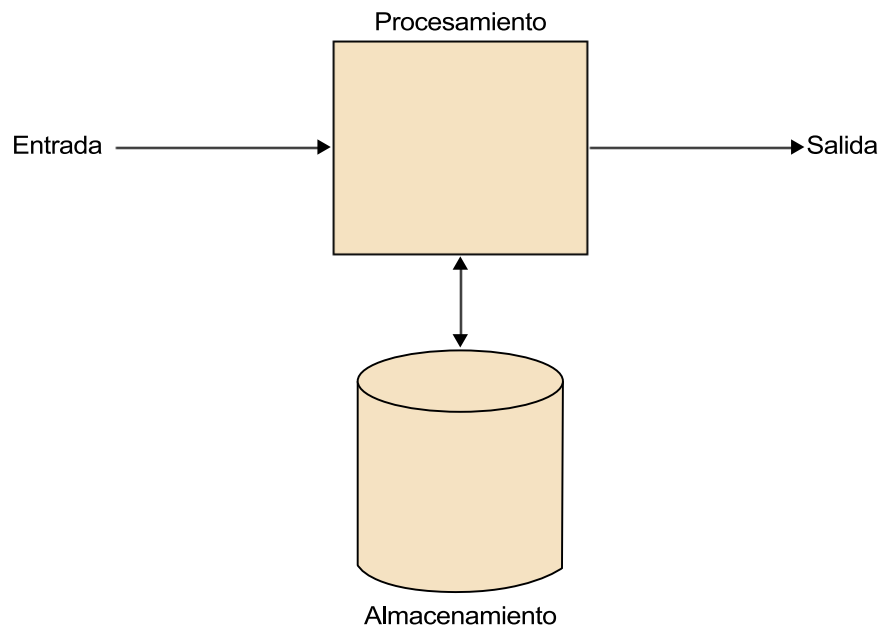
### Ejemplo de sistema de información

El ejemplo más paradigmático de sistema de información es el censo de una población. Herman Hollerith creó en 1890 un sistema de información para gestionar de manera automática una gran cantidad de datos que debían ser tratados. La diferencia principal de este primer sistema de información complejo con los actuales es que aquél era completamente mecánico, en vez de basarse en la electrónica. Pero el sistema de Hollerith sirvió de inspiración para la computación moderna, por lo que no deja de ser lógico que el primer gran sistema de información fuera instalado en la Oficina del Censo norteamericana en 1951, el UNIVAC I, uno de los primeros ordenadores.

En la actualidad, los sistemas de información se utilizan básicamente con tres objetivos:

- Automatizan procesos.
- Tratan información para la ayuda de decisiones.
- Buscan ventajas competitivas.

Independientemente de la tecnología que se utilice, los sistemas de información constan de cuatro actividades: entrada, procesamiento, almacenamiento y salida de información, como se puede observar en la figura siguiente.

**Actividades básicas de un sistema de información**

Ateniéndonos a la aplicación del sistema de información, tenemos un gran número de programas dentro de este ámbito, como:

- **Programas de gestión de tiempo y recursos.** Aquí es donde podemos encontrar los famosos ERP<sup>14</sup> utilizados para la gestión empresarial. Sin embargo, hay otros programas como los de contabilidad o los de gestión de tareas y agenda que también se pueden englobar en esta categoría.
- **Gestión de datos.** Se trata de software que permite gestionar cantidades de datos importantes, aunque no tan abundantes ni tan variadas como para tener una base de datos específica. Así, tenemos aplicaciones de gestión de contactos, las hojas de cálculo o las pequeñas bases de datos personales.
- **Documentación.** Los más conocidos dentro de este apartado son los procesadores de texto, aunque también existe software para publicaciones, software para la creación de diagramas y software para realizar presentaciones.
- **Software analítico.** Son programas que permiten el cálculo o la manipulación de datos numéricos para su análisis, con carácter estadístico, financiero o de otro tipo.
- **Software cooperativo.** Software utilizado para comunicarse con otras personas, tanto de manera asíncrona (correo electrónico, blogs o wikis) como síncrona (sistemas de mensajería instantánea o software de telefonía IP).

<sup>(14)</sup>Del inglés *Enterprise Resource Planning*, planificación de recursos de la empresa.

La importancia que ha cobrado Internet en los últimos tiempos ha provocado que los sistemas de información se estén adaptando a un entorno distribuido, en el que la información y los procesos no se encuentran localmente.

**Ved también**

En el módulo "La World Wide Web" trataremos la importancia de Internet.

## Resumen

En este módulo hemos visto las bases de funcionamiento de los sistemas informáticos. Hemos empezado con una visión histórica de los hitos más importantes de la historia de la computación y hemos comprobado que los sistemas informáticos tienen una historia relativamente reciente y que en los últimos años han entrado en casi todos los ámbitos de las actividades humanas.

También hemos estudiado cómo la informática trata la información. Hemos aprendido que los números y los caracteres se codifican en una serie de bits. De esta manera, la información se puede guardar utilizando diferentes tecnologías, a la vez que se permite su procesamiento mediante circuitos lógicos.

Una vez que hemos visto las bases del tratamiento de la información, hemos tratado las diferentes partes que forman el ordenador, a la vez que hemos hecho énfasis en aquellos aspectos del hardware que caracterizan el rendimiento y la potencia de la máquina. Finalmente, hemos clasificado los ordenadores en función de su tamaño y complejidad.

La segunda parte del módulo se ha dedicado a presentar el software. Hemos visto cómo el software tiene como objetivo resolver problemas. Así, para crear un programa, primero se especifica un algoritmo en lenguaje natural que, posteriormente, se traducirá. El siguiente paso consta del proceso de compilación, mediante el que se transforma el código fuente en código máquina (o binario), que es lo que ejecuta un ordenador.

Finalmente, hemos presentado los diferentes tipos de software que existen. Lo que se ha visto con más detenimiento es el sistema operativo, el software que gestiona gran cantidad de los servicios del ordenador. Se ha mostrado cómo el sistema operativo ofrece soporte a otros programas para los usuarios finales que, como su nombre indica, son con los que habitualmente interactúan las personas. Para acabar, se ha presentado con un poco más de detalle un tipo de software de usuario final como es el de los sistemas de información. Éstos se encuentran en todas partes en entornos corporativos y domésticos, y con el apogeo de Internet están ganando una nueva dimensión.





## Actividades

1. Un amigo nuestro nos ha comentado que quiere aprender a programar. Lo primero que se le ha ocurrido es abrir con el editor de textos un archivo ejecutable de una aplicación (por ejemplo, word.exe). ¿Qué es lo que verá? ¿Podrá programar así? ¿Por qué no?

### Solución

Solución de la actividad 1

Al abrir un archivo ejecutable, verá el contenido en binario del archivo. De esta manera, es casi imposible programar, ya que es ininteligible para los seres humanos.

2. ¿Qué le aconsejaríamos a nuestro amigo si quisiera programar?

### Solución

Solución de la actividad 2

Para programar, lo mejor es que haga uso de un lenguaje de programación en el que podrá realizar las instrucciones pertinentes de manera más sencilla. Los lenguajes de programación permiten utilizar un lenguaje semiestructurado que es fácil de entender para los humanos.

3. Una vez que le hemos convencido, nuestro amigo ha escrito unas cuantas líneas de código fuente. Ahora quiere ejecutarlas, ¿qué debe hacer?

### Solución

Solución de la actividad 3

Compilarlo, de manera que las instrucciones en lenguaje de programación del código fuente sean transformadas en instrucciones en binario entendibles por el computador.

4. Estudiad la vida de Ada Lovelace, la primera programadora de la historia. A un lenguaje de programación le han dado su nombre; investigad en qué ámbitos se suele utilizar este lenguaje.

5. Queremos incluir en binario la información de nuestro nombre (en ASCII), nuestra fecha de nacimiento (una cifra para el día, otra para el mes y una tercera para el año) y nuestro lugar de nacimiento (en ASCII). ¿Cuántos bits ocuparía toda esta información?

6. Buscad en Internet una foto del primer ratón.

7. Tenemos dos puertas lógicas AND y NOT, encadenadas una detrás de otra. Primero se encuentra la puerta AND y, a continuación, su salida, que es la entrada de la puerta NOT. ¿Qué obtendríamos a la salida de la segunda puerta si en la entrada de la primera puerta tenemos los valores "Verdadero" y "Falso"?

8. Nombrad diferentes sistemas operativos que conozcáis. Investigad en la Wikipedia las familias que existen.

9. Enumerad una docena de programas que funcionen tanto en sistemas operativos Windows como en Linux.

10. Cread un algoritmo en lenguaje natural para llegar desde vuestro lugar de trabajo hasta la cafetería más cercana.

11. Buscad en el periódico una oferta actual de un ordenador de sobremesa y de un portátil. Comparad en ambos casos los diferentes tipos de memorias que incluyen, así como su precio.

## Ejercicios de autoevaluación

Indicad cuáles de la siguientes afirmaciones son correctas y cuáles son falsas.

1. Los antecesores de los ordenadores, las máquinas analíticas, fueron ideadas mucho antes del siglo XX, pero por problemas ingenieriles no pudieron ser construidas.

- a) Verdadero
- b) Falso

2. La primera generación de ordenadores modernos vino de la mano de los transistores.

- a) Verdadero
- b) Falso

3. Los entornos de ventanas fueron ideados en la década de los setenta por Xerox, aunque tardaron bastantes años en llegar al gran público.

- a) Verdadero
- b) Falso

4. El byte 00000111 significa exclusivamente el número 7 en binario.

- a) Verdadero
- b) Falso

5. El año 2009 ocupa en ASCII en total 4 bytes.

- a) Verdadero
- b) Falso

6. Cuando apagamos el ordenador, se borra la memoria principal (RAM).

- a) Verdadero
- b) Falso

7. Los programadores crean sus programas mayoritariamente en código binario.

- a) Verdadero
- b) Falso

8. El sistema operativo se encarga de gestionar los recursos de un ordenador.

- a) Verdadero
- b) Falso

9. Un algoritmo debe estar escrito en un lenguaje de programación de alto nivel.

- a) Verdadero
- b) Falso

10. Un *mainframe* es un ordenador central al que se le pueden conectar múltiples ordenadores denominados terminales.

- a) Verdadero
- b) Falso

## **Solucionario**

### **Ejercicios de autoevaluación**

1. a

2. b

3. a

4. b

5. a

6. a

7. b

8. a

9. b

10. a

## Glosario

**algoritmo** *m* Conjunto finito de pasos que sirven para resolver un problema o llevar a cabo una tarea.

**analista de software** *m* Persona, generalmente de perfil técnico, que realiza un análisis de los requisitos que ha de cumplir un futuro programa de ordenador.

**arquitectura (de ordenadores)** *f* Se trata de un concepto que engloba el diseño y el funcionamiento de los ordenadores que especifican, entre otros aspectos, el formato y el conjunto de instrucciones. Existen muchas arquitecturas, pero la más popular es i386, porque es la utilizada en los PC.

**byte** *m* Grupo de 8 bits.

**cobol** *m* Lenguaje de programación. Acrónimo de Common Organization Business Oriented Language (lenguaje común orientado a la organización de negocios), aunque también se puede encontrar como Common Business Oriented Language (lenguaje común orientado a negocios). Se trata de un lenguaje de programación de finales de 1960 cuyo propósito era poder ser utilizado en cualquier ordenador. Actualmente, su utilización es bastante limitada, aunque se encuentra con frecuencia en aplicaciones bancarias y programas antiguos.

**código binario** *m* Véase código máquina. Sistema de numeración basado en la utilización de dos valores o bits (típicamente '0' y '1'), en los que a su vez se basa el procesamiento automático de la información. Todo lo que se procesa con un ordenador acaba siendo transformado a código binario.

**código fuente** *m* (También conocido como fuentes.) Se trata de las instrucciones de ordenador escritas en un lenguaje de programación. En la fase de compilación se transforma en código máquina. Para que el software sea libre, el código fuente debe ser accesible, ya que, en caso contrario, la posibilidad de realizar modificaciones, aunque no sea imposible, se dificulta sobremanera.

**código máquina** *m* (También conocido como código binario.) Se trata del código que los ordenadores pueden ejecutar. Consta de unos y ceros, aunque existen otros modos de representación como octal o hexadecimal. El código máquina es difícilmente comprensible para los humanos –y la creación de complejas aplicaciones casi imposible–, por lo que se crearon los lenguajes de programación de alto nivel.

**compilador** *m* Se encarga principalmente de traducir los ficheros escritos a lenguajes de programación (comprensibles para los humanos) en código máquina (unos y ceros, comprensibles generalmente sólo por los ordenadores).

**computador** *m* Artificio capaz de hacer cálculos y mostrar resultados sobre los datos numéricos que se introducen. Su complejidad ha ido aumentando y su finalidad ha ido variando, hasta convertirse en aparatos como las calculadoras de bolsillo o los ordenadores personales actuales.

**C/C++** *m* Lenguajes de programación. C fue creado a principios de los años setenta para el desarrollo de Unix; se trata de uno de los primeros lenguajes de programación que permiten la programación de alto nivel. C++ es una versión posterior de C que añade técnicas modernas de programación. La mayoría del código libre está escrito en C o C++.

**depuración** *f* Proceso mediante el que se buscan (y corrigen) errores en el software. Véase también *depurador*.

**depurador** *m* Programa de ordenador utilizado a la hora de crear software para encontrar y corregir errores de programación.

**drivers** *m* Software que se encarga de interactuar entre el sistema operativo y los dispositivos (hardware).

**hardware** *m* Conjunto de dispositivos físicos que componen el ordenador: la pantalla, el teclado, el ratón, etc.

**i386** Arquitectura de ordenador típica de los ordenadores personales (PC).

**Java** *m* Lenguajes de programación. Lenguaje de programación creado en los años noventa con muchas similitudes sintácticas con C y C++, pero con características de programación de más alto nivel de abstracción.

**kernel** *m* Núcleo del sistema operativo. Es el que se encarga de las labores de más bajo nivel (el nivel más cercano al hardware), como la gestión de la memoria, la entrada/salida de dispositivos, etc. El kernel más popular en el mundo del software libre es Linux, aunque existen muchos más (por ejemplo, los sistemas BSD tienen uno propio).

**lenguaje ensamblador** *m* Lenguaje de bajo nivel de abstracción que permite crear programas de ordenador. Se trata del lenguaje de programación más cercano al código máquina.

**lenguaje de programación** *m* Conjunto de reglas semánticas y sintácticas utilizadas para dar instrucciones a un ordenador. Los lenguajes de programación permiten trabajar a un nivel de abstracción superior que con código máquina, lo que facilita la creación y mantenimiento de programas informáticos. Existen miles de lenguajes de programación. Algunos ejemplos son C, C++, ADA, Java, Pascal y COBOL.

**lenguaje natural** *m* Lenguaje escrito o hablado por los humanos con fines de comunicación.

**microprocesador** *m* Elemento del hardware del sistema informático encargado de procesar la información, así como de gestionar la ejecución de los pedidos que conforman los programas y el traspaso de datos entre los diferentes componentes del computador.

**modelo de desarrollo en cascada** *m* Procedimiento para crear software, basado en dividir el proceso de desarrollo en una secuencia de tareas que se realizan una detrás de otra. Las tareas se llevan a cabo siguiendo el orden siguiente: definición de los requisitos, análisis y diseño de software, implementación y prueba de unidades, integración y pruebas del sistema y operación y mantenimiento.

**multicomputador/multiprocesador** *m* Nombre genérico que recibe un sistema informático basado en el uso de más de un computador o procesador.

**Pascal** *m* Lenguaje de programación de la década de los setenta escasamente utilizado hoy en día, aunque algunos de sus sucesores sí que cuentan con amplio eco en la industria del software.

**sistema operativo** *m* Programa responsable de la gestión y coordinación de actividades y del reparto de recursos de un ordenador.

**software** *m* Componente intangible en la informática. Generalmente, se trata de una serie de instrucciones elaboradas por humanos en lenguajes de programación de alto nivel (código fuente) que luego son traducidas por un compilador a código máquina (unos y ceros comprendidos por las máquinas). El software se divide en software de sistema, parte que corresponde a los sistemas operativos, o de aplicación, que agrupa los programas que el usuario suele utilizar. Estrictamente, el software también incluye la documentación del programa, aunque ésta se encuentre en un manual.

**tabla de verdad** *f* Una tabla de verdad específica, para las diferentes combinaciones de los bits de entrada, cuál es la salida de una función lógica o de un circuito de cálculo.

**tiempo de ejecución** *m* Se trata del espacio temporal en el que el ordenador está ejecutando las instrucciones correspondientes a un programa de software.

## Bibliografía

**Angulo, J. M. et al.** (2003). *Fundamentos y estructura de computadores*. Thomson Paraninfo.

**Barceló, M.** (2008). *Una historia de la Informática*. Barcelona: Editorial UOC.

**Beekman, G.** (2004). *Computer Confluence. Standard*. (6.<sup>a</sup> ed.) Addison-Wesley.

**Beekman, G.** (2004). *Introducción a la informática*. Pearson Prentice Hall.

**Leiva Olivencia, J. L.** (2009). *Informática para Estudiantes*. Editorial Abecedario. ISBN 9788492669011.

**Pareja, C.; Andreyo, Á.; Ojeda, M.** (1993). *Introducción a la Informática*. ISBN: 84-7491-489-2.

**Reina, Pedro.** *Curso de Informática*.

**Reynolds, G. W. y otros** (2000). *Principios de sistemas de información*. (4.<sup>a</sup> edición) Thomson Paraninfo.

# Aspectos tecnológicos de las redes e Internet

Antoni Martínez Ballesté

PID\_00150270



Universitat Oberta  
de Catalunya

[www.uoc.edu](http://www.uoc.edu)



Los textos e imágenes publicados en esta obra están sujetos –excepto que se indique lo contrario– a una licencia de Reconocimiento-NoComercial-SinObraDerivada (BY-NC-ND) v.3.0 España de Creative Commons. Podéis copiarlos, distribuirlos y transmitirlos públicamente siempre que citéis el autor y la fuente (FUOC. Fundació per la Universitat Oberta de Catalunya), no hagáis de ellos un uso comercial y ni obra derivada. La licencia completa se puede consultar en <http://creativecommons.org/licenses/by-nc-nd/3.0/es/legalcode.es>



# Índice

|                                                                     |           |
|---------------------------------------------------------------------|-----------|
| <b>Introducción.....</b>                                            | <b>5</b>  |
| <b>Objetivos.....</b>                                               | <b>6</b>  |
| <b>1. Las redes de computadores.....</b>                            | <b>7</b>  |
| 1.1. Tipo de redes .....                                            | 8         |
| 1.1.1. Redes según su tamaño .....                                  | 8         |
| 1.1.2. Redes según el sistema de comunicación .....                 | 9         |
| 1.2. Organismos reguladores de las telecomunicaciones e Internet .. | 10        |
| 1.3. Protocolos de comunicaciones .....                             | 11        |
| <b>2. La transmisión de información.....</b>                        | <b>13</b> |
| 2.1. El medio físico .....                                          | 13        |
| 2.2. Envío de tramas .....                                          | 14        |
| 2.3. El medio compartido .....                                      | 16        |
| <b>3. Las redes de área local.....</b>                              | <b>18</b> |
| 3.1. Tecnologías LAN .....                                          | 19        |
| 3.1.1. Ethernet .....                                               | 19        |
| 3.1.2. LAN inalámbrica .....                                        | 21        |
| 3.1.3. Redes de área local de gran alcance .....                    | 23        |
| 3.2. Ejemplos de LAN .....                                          | 24        |
| 3.2.1. Una red para una pequeña oficina .....                       | 24        |
| 3.2.2. Una LAN corporativa .....                                    | 26        |
| <b>4. Internet.....</b>                                             | <b>28</b> |
| 4.1. Evolución de Internet .....                                    | 28        |
| 4.2. La interconexión de redes .....                                | 29        |
| 4.2.1. Datagramas .....                                             | 30        |
| 4.2.2. Direcciones IP .....                                         | 30        |
| 4.2.3. <i>Routers</i> .....                                         | 31        |
| 4.2.4. Estructura de Internet .....                                 | 32        |
| 4.3. El transporte de la información .....                          | 35        |
| 4.3.1. El protocolo TCP .....                                       | 35        |
| 4.3.2. Los puertos .....                                            | 36        |
| 4.3.3. El empaquetado .....                                         | 36        |
| 4.4. Los nombres de dominio .....                                   | 37        |
| <b>5. El correo electrónico.....</b>                                | <b>39</b> |
| 5.1. Direcciones y mensajes .....                                   | 40        |
| 5.2. Envío de correos .....                                         | 42        |

|                                             |    |
|---------------------------------------------|----|
| 5.3. Lectura de un correo electrónico ..... | 43 |
| <b>Resumen</b> .....                        | 46 |
| <b>Actividades</b> .....                    | 47 |
| <b>Ejercicios de autoevaluación</b> .....   | 47 |
| <b>Solucionario</b> .....                   | 48 |
| <b>Glosario</b> .....                       | 50 |
| <b>Bibliografía</b> .....                   | 51 |

## Introducción

Desde la proliferación de los sistemas informáticos, los computadores se han convertido en una pieza clave en las redes de comunicaciones. Para poner un ejemplo, el proceso de digitalización de las redes telefónicas fue acompañado por la adopción de conmutadores telefónicos basados en computadores. Paralelamente, los centros de investigación y militares hicieron uso de estas redes para implantar lo que acabaría siendo Internet. Durante los años ochenta del siglo XX, diferentes sistemas de información remotos estaban disponibles para los usuarios que se conectaran mediante una llamada de teléfono.

En los entornos de oficina, la incorporación de ordenadores personales como sustitución de los antiguos terminales conectados a *mainframes* propició la adopción de estándares para redes de área local que dieran soporte a la compartición de recursos.

Finalmente, el estallido de Internet, el acceso de banda ancha y el aumento del número de ordenadores por habitante han hecho de las redes de computadores un elemento clave en el desarrollo de la sociedad de la información.

En este módulo, estudiaremos el funcionamiento de las redes de computadores. Sin ánimos de ser exhaustivos ni de utilizar un lenguaje orientado a ingenieros, veremos aspectos básicos de las redes más habituales en nuestro entorno y cómo influyen las tecnologías de Internet.

En primer lugar, estudiaremos las bases esenciales de las redes de computadores. En segundo lugar, presentaremos el funcionamiento y uso de las tecnologías de red de área local, ya que son tecnologías muy extendidas en entornos laborales e incluso a nivel doméstico. Una parte del módulo estará dedicada a explicar la evolución y el funcionamiento de Internet (basado en el conjunto de protocolos conocido como TCP/IP). Finalmente, analizaremos el funcionamiento del correo electrónico, lo que nos permitirá ubicar sobre un escenario real algunos de los conceptos estudiados a lo largo del módulo.

## Objetivos

Los objetivos que el estudiante habrá alcanzado al finalizar este módulo son:

- 1.** Conocer los diferentes tipos de redes de computadores.
- 2.** Comprender las bases de la transmisión de información en una red de computadores.
- 3.** Distinguir los elementos que forman parte de una red de área local.
- 4.** Comprender las bases del funcionamiento de Internet.
- 5.** Conocer algunos de los servicios de comunicación que funcionan en Internet.
- 6.** Relacionar los aspectos descritos en el módulo con aplicaciones reales.

## 1. Las redes de computadores

Multitud de procesos, actividades y servicios se fundamentan o complementan por medio de las redes de comunicaciones. Las hay de varios tipos, por ejemplo las redes de telefonía fija y móvil, las redes en procesos de fabricación automatizados, las redes de difusión de televisión y radio, etc. Las redes de computadores son el sistema resultante de interconectar ordenadores mediante una determinada tecnología de transmisión de datos. La función de esta interconexión es, básicamente, la comunicación de información entre los ordenadores que están conectados.

En este módulo nos centraremos en las redes de computadores, dado que son la herramienta fundamental en la que se basa y desarrolla gran parte de lo que llamamos *sociedad de la información*.

Hoy en día podemos encontrar una gran variedad de dispositivos conectados a una red de computadores, como:

- Ordenadores personales, para realizar tareas ofimáticas, de diseño, administrativas, de comunicación mediante Internet, de ocio, etc.
- Ordenadores utilizados como servidores, con gran capacidad de cálculo y almacenamiento, para gestionar información como servicios web, correo electrónico, bases de datos, etc.
- Ordenadores móviles, como portátiles o PDA conectados por medio de tecnologías inalámbricas.
- Discos duros, impresoras y otros periféricos conectados directamente a la red.
- Finalmente, debemos mencionar los equipos dedicados a interconectar equipos y redes, como los conmutadores y los *routers*.

Sin embargo, la proliferación de los ordenadores como parte integrante de coches, grabadoras de vídeo, teléfonos móviles, etc., permite que estos dispositivos también puedan formar parte de una red de computadores.

Uno de los principales usos de las redes de computadores es compartir recursos, como la información en una página web o una base de datos.

El objetivo principal de las redes de computadores es la interconexión de equipos con el propósito fundamental de compartir, enviar y recibir información.

### Ejemplo de redes de computadores

Algunos casos de redes de computadores son:

- En una oficina pequeña hay cinco ordenadores personales conectados a una red de área local. Comparten una conexión a Internet y una impresora láser. Hay un sexto ordenador, más potente, que hace de servidor de un sistema de gestión contable. Todo el mundo puede imprimir utilizando la impresora compartida y todo el mundo puede navegar por Internet.
- En una productora de cine de animación, disponen de 16 ordenadores potentes conectados a una red de área local. Los diseñadores, para obtener una secuencia de la película, reparten el proceso de renderización entre los 16 ordenadores. Así pues, cada ordenador procesará una parte de la secuencia y tendrán el producto final en menos tiempo que si lo hubiera procesado uno solo.
- Un grupo de amigos se reúne para jugar a un juego de ordenador en red. Con sus ordenadores portátiles establecen una red inalámbrica para poder jugar juntos.
- En un hogar, en el estudio, hay un ordenador con un lector de tarjetas y una grabadora de televisión digital. En la sala, hay un disco duro multimedia con una conexión en red de área local. A este disco se envían las fotografías tomadas con la cámara digital, las grabaciones de programas de televisión, etc. Este disco tiene acceso a Internet, de modo que se puede conectar con servicios de radio y vídeo a Internet.
- Una empresa con sede en Barcelona está conectada en Internet con las delegaciones de Madrid y Valencia. Las tres sedes comparten un mismo sistema de información, ubicado en Barcelona. Para establecer llamadas telefónicas, también utilizan la conexión a Internet.

En los próximos subapartados, iremos viendo las tecnologías que hacen posibles éstos y otros escenarios.

#### Proceso de renderización

La renderización es un proceso para generar una imagen o animación en 3D a partir de un modelo descrito en un lenguaje específico utilizando una aplicación informática.

## 1.1. Tipo de redes

Clásicamente, las redes de computadores tienen diferentes clasificaciones en función de qué propiedad se utilice para clasificarlas. En nuestro caso, veremos una clasificación en función del tamaño de la Red y en función del sistema de comunicación.

### 1.1.1. Redes según su tamaño

Según las dimensiones físicas de la red, nos podemos encontrar con los siguientes casos:

- Una **red de área personal**<sup>(1)</sup> (PAN) es una red formada por dispositivos separados por pocos metros. Por ejemplo, la conexión de un teléfono móvil con el equipo de audio de un coche, la conexión de un ordenador con sus periféricos por medio de conexión inalámbrica, etc.
- Una **red de área local**<sup>(2)</sup> (LAN) se caracteriza por la proximidad de los equipos conectados (hasta 100 metros aproximadamente) y por la alta veloci-

<sup>(1)</sup>En inglés *Personal Area Network*.

<sup>(2)</sup>En inglés *Local Area Network*.

dad de transferencia de datos que ofrece. Sin embargo, hoy en día es posible tener conexiones de gran alcance que ofrecen grandes velocidades, con lo que podemos tener una LAN formada por redes ubicadas en diferentes puntos de una ciudad.

- Una **red de área metropolitana**<sup>3</sup> (MAN), que abarca generalmente el área de una ciudad. Por ejemplo, la red de distribución de televisión por cable de una operadora de comunicaciones.
- Una **red de área extendida**<sup>4</sup> o **de largo alcance** (WAN) se caracteriza por que cubre un gran territorio. Un posible ejemplo sería una red de datos que interconecte todas las universidades de un país.

**Internet** se podría considerar como la mayor red de computadores, resultante de la interconexión de varias redes de área extendida destinadas a la transmisión de datos.

### 1.1.2. Redes según el sistema de comunicación

Según cómo se transmita la información, una red de computadores se puede clasificar en:

- **Redes punto a punto**, que conectan dos dispositivos. Un ejemplo sería una conexión en Internet entre una cámara de vigilancia ubicada en una ciudad y un ordenador ubicado en otra.
- **Redes de difusión**<sup>5</sup>, en las que, debido a las características del sistema de transmisión de datos, todas las máquinas que se encuentran dentro del alcance de un emisor recibirán la información que éste genera, sean o no receptoras. Un ejemplo son las redes LAN inalámbricas, en las que todos los ordenadores que se encuentran al alcance del punto de acceso (término que estudiaremos más adelante) pueden acceder a la información.
- **Redes de difusión selectiva**<sup>6</sup>, en las que la información se envía a un grupo de máquinas. En contraposición a las redes de difusión, las máquinas que no pertenecen al grupo no la reciben. Un ejemplo sería la televisión basada en Internet, por medio de la que se envía una película solamente a los usuarios que han abonado la tarifa correspondiente.

En la figura siguiente podemos observar los tipos de redes según cómo se accede a la información:

- **Modelo cliente-servidor.** Éste es el modelo más clásico, en el que unas máquinas (los clientes) piden la información a otras (servidores). Por ejemplo, pedir el estado de la cuenta corriente al servicio web de nuestra caja de ahorros. El servidor es una máquina potente o un conjunto de máqui-

#### Redes SOHO

Las redes para los hogares o las pequeñas oficinas reciben el nombre de SOHO (del inglés *Small Office, Home Office*).

<sup>(3)</sup>En inglés *Metropolitan Area Network*.

<sup>(4)</sup>En inglés *Wide Area Network*.

<sup>(5)</sup>En inglés, *broadcast*.

#### Ved también

Podéis ver la definición de punto de acceso en el subapartado 3.1.

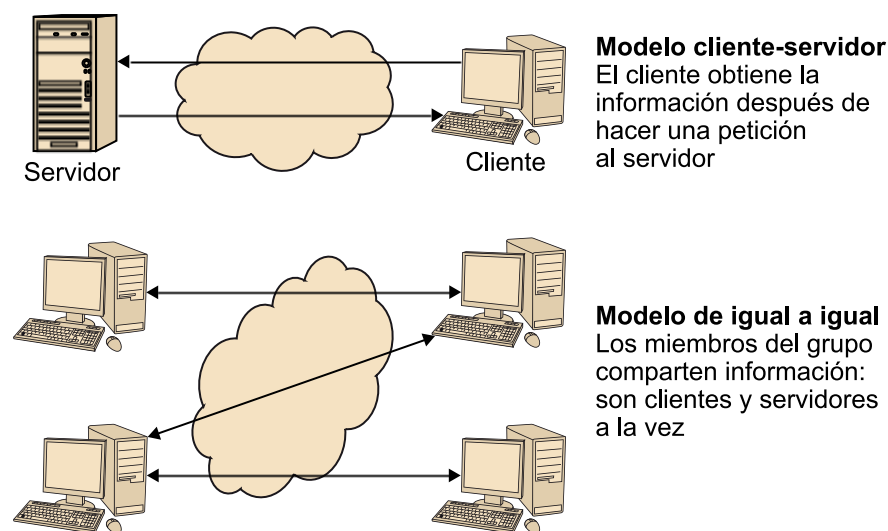
<sup>(6)</sup>En inglés *multicast*.

nas trabajando conjuntamente, como hemos visto en el primer módulo. El servidor funciona las 24 horas del día, ya que necesitamos acceder a la cuenta corriente desde cualquier rincón del mundo. Los servidores contienen y centralizan la información a la que acceden los clientes.

- **Redes de igual a igual<sup>7</sup> (P2P).** En estas redes no hay un servidor único que contenga toda la información. De hecho, todos los miembros de la red tienen el mismo papel: los clientes ejercen al mismo tiempo de servidores de información. Existen ejemplos de programas informáticos de compartición de ficheros basados en este tipo de redes.

<sup>(7)</sup>En inglés *Peer-to-Peer*.

Tipo de modelos de acceso a la información



## 1.2. Organismos reguladores de las telecomunicaciones e Internet

Desde la aparición de los primeros sistemas de transmisión de información, una serie de organismos se han dedicado a su regulación y estandarización. A continuación, destacamos los más relevantes:

- El **Instituto de Ingenieros Eléctricos y Electrónicos<sup>8</sup> (IEEE)** es una asociación profesional que se encarga de la estandarización de muchos sistemas electrónicos de transmisión. En relación con las redes de computadores, tenemos los estándares referentes a las tecnologías de redes de área local. Es muy antigua, ya que funciona desde el siglo XIX (el nombre actual tiene fecha de 1963).
- La **Unión Internacional de Telecomunicaciones<sup>9</sup> (ITU)**, data del año 1934 y depende de las Naciones Unidas. Entre los aspectos que regula, encontramos algunos estándares de videoconferencia, transmisión de información por las líneas telefónicas, etc.

<sup>(8)</sup>En inglés, *Institute of Electrical and Electronic Engineers*.

<sup>(9)</sup>En inglés, *International Telecommunication Union*.



- El **Grupo de Trabajo en Ingeniería de Internet**<sup>(10)</sup> (IETF) es una organización que se encarga de los estándares relacionados con las aplicaciones y protocolos en los que se fundamenta Internet. Se creó en el año 1986.

<sup>(10)</sup>En inglés, *Internet Engineering Task Force*.

## IETF

Los documentos mediante los que se publican los acuerdos de la IETF se llaman RFC (*Request For Comments*). Se pueden consultar en su web. Los hay de todo tipo. ¡Incluso el documento número 1.149 trata sobre cómo enviar información con palomas mensajeras!

- La **Corporación de Internet para Nombres y Números Asignados**<sup>(11)</sup> (ICANN), es una organización internacional que es responsable de administrar los dominios de Internet y la numeración de direcciones IP.
- La **Sociedad Internet**<sup>(12)</sup> (ISOC) es un organismo que se dedica a promover el uso de Internet por parte de la sociedad.
- Finalmente, debemos nombrar a la **Organización Internacional para la Estandarización**<sup>(13)</sup> (ISO). Este organismo se encarga de promover el desarrollo de normas internacionales de fabricación, comercio y comunicación en muchos ámbitos. En el terreno de la informática y de las comunicaciones, es responsable de recoger estándares, como el de compresión de vídeo MPEG.

<sup>(11)</sup>En inglés, *Internet Corporation For Assigned Names and Numbers*.

## Ved también

Los conceptos de dirección IP y dominios de Internet se verán en el subapartado 4.2.

<sup>(12)</sup>En inglés, *Internet Society*.

<sup>(13)</sup>En inglés, *International Organization for Standardization*.

## 1.3. Protocolos de comunicaciones

La gran mayoría de estándares relacionados con las redes de computadores describen protocolos de comunicaciones en un determinado nivel.

Un protocolo de comunicaciones establece los pasos para llevar a cabo una comunicación de información en un nivel específico, es decir, el envío de mensajes entre los comunicantes, así como el contenido de estos mensajes.

De hecho, la ISO, con el apoyo de la ITU, publicó en el año 1977 unas directrices para estandarizar los sistemas y aplicaciones basados en los **sistemas abiertos** (sistemas que se interconectan formando, precisamente, redes de computadores). La Interconexión para Sistemas Abiertos<sup>(14)</sup> (OSI) recoge estas directrices. Hasta que el ISO no presentó las directrices OSI, era muy complicado que las tecnologías de red de un fabricante pudieran interactuar con soluciones de otros fabricantes.

<sup>(14)</sup>En inglés, *Open Systems Interconnection*.

En el OSI, se especifica que las propuestas de interconexión de sistemas se deben ubicar dentro de uno de los siete niveles siguientes: nivel 1 (físico), nivel 2 (enlace), nivel 3 (red), nivel 4 (transporte), nivel 5 (sesión), nivel 6 (presentación) y nivel 7 (aplicación).

### **Ejemplo de protocolo OSI**

Un protocolo ubicado en el nivel físico del OSI tratará sobre cómo enviar bits sobre un determinado medio de transmisión, como la fibra óptica. Un protocolo ubicado en el nivel de aplicación tratará, por ejemplo, de cómo acceder a la cuenta de correo electrónico y descargar los mensajes.

### **El modelo Internet**

Aunque el ISO presentó un modelo para conseguir el objetivo de hacer posible la interconexión de sistemas abiertos, en la práctica la interconexión entre sistemas se ha llevado a cabo mediante protocolos diseñados para Internet. Estos protocolos a menudo no corresponden sólo a un nivel del OSI, sino que corresponden a varios niveles.

## 2. La transmisión de información

En este apartado, nos centraremos en comprender cuáles son las bases del envío de información en las redes de computadores. En primer lugar, describiremos el funcionamiento de los medios físicos que se utilizan y, finalmente, profundizaremos en los mecanismos del envío de mensajes sobre este medio físico.

### 2.1. El medio físico

Las redes de computadores utilizan diferentes medios físicos para la interconexión de equipos. A continuación, los describimos.

Por una parte, están los medios guiados, es decir, cables por los que se transmite la información. Entre éstos destacan:

- El **cable de pares trenzados**<sup>15</sup>. Se trata de un par de hilos de cobre delgados que forman trenzas. Estos pares se encuentran agrupados dentro de un tubo. Los diferentes diámetros del cobre y la cantidad de trenzas establecen diferentes categorías de estos cables, y los más utilizados son los de categoría 5. Uno de los usos más conocidos es la conexión de los ordenadores en una red de área local, por medio de un conector RJ-45.
- La **fibra óptica**, que permite velocidades muy elevadas, se utiliza básicamente para interconectar redes que precisan de estas velocidades (por ejemplo, redes de operadoras de comunicaciones o de distribución de televisión por cable).

<sup>(15)</sup>En inglés, *twisted pair*.



A la izquierda, un tubo de pares trenzados, y a la derecha, un cable con el conector RJ-45 para una red de área local.

#### El ancho de banda

El concepto "ancho de banda" puede tener varios significados. En este módulo, entenderemos por ancho de banda la cantidad de información por segundo que permite un determinado sistema de transmisión. A partir de un bit de información por segundo (1 bit/s), podemos ir haciendo múltiplos: 1 kbit/s son 1.000 bit/s; 1 Mbit/s son 1.000 kbit/s, etc.

En el caso del cable trenzado, la señal que circula es electromagnética y se comporta como una onda. En la fibra óptica viaja, literalmente, un rayo de luz.

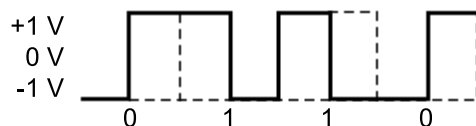
En contraposición a los medios guiados, varias tecnologías de comunicación utilizan el vacío como medio físico. Por ejemplo, las redes inalámbricas utilizan este medio físico. Otro sistema muy utilizado son los radio-enlaces, típicamente para conexiones de gran ancho de banda.

### Ejemplos de cómo se envía la información sobre el medio

A la hora de especificar cómo se envía la información sobre el medio, encontramos un grupo de posibilidades. Para ejemplarizarlo, sin pretender ser exhaustivos, veremos dos ejemplos:

- Los unos y ceros que se envían **sobre una onda electromagnética** (sea en el vacío o por medio de un cable) incidirán en la forma de esta onda. Para transmisiones digitales, el bit '0' se podría transmitir como un paso de voltaje negativo en positivo y el bit '1' en el paso de voltaje positivo a negativo.

Envío de bits, en concreto 0110, sobre una onda electromagnética digital



- Los bits se envían sobre la fibra óptica **encendiendo y parando el emisor**. De esta manera, y para hacerlo sencillo, un '1' implicaría emitir luz mediante la fibra durante un tiempo determinado (habitualmente, un tiempo del orden de nanosegundos), mientras que un '0' implica no emitir.

Los protocolos que definen cómo se envían los bits sobre el medio pertenecen al nivel físico del OSI.

### Transmisiones analógicas

Muchos sistemas de transmisión envían la información digital como si de una señal analógica se tratara. Por ejemplo, las tecnologías de redes inalámbricas, que estudiaremos en el subapartado 3.1.

Para hacerlo, se utilizan conjuntos de bits que se envían de golpe (símbolos de un código). Cada símbolo representa un cambio de amplitud y/o fase en la onda que se emite.

## 2.2. Envío de tramas

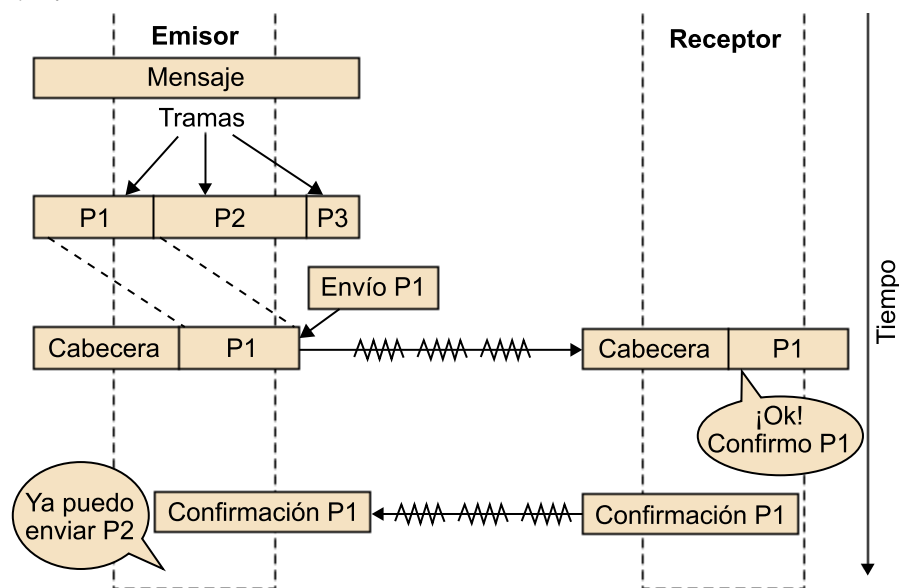
Una vez que hemos visto cómo se envían los bits mediante las redes de computadores (y, en general, en las redes de comunicaciones), veremos cómo se organiza la información a la hora de enviarla por medio de un enlace punto a punto, es decir, de un nodo emisor hacia un nodo receptor. En este caso, estamos hablando de protocolos del nivel de enlace del OSI.

Vamos a verlo de manera didáctica sobre un ejemplo. Supongamos que hay dos computadores unidos por una fibra óptica y que queremos enviar un texto de 1.100 bytes de longitud mediante esta conexión. La figura siguiente muestra el ejemplo.

#### Más pequeños que el segundo

Un milisegundo (ms) es la milésima parte del segundo. Un microsegundo ( $\mu$ s) es la milonésima parte del segundo. ¡Un nanosegundo (1 ns) es la millonésima parte de un segundo!

## Ejemplo de envío de tramas



Lo primero que hará el emisor es dividir la información en paquetes llamados **tramas**. Cada protocolo de comunicación tiene un **tamaño máximo de paquete**<sup>16</sup>. En nuestro protocolo de enlace, el tamaño máximo de trama es de 500 bytes. Esto significa que el mensaje se dividirá en tres porciones: las tramas P1 y P2, de 500 bytes cada una, y la trama P3, con los 100 bytes restantes.

<sup>(16)</sup>En inglés, se llama *Maximum Transfer Unit* (MTU).

Antes del envío, el protocolo de enlace añadirá una serie de información complementaria en la trama, en un espacio inicial llamado **encabezamiento**:

- Un **indicador de inicio de trama**. Habitualmente, se trata de un patrón de bits y sirve para indicar al receptor que se empieza a recibir información.
- La **longitud de la trama**, a fin de que el receptor sepa cuándo se acaba la trama y pueda darla por recibida.
- Un **código de comprobación**, a fin de que el receptor detecte si ha habido algún error durante la transmisión (por ejemplo, que un '0' se hubiera visto afectado por una interferencia y se hubiera convertido en un '1').

### Códigos de comprobación

Los códigos de comprobación permiten comprobar, con casi total seguridad, que la información recibida o registrada es correcta. Un ejemplo cotidiano es la letra del NIF. Ésta se obtiene con una fórmula matemática y permite comprobar que el número de DNI se ha introducido correctamente. A partir del número, se calcula la letra, y si difiere de la que debería ser, es porque el DNI se ha introducido mal.

El paquete se enviará emitiendo los '0' y los '1' de los bytes por medio de la fibra óptica, tal como hemos visto anteriormente.

Una vez que la información ha llegado al receptor, éste comprobará su validez (haciendo uso del código de comprobación). Si todo ha ido bien, el receptor responderá con una **trama especial de confirmación**, e indicará que la infor-

mación se ha recibido correctamente. Una vez recibida esta confirmación, el emisor da P1 por enviado y ya puede enviar P2. Notad que si no se hubiera recibido nada, el emisor, al haber pasado un rato y no haber recibido ninguna confirmación, reenviaría la trama.

### 2.3. El medio compartido

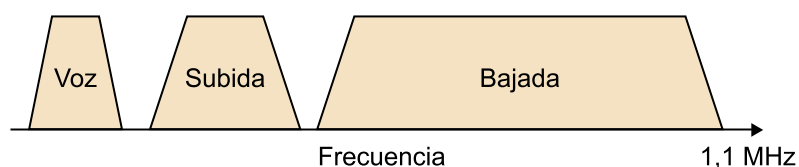
En el ejemplo del subapartado anterior sólo había un emisor y un receptor unidos por una fibra óptica. En el mundo de las comunicaciones, y de las redes de computadores en particular, el medio que se utiliza para comunicarse suele estar compartido. Con una serie de ejemplos iremos viendo diferentes maneras de compartir el medio.

En el caso de la televisión o la radio, existen diferentes canales y emisoras que están compartiendo el medio. A fin de que no haya problemas, hay una regulación del espectro radioeléctrico: se tiene cuidado de que cada uno de los canales tenga asignada una frecuencia determinada y que no haya más de un canal usando la misma frecuencia. Este sistema se llama **multiplexación por división de frecuencia** y no sólo se utiliza en la radio y la televisión.

Por ejemplo, los sistemas de línea de abonado digital asimétrica<sup>17</sup> (ADSL) utilizan este sistema para conectar la red de computadores de casa a Internet. Como se puede ver en la figura siguiente, por el cable telefónico circulan tres tipos de información, cada uno por su frecuencia asignada: la voz de las llamadas telefónicas, la información digital que viene de Internet (bajada) y la información digital que nosotros enviamos a Internet (subida).

<sup>(17)</sup>En inglés, *Asymmetric Digital Subscriber Line*.

División de la frecuencia en el sistema ADSL



Si lo que se está compartiendo es una fibra óptica, se tiende a realizar una **multiplexación por división del tiempo**. Supongamos que una misma fibra está siendo utilizada por cuatro comunicaciones. En ese caso, la fibra estará disponible durante un instante determinado de tiempo para la comunicación número 1; el siguiente instante de tiempo lo estará para la comunicación 2 y así sucesivamente. Una vez haya acabado la comunicación número 4, la fibra volverá a estar disponible para la comunicación 1.

Otro método de compartición del acceso en el medio se basa en la distribución de éste por parte de un **dispositivo maestro**. Por ejemplo, en la tecnología Bluetooth, los dispositivos próximos forman una red llamada piconet. En cada

piconet se elige un dispositivo maestro que va preguntando a los demás dispositivos (que hacen las funciones de **esclavo**) quién debe utilizar el medio. En el caso de que alguien lo necesite, lo tendrá disponible durante cierto tiempo.

### 3. Las redes de área local

Una vez que hemos visto cómo se transmite la información sobre un medio físico, nos centraremos en el funcionamiento de las redes que tenemos más al alcance, ya sea en el trabajo o en casa: las redes de área local<sup>18</sup>.

En primer lugar, revisaremos cuáles son las funcionalidades de una red de área local. Acto seguido, estudiaremos los conceptos fundamentales de las tecnologías en las que se basan.

Las redes de computadores aparecieron cuando los terminales simples conectados a ordenadores principales fueron siendo sustituidos por ordenadores personales. La compartición de recursos es una de las finalidades de la interconexión en red dentro de un entorno en área local. A continuación, citamos los más habituales:

- **Compartición de impresoras.** Los diferentes trabajadores comparten un par de impresoras que también se encuentran conectadas a la red. De esta manera, no hace falta que cada ordenador tenga una impresora. Además, una impresora puede ser de tecnología láser para grandes volúmenes de copias y la otra puede ser una impresora de inyección con alta calidad fotográfica.
- **Copias de seguridad.** Los trabajadores guardan sus documentos en una carpeta que, a pesar de ser accesible desde su entorno de trabajo del ordenador, se encuentra físicamente en un ordenador que hace de servidor. Cada noche, este ordenador envía una copia de seguridad hacia un disco duro que se encuentra conectado a la red. De esta manera, no hace falta que cada ordenador cuente con un sistema propio de copias de seguridad: los documentos estarán guardados en el ordenador en el que se realizan las copias. La alta velocidad de la red permitirá trabajar como si los documentos estuvieran guardados en los discos duros de los ordenadores de los trabajadores.
- **Conexión compartida en Internet.** Todo el mundo puede navegar por Internet gracias a la conexión compartida. Sería impensable proveer con una conexión a Internet a cada uno de los ordenadores. Esta conexión se realiza mediante un punto de conexión que dispone de los mecanismos de seguridad adecuados para proteger los ordenadores de la red de los ataques informáticos que pueden venir de Internet.

(18) De ahora en adelante, identificadas por su acrónimo en inglés: LAN.

#### Caracterización de una LAN

Recordad que una LAN se caracteriza por la proximidad de los equipos conectados y por la alta velocidad de transmisión de datos.



- **Soporte para el sistema de telefonía.** El sistema interno de comunicación se lleva a cabo a través del propio ordenador: con unos teléfonos de voz sobre IP, la empresa ha conseguido un ahorro considerable al prescindir de una centralita telefónica.

#### La voz sobre IP

Las tecnologías de voz sobre IP (VoIP) permiten el uso de una red de computadores (como una red de área local o con Internet) para realizar llamadas de voz.

### 3.1. Tecnologías LAN

En este subapartado, estudiaremos, desde un punto de vista práctico, las tecnologías que posibilitan las LAN. Más adelante ilustraremos, en términos de equipos y características, diferentes tipos de LAN que podemos encontrar en nuestro entorno más próximo. Las tecnologías que estudiaremos definen protocolos tanto en el nivel físico como en el nivel de enlace.

#### 3.1.1. Ethernet

Las tecnologías Ethernet datan de mediados de la década de los setenta. La idea básica era que diferentes máquinas estaban conectadas a un cable (llamado *ether*) para enviar y recibir información.

Como el medio era compartido, y no era viable prever diferentes canales para cada estación, se tuvo que implementar un sistema de acceso al medio basado en el acceso múltiple por medio de escuchar a la portadora de la señal (CS-MA<sup>19</sup>). Este sistema consiste en escuchar al medio (mirar si hay voltaje en el cable) y, si nadie lo utiliza, transmitir la información. Si alguien está enviando información, hay que esperar un tiempo aleatorio antes de volver a intentarlo.

<sup>(19)</sup>En inglés, *Carrier Sense Multiple Access*.

Las primeras Ethernet permitían una velocidad de 10 Mbit/s. Éstas podían tener dos topologías físicas de las máquinas:

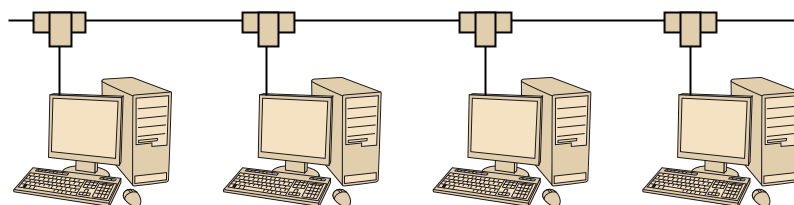
- Topología de **bus**, por medio de cables coaxiales, parecidos a los de las antenas de televisión. Las máquinas se conectaban formando una cadena. Nótese que si fallaba la conexión en cualquier punto, fallaba toda la red.
- O bien topología de **estrella**, en la que las máquinas se conectan a un concentrador (en inglés *hub*) con cables trenzados y conectores RJ-45. Esta disposición es la que ha perdurado hasta la actualidad.

#### Puertos de conexión

Los concentradores disponen de conexiones hembra llamadas puertos de conexión. Hay aparatos con 4, 8, 16 e incluso más puertos.

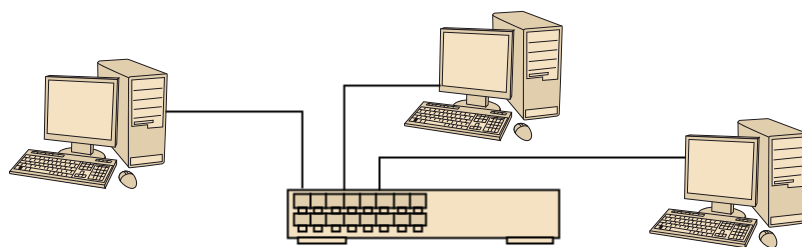
## Tipo de topologías

### Topología de bus



Todo lo que emite un ordenador se extiende por el cable

### Topología de estrella



Todo lo que emite un ordenador lo propaga el *hub*

### Direcciones físicas

Las direcciones físicas (o direcciones MAC, del inglés *Medium Access Control*) están formadas por grupos de 12 dígitos hexadecimales, por ejemplo: 3A:02:1C:9B:7A:A9.

La tecnología Ethernet, que aparece en el estándar IEEE 802.3, fija un modelo de paquete que contempla, específicamente, una **dirección física** de origen y una de destino. Así, a pesar de enviar los paquetes por un cable en el que hay varias máquinas conectadas, el propio paquete indica cuál de estas máquinas es la receptora de la información. Y la receptora sabrá quién le ha enviado la información, por si acaso le tiene que contestar.

A mediados de los años noventa, el crecimiento de la capacidad de los equipos informáticos y de la información que procesaban impulsó un nuevo sistema Ethernet a más velocidad. Así pues, aparece el **Fast Ethernet** (literalmente, Ethernet rápida), a una velocidad de 100 Mbit/s.

En paralelo, se popularizan los conmutadores (en inglés, *switch*), que agilizan la transmisión de información mediante las redes Ethernet.

Los **conmutadores** son capaces de conocer qué máquinas hay conectadas en cada puerto. A diferencia del concentrador, el conmutador no extiende la información recibida, sino que la envía al puerto en el que se halla el ordenador que debe recibir la información. Esto permite que la red sea mucho más eficiente.

Actualmente, las velocidades que ofrece Ethernet superan los 1.000 Mbit/s, lo que da lugar a la **Gigabit Ethernet**. Ethernet es una tecnología que ha pasado a ser un estándar sobradamente usado. No sólo ha desbancado otros sistemas

de LAN, como las *Token Ring* (populares durante los años ochenta y noventa), sino que actualmente también está desbancando a tecnologías para redes MAN y WAN.

Si hace unos años la conexión a una LAN Ethernet se llevaba a cabo mediante un hardware de precio elevado, hoy en día todos los ordenadores personales, ya sean de sobremesa o portátiles, vienen equipados de serie con un conector para red Ethernet.

### 3.1.2. LAN inalámbrica

El objetivo de las LAN inalámbricas (en inglés *wireless LAN*) es dotar de conexión a una LAN a los ordenadores móviles, básicamente portátiles y teléfonos móviles avanzados. Estas conexiones inalámbricas se han popularizado en los entornos públicos y en los propios hogares.

Las LAN inalámbricas se caracterizan por ofrecer conexión a una LAN sin la necesidad de prever e instalar un cableado.

Si para conectarnos a una LAN cableada debemos conectarnos a un conmutador o un concentrador utilizando un cable, para conectarnos a una red inalámbrica debemos conectar con un **punto de acceso**. Este dispositivo permite la conexión a una red que se identifica con un **identificador de servicio**<sup>20</sup> (SSID). El alcance de esta red inalámbrica variará en función de dónde se encuentre ubicado el punto de acceso: en un espacio abierto será considerablemente mayor que si se encuentra dentro de un edificio con paredes de hormigón.

#### Wi-Fi

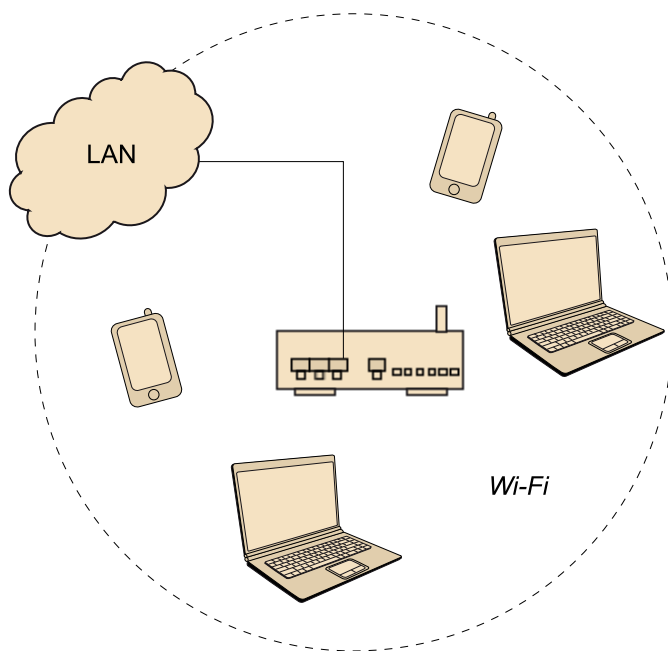
Las redes LAN sin hilos reciben popularmente el nombre Wi-Fi.

#### La banda ISM

Las redes inalámbricas que podemos tener en casa funcionan en la banda ISM (del inglés *Industrial, Scientific and Medical*). Se trata por una parte de frecuencias en las que se pueden usar aparatos sin la necesidad de pedir licencia (como sí que pasaría si quisiéramos poner un emisor de televisión).

<sup>(20)</sup> En inglés, *Service Set Identifier*.

### LAN sin hilos



La figura muestra dos portátiles y dos agendas digitales conectadas a una LAN mediante un punto de acceso.

Una de las principales funcionalidades del punto de acceso es conectarnos a una LAN cableada. Así pues, desde una misma red, tanto los equipos fijos como los móviles pueden compartir recursos.

Las LAN sin hilos se encuentran estandarizadas bajo el IEEE 802.11. Desde la aparición del estándar, ha habido una serie de variantes, entre las que destacamos:

- **IEEE 802.11a.** Utiliza la banda de los 5GHz y permite la transmisión a 54 Mbit/s. Como funciona en una banda de frecuencia diferente a la variante *b*, los dos estándares no son compatibles entre sí. Por ello, los fabricantes suelen ofrecer equipos que implementan las dos variantes.
- **IEEE 802.11b.** Utiliza la banda de los 2,4 GHz y permite la transmisión a 11 Mbit/s. La ventaja de este estándar respecto a la variante *a* es el aumento de la calidad de la señal. En la variante *a* se necesitan más puntos de acceso para tener la misma calidad de servicio, ya que las ondas en frecuencias altas se absorben más fácilmente por los obstáculos, por ejemplo, las paredes de una oficina.
- **IEEE 802.11g.** Es el estándar actual con más éxito. Funciona en la banda de los 2,4 GHz (por lo que es compatible con el 802.11b) y permite velocidades de 54 Mbit/s.

### Adaptación de velocidades

Las velocidades que hemos apuntado son velocidades máximas, ya que en función de la calidad de la señal que recibe el ordenador conectado a la LAN inalámbrica, la velocidad de transmisión se irá ajustando para hacerse resistente frente a los errores.

Actualmente, existen otras variantes, algunas en proceso de estandarización, para aumentar la velocidad de transmisión y la seguridad en las redes inalámbricas.

La seguridad en las LAN inalámbricas es algo importante. Si para conectarnos a una LAN Ethernet es necesario tener acceso físico al dispositivo que nos conecta, para conectarse a una LAN inalámbrica basta con estar dentro del alcance del punto de acceso. Así pues, se puede dar el caso de que un vecino se conecte a nuestro punto de acceso y, por ejemplo, se dedique a navegar por Internet utilizando nuestra línea ADSL (o peor: que se dedique a curiosear entre las carpetas compartidas de nuestra red, etc.).

Con el fin de proteger el acceso indebido a nuestra LAN inalámbrica, hay una serie de técnicas, entre las que destaca la protección con el **sistema de privacidad equivalente a la de una red cableada** (WEP<sup>21</sup>). Este sistema se basa en el conocimiento de una contraseña para tener permiso para conectarse a una LAN inalámbrica. Si no se conoce la contraseña, el punto de acceso nos permitirá la conexión a la red inalámbrica.

(<sup>21</sup>) En inglés, *Wired Equivalent Privacy*.

### 3.1.3. Redes de área local de gran alcance

Las LAN han sido históricamente, como su nombre indica, redes de alcance local, es decir, cubren una oficina o, como mucho, todo un edificio. Para la interconexión de diferentes LAN (es decir, la interconexión de varios edificios, estuvieran cerca o en diferentes ubicaciones geográficas) se han utilizado las tecnologías WAN.

#### Ved también

Recordad que hemos definido las tecnologías WAN en el subapartado 1.1.1 de este módulo.

Las tecnologías WAN permiten establecer conexiones a gran distancia. He ahí algunos ejemplos:

- La **Red Digital de Servicios Integrados** (XDSI) permite la conexión de dos redes mediante una conexión con una velocidad más bien baja (entre 64 Kbit/s y 2 Mbit/s). Funciona sobre una conexión telefónica.
- El **Modo de Transferencia Asíncrona**<sup>22</sup> (ATM) es una propuesta más reciente que la anterior (data de principios de los noventa) y permite la convivencia, dentro de una misma conexión, de diferentes tipos de información (por ejemplo, datos de Internet o videoconferencias). Las velocidades de ATM son bastante superiores y los 622 Mbit/s son una velocidad típica.

(<sup>22</sup>) En inglés, *Asynchronous Transfer Mode*.

Ethernet ha evolucionado en cuanto a la velocidad de transmisión (1 Gbit/s, 10 Gbit/s, etc.) y en cuanto a su alcance: hoy en día, las operadoras de comunicaciones ofrecen la posibilidad de interconectar redes remotas, separadas a kilómetros de distancia, utilizando Ethernet de alta velocidad.

Aunque las anteriores tecnologías permiten ofrecer la compartición de recursos remotos y la interconexión de redes, actualmente el panorama se ha revolucionado con las tecnologías **Ethernet de gran alcance**.

Esto permite a trabajadores situados en puntos geográficamente lejanos compartir recursos y comunicarse como si estuvieran dentro de un mismo edificio. La tecnología Ethernet está desbancando a la ATM, que, además de ser muy compleja de configurar y gestionar, implica la adquisición de equipos más caros que los conmutadores para Ethernet de gran velocidad.

Una tecnología relacionada con las LAN de gran alcance es la tecnología de **red privada virtual**<sup>(23)</sup> (VPN). Esta tecnología permite utilizar una conexión a Internet para conectar dos ordenadores remotos o LAN remotas. Por ejemplo, podemos conectar el ordenador de casa a la LAN de la empresa y trabajar con sus recursos como si estuviéramos conectados a nuestro puesto de trabajo: es decir, pueden acceder a los servidores, a los programas de gestión, a las carpetas compartidas, etc. Los teléfonos móviles avanzados también permiten estas conexiones.

<sup>(23)</sup>En inglés, *Virtual Private Network*.

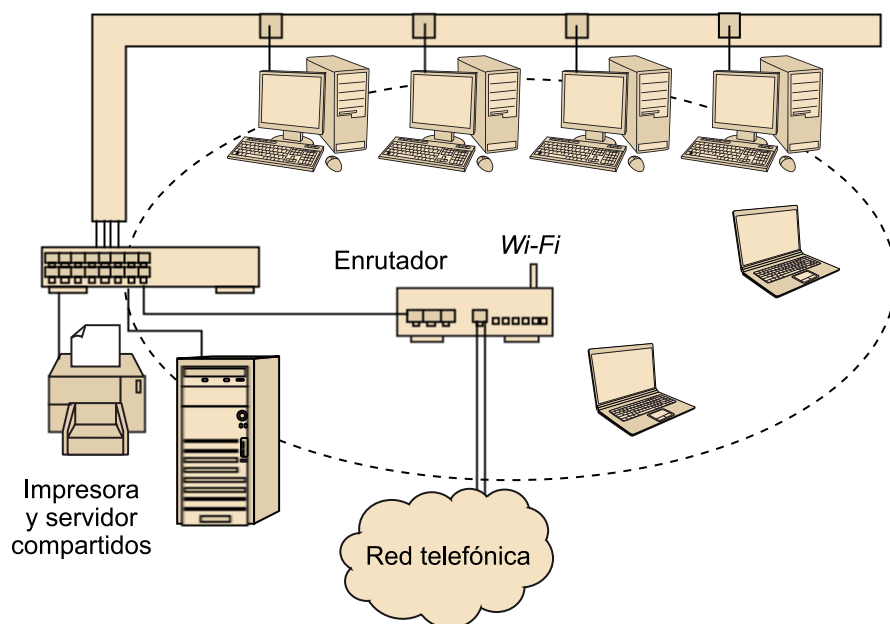
### 3.2. Ejemplos de LAN

Una vez que hemos visto las tecnologías que dan soporte a las LAN, veremos un par de ejemplos de redes que podemos encontrar a nuestro alrededor. Primero, veremos una red como la que podemos tener en una pequeña oficina y después, veremos un ejemplo de red de gran alcance.

#### 3.2.1. Una red para una pequeña oficina

En la figura siguiente, se muestra el esquema de una LAN para una pequeña oficina.

## Ejemplo LAN de una oficina



Las características de esta red son:

- Hay cuatro ordenadores de trabajo conectados a un conmutador Fast Ethernet. Cada ordenador dispone de un conector hembra RJ-45 que conecta con un cable trenzado de categoría 5 el ordenador con la roseta de la pared. A través de una canalización, el cable llega al conmutador.
- Hay un ordenador servidor que guarda los documentos de trabajo de los administrativos y administrativas. También está conectado a la red mediante el conmutador.
- Hay una impresora láser a color conectada a la red también por medio del conmutador Fast Ethernet.
- Los equipos comparten una conexión en Internet mediante un aparato que lo hace posible (*router*). Por una parte, este aparato está conectado al conmutador a través de un cable trenzado de categoría 5. Por otra, hay una conexión a la línea telefónica por medio de un cable de categoría 1 (sin trenzas) y un conector RJ-11.
- Este *router* incluye un punto de acceso que permite la conexión a la red inalámbrica. De esta manera, si se debe conectar algún ordenador más a la red, se puede utilizar esta conexión y no hay que prever un cableado nuevo.

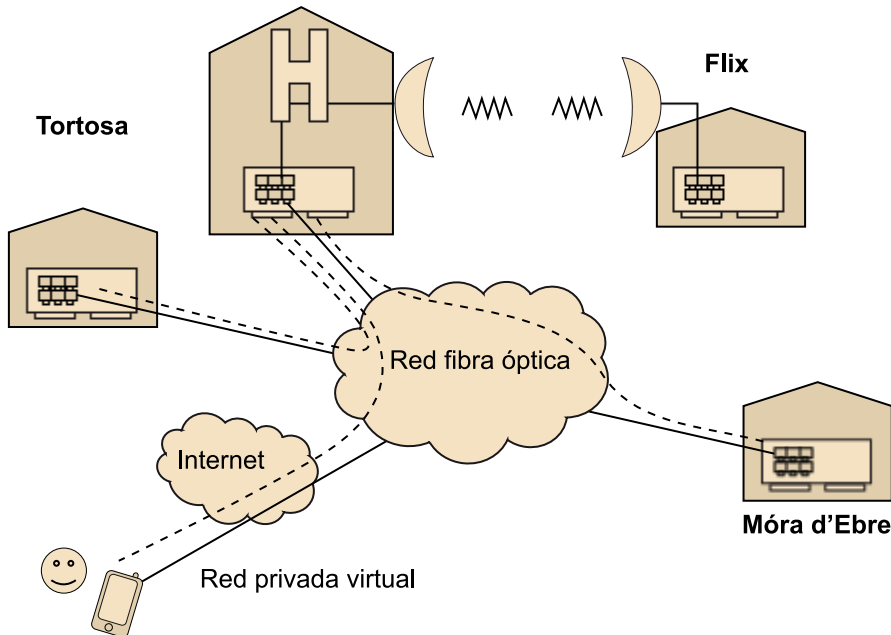
**Ved también**

Los *routers* los describiremos en el subapartado 4.2.

### 3.2.2. Una LAN corporativa

En este segundo ejemplo que se muestra en la figura siguiente, describimos la red para un hospital y tres centros de asistencia primaria (CAP). El hospital y un CAP se encuentran en Tortosa; los otros dos CAP están en Móra d'Ebre y en Flix.

Ejemplo de una LAN Corporativa



Vamos a ver cuáles son las características técnicas de las conexiones utilizadas:

- Los CAP de Tortosa y Móra d'Ebre se conectan al hospital mediante sendas conexiones Gigabit Ethernet. La operadora de telecomunicaciones les ofrece este servicio, que funciona gracias a la fibra óptica que tienen instalada por el territorio. La velocidad de conexión será, pues, de 1 Gbit/s.
- El CAP de Flix se conecta al hospital mediante un radio enlace que proporciona un ancho de banda de 500 Mbit/s.

Vamos a ver algunos de los servicios que se pueden desplegar en esta red. En cada centro existen diferentes tipos de trabajadores: médicos y enfermeras, administrativos, directivos y técnicos. Gracias al ancho de banda que interconecta los CAP con el hospital, los administrativos pueden trabajar contra un único servidor, ubicado en el hospital. De esta manera, pueden programar visitas y esta información estará centralizada en un único punto. Igualmente, los historiales médicos se encuentran digitalizados y ubicados en un único servidor central en el hospital. Desde cualquier CAP, los médicos pueden acceder a éste.



Finalmente, los médicos que hacen visitas domiciliarias pueden conectarse a la LAN desde Internet, utilizando una VPN.

### **VLAN**

A la hora de gestionar los centenares de equipos de una LAN corporativa se suele utilizar la tecnología VLAN (del inglés *Virtual Local Area Network*). Esta tecnología permite gestionar máquinas geográficamente distantes como si estuvieran conectadas a un mismo conmutador.

## 4. Internet

A finales de los años sesenta, aparecen los primeros intentos de interconexión de redes. Fue el caso de la interconexión de universidades americanas con centros de investigación o gubernamentales. El desarrollo de estas ideas ha dado lugar a lo que hoy conocemos como Internet.

Internet es la **red de computadores global** que conecta mundialmente multitud de sistemas de información. Internet nos permite comunicarnos, buscar información y, desde hace un tiempo, desarrollar todo tipo de actividades sociales.

Los sistemas implementados para la interconexión de redes han pasado a ser un estándar incluso en el mundo de las LAN. Así pues, dentro de un entorno de oficina puede haber un servicio Internet (como una página web) de acceso restringido para los trabajadores.

En este apartado, además de ver un pequeño resumen de la evolución de Internet, estudiaremos cómo funcionan las tecnologías que hacen posible la interconexión de redes.

### 4.1. Evolución de Internet

A continuación, para tomar conciencia de lo que ha supuesto Internet, vamos a ver un poco cuál ha sido su historia y evolución.

A finales de los años sesenta, la agencia norteamericana de proyectos avanzados de investigación (ARPA<sup>24</sup>) subvencionó a cuatro universidades para que implementaran un sistema de interconexión. Esto resultó mediante la red ARPANET, que conectó, el día 5 de diciembre de 1969, a las universidades de Stanford, Santa Barbara, Utah y California.

Internet empezó como un proyecto de investigación para interconectar sistemas y redes en un momento en el que no había estándares con respecto a conexiones y sistemas operativos.

#### Primera aparición de Internet

En el año 1974, Vinton Cerf titula un RFC "Specification of Internet Transmission Control Protocol"; es la primera vez que aparece la palabra *Internet*.

<sup>(24)</sup>En inglés, *Advanced Research Projects Agency*.

Durante los años setenta, esta red fue incrementando el número de máquinas conectadas y se convirtió en el centro de creación de los sistemas y tecnologías que han hecho de Internet una realidad.

Durante los mismos años, se fueron desarrollando sistemas de interconexión llamados X.25 para poder interconectar servidores de diferentes organismos. Sin embargo, los aficionados a la informática podían usar diferentes tipos de servicios (como las BBS), utilizando el ordenador personal y un módem conectado a la línea telefónica a velocidades que hoy consideramos muy lentas.

### Las BBS

Los sistemas de tablón de anuncios o sistemas de carteleros (del inglés *Bulletin Board Systems*) eran sistemas residentes en ordenadores servidores en los que uno se podía conectar mediante la línea telefónica. Había servicios mantenidos por organismos oficiales, medios de comunicación, etc., e incluso servicios mantenidos por aficionados a la informática. Desde estos servicios se podía acceder a noticias, comprar entradas e incluso descargar software.

Durante los años ochenta, por todo el mundo se empezaron a implantar sistemas de interconexión de redes basados en la propuesta de Vinton Cerf, considerado uno de los padres de Internet, gracias a que sistemas diferentes podían interconectarse de manera simple, usando todo tipo de conexiones físicas (conexiones telefónicas, radio, satélite, etc.).

Las operadoras de comunicaciones empezaron a ofrecer servicios de conexión a la flamante "red de redes", a la vez que algunas aplicaciones ganaban popularidad, como las páginas web con hiperenlaces creadas por Tim Berners Lee en el año 1989. A mediados de los años noventa, cuando Internet era utilizada básicamente por la comunidad científica y por los aficionados y profesionales de las tecnologías de la información, se produce el estallido tanto de servidores como de número de usuarios, hasta llegar al gran público en general.

Actualmente, Internet es un "componente" esencial en cualquier ordenador personal, no sólo en un entorno de trabajo, sino también en casa, para el ocio y la comunicación.

## 4.2. La interconexión de redes

La interconexión de redes que forma Internet se realiza mediante conexiones de gran ancho de banda, que funcionan sobre fibra óptica. El hecho de que los servicios ofrecidos en Internet sean accesibles desde todos los países implica que hay una conexión entre todos los países (es decir, todos los países están conectados a Internet).

Los servicios que ofrece Internet se basan, en general, en el modelo cliente-servidor que hemos visto anteriormente. Por lo tanto, hay servidores que ofrecen un servicio determinado (una página web, un gestor de correo electrónico, un servicio de vídeos, etc.) y clientes que se conectan a él.

### Ved también

La web se verá en el módulo "La World Wide Web".

### 4.2.1. Datagramas

La información que nos llega de Internet recorre un camino, más o menos largo, que va desde el servidor que aloja el servicio hasta nuestro ordenador. Este camino está formado por diferentes enlaces punto a punto.

La información que se transmite por Internet se divide en **datagramas**. Por ejemplo, cuando recibimos un mensaje de correo electrónico en el fondo estamos recibiendo los diferentes datagramas o porciones que lo conforman. A fin de que los datagramas sepan qué camino deben recorrer para ir del servidor al cliente o viceversa, hay dos elementos clave: las direcciones IP y los *routers*. Estos conceptos se sitúan en el nivel 3 (red) del modelo OSI.

#### Ved también

Recordad que hemos tratado el modelo OSI en el subapartado 1.3 de este módulo.

Un datagrama no se envía directamente sobre el medio, sino que siempre utilizará una tecnología de enlace subyacente. Por ejemplo, un datagrama se enviará entre dos máquinas conectadas a una misma LAN sobre una trama Ethernet.

### 4.2.2. Direcciones IP

El uso de datagramas y la identificación de redes con direcciones IP da lugar al protocolo IP.

Una **dirección IP** (del inglés *Internet Protocol*, protocolo de Internet) es un número que identifica una red de entre las muchas que forman parte de Internet. Una parte de este número llega a identificar un ordenador dentro de aquella red.

El formato de una dirección IP consta de 32 bits que se representan en cuatro grupos de 3 dígitos decimales separados por punto. Como los grupos son de 8 bits, los números decimales van del 0 al 255.

#### Versión actual del protocolo IP

La versión que se utiliza actualmente del protocolo IP es la versión 4, en la que se pueden identificar  $2^{32}$  ordenadores. Debido a la gran demanda de nuevas redes conectadas a Internet, se está implantando la versión 6 del protocolo, en la que las direcciones son de 128 bits. ¡En esta versión se podrán identificar  $2^{128}$  ordenadores!

#### Ejemplo de dirección IP

Un ejemplo de dirección IP es 85.32.69.117, que podría ser la dirección de un servicio Internet de correo electrónico. En este caso, el servidor se encuentra dentro de la red 85.32.69 y ese servidor en el que se ubica este servicio dentro de la red tendría el número 117.

Tras los números de las direcciones IP hay un grupo de implicaciones. A continuación, y sin ánimos de ser exhaustivos, veremos algunas de ellas:

- La asignación de direcciones IP en redes está centralizada por el ICANN. No puede ser que cada red tenga una dirección IP definida arbitrariamente sin tener en cuenta una perspectiva global.
- Las direcciones IP siguen un modelo jerárquico: hay direcciones reservadas para las redes muy grandes, para las redes medianas y para las redes pequeñas. Redes grandes hay pocas, mientras que de las pequeñas existe un número muy grande.
- Hay un número de direcciones reservadas para redes privadas, como las que podemos tener en casa. Un ejemplo son las redes que empiezan por 192.168.

**Ved también**

Recordad que hemos hablado del ICANN en el subapartado 1.2 de este módulo.

### 4.2.3. Routers

La interconexión de redes se produce gracias a los *routers*. Éstos se conocen más por su nombre en inglés que por su traducción al español: encaminador.

Un **router** es un dispositivo de red que procesa datagramas. En función de la dirección IP de destino que indica el datagrama, y de acuerdo con una **tabla de encaminamiento** definida en el *router*, el datagrama saldrá por una de las salidas del *router*.

Por lo tanto, un *router* unirá varias redes, cada una con dirección IP diferente.

Los *routers* son elementos que ya habíamos mencionado anteriormente, en concreto en el ejemplo de la red para una pequeña oficina. En aquel caso se trataba de un *router* ADSL. Su misión era enviar a Internet las peticiones de las máquinas de la LAN y, al mismo tiempo, enviar a las máquinas de la LAN las respuestas a sus peticiones cuando llegaban de Internet. Los datagramas viajan de una red (la red privada LAN) a la otra (Internet) gracias al *router*.

Si el datagrama tiene un destino desconocido para el *router*, éste reenviará el datagrama hacia un *router* que puede estar situado incluso en otro país.

#### ***Time to Live***

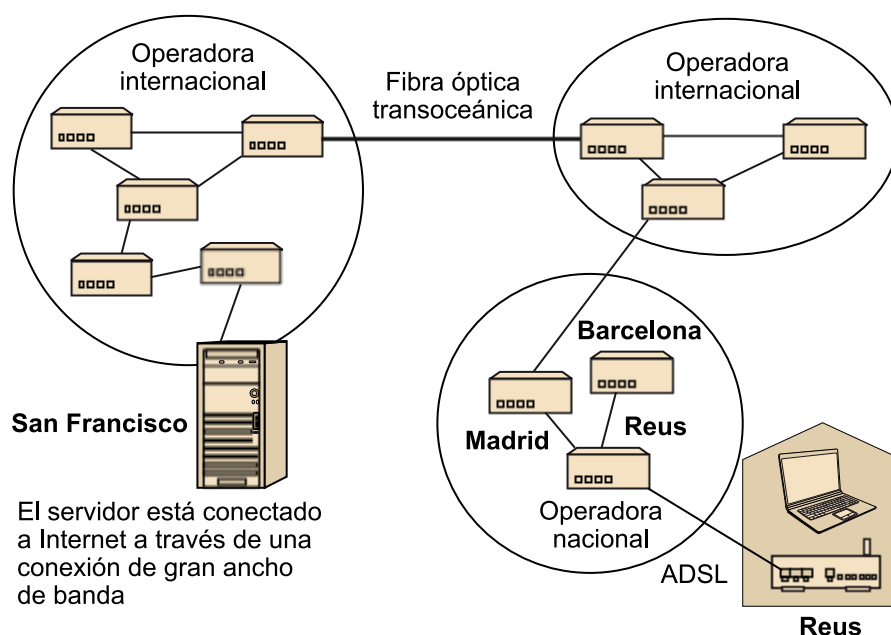
El *Time to Live* (en español, tiempo de vida) es una información que se incluye en el encabezamiento de los datagramas. Es un valor numérico, normalmente 256 o 128, que decrece cada vez que el datagrama pasa por un *router*. Cuando el valor llega a 0, se entiende que el datagrama tiene un destino inexistente y el *router* ya no lo reenviará. ¡De esta manera, se evita que datagramas con direcciones de destino erróneas estén dando vueltas por Internet eternamente!

Es evidente, pues, que los *routers* no conocen toda la estructura de Internet (es decir, la dirección IP de todas las redes que forman parte de ella). Es necesario, por lo tanto, una estructura jerárquica de Internet que explicaremos a continuación.

#### 4.2.4. Estructura de Internet

En este subapartado veremos, mediante un ejemplo, que se muestra en la figura siguiente, cómo se organiza Internet para permitir de una manera eficiente la comunicación desde nuestro ordenador cliente hasta un servidor ubicado en otro continente, por ejemplo de Reus a San Francisco. Esta comunicación se produce a partir de una serie de saltos entre redes.

Ejemplo de la estructura jerárquica de Internet



Un datagrama recorrerá un camino de red en red para ir de una IP de origen hacia una IP de destino. Cuando un datagrama sale de nuestro ordenador, va hacia la operadora de comunicaciones que nos da el acceso a Internet (llamada proveedor de servicios de Internet, ISP<sup>25</sup>). Para conectarnos a Internet, la operadora nos puede ofrecer varias tecnologías:

<sup>(25)</sup>En inglés, *Internet service provider*.

- La **conexión telefónica** ofrece velocidades bajas (como mucho 56 Kbit/s) y en modo analógico. Eso implica que la conexión es propensa a sufrir interferencias y errores. La conexión telefónica también puede funcionar sobre líneas telefónicas digitales XDSL.
- El **ADSL**, que ya hemos apuntado anteriormente, ofrece sobre la línea telefónica velocidades de bajada de información considerablemente elevadas (entre 1 y 20 Mbit/s). Podemos utilizar la línea para llamadas a la vez que utilizamos Internet.

- El **cable** es una tecnología que funciona en muchas áreas metropolitanas. Por el mismo cable llegan el teléfono, la televisión e Internet. Las velocidades proporcionadas son similares a las de ADSL.
- La **conexión inalámbrica**, habitual en entornos rurales, permite conexiones a Internet sin la necesidad de hacer llegar cableado a pueblos apartados de las grandes centrales telefónicas digitales. Las conexiones por medio de teléfonos móviles también formarían parte de este grupo.

### **El triple play**

El triple *play* se define como la terna de servicios que una operadora de comunicaciones puede ofrecer mediante una única conexión. Por ejemplo, usar la línea telefónica para llamadas, Internet (con ADSL) y televisión (televisión IP sobre ADSL).

Si estamos conectados a la operadora mediante un *router*, el datagrama se enviará desde el ordenador hacia éste utilizando la LAN (por lo tanto, se enviará sobre una trama). Una vez en el *router*, se utilizará la tecnología que nos proporcione la operadora, por ejemplo ADSL. Las diferentes tecnologías de nivel de enlace implican que el datagrama debe viajar sobre diferentes tipos de paquetes a lo largo de su recorrido.

Ya sea con ADSL, cable, llamada telefónica, etc., el datagrama llegará a la central de la operadora de comunicaciones. Una vez en Reus, otro *router* verá, gracias a la IP de destino, que el datagrama no está destinado a ninguna de las centrales del Estado. Por lo tanto, el *router* lo enviará hacia un nodo situado, por ejemplo, en Madrid.

En este nodo, un *router* verá que el datagrama va destinado a una red internacional. Por lo tanto, lo enviará hacia un nodo situado, por ejemplo, en París. Una vez en París, el *router* verá que la dirección IP del datagrama corresponde a una red ubicada en Estados Unidos, etc. Durante el tráfico por las diferentes operadoras, el datagrama viaja por conexiones de fibra óptica de muy alta velocidad.

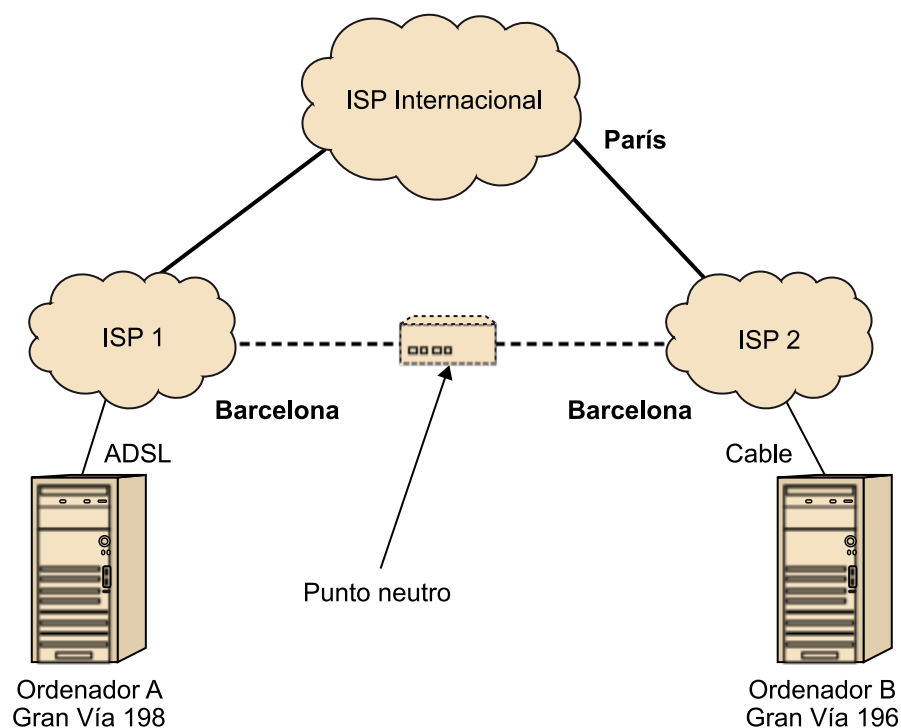
Con este ejemplo, y, refiriéndonos de nuevo a la figura anterior, vemos que hay una clara jerarquización de Internet. La operadora que nos conecta a Internet sólo cuenta con una conexión hacia el extranjero. La operadora situada en París podría tener nodos en los Estados Unidos, Japón y África. Es evidente que esta operadora es de un nivel jerárquico superior que la que nos da conexión mediante la ADSL.

Además, hay algunos conceptos un tanto más complejos que a continuación apuntaremos. En primer lugar, las operadoras de Internet de nivel bajo tienen más de una conexión hacia operadoras internacionales. Así, si una conexión cayera (por ejemplo, a causa de unas obras que cortaran la canalización de fibra), la información de la operadora tendría un camino alternativo para ir hacia operadoras internacionales.

Otro aspecto importante son los **puntos neutros**. Varias operadoras de nivel bajo están conectadas entre ellas mediante los puntos neutros, *routers* que permiten que el tráfico generado por ordenadores geográficamente próximos, pero con ISP diferentes, no deba conducirse siempre por medio de una operadora internacional.

En el ejemplo de la figura siguiente, si desaparece el punto neutro y el ordenador A, situado en el número 198 de la Gran Vía de Barcelona, se debe comunicar con el ordenador B, situado en el edificio de al lado, los datagramas deberán pasar por París. Sus proveedores de Internet son diferentes, con lo que si se deben intercambiar información, lo harán a nivel de un ISP internacional.

Punto neutro entre ISP próximos



Los sistemas estándar de encaminamiento que siguen los *routers* que hay en Internet provocan que estos últimos envíen los datagramas teniendo en cuenta una serie de parámetros (como el estado del tráfico de datagramas en las redes vecinas), y no sólo la dirección IP de destino.

Actualmente la red que da soporte a Internet es, pues, un embrollo de conexiones de fibra óptica (canalizaciones con centenares y miles de cables) interconectadas por *routers*. La simplicidad de los protocolos de encaminamiento que dan soporte, la amplia aceptación de IP como medio de transporte de aplicaciones telemáticas y el estallido del uso habitual y universal de Internet presagian una larga vida a esta manera de transmitir información.



### 4.3. El transporte de la información

Hemos visto cómo la información es capaz de ir de una punta a la otra de mundo. Sin embargo, a la hora de transportar la información hay un par de cuestiones que debemos tener presentes:

- Los datagramas ¿pueden recorrer caminos diferentes para ir de un mismo origen a un mismo destino, por ejemplo, de Reus a Japón?
- Una vez que un datagrama llega al ordenador, ¿cómo sabe si debe ir hacia el programa gestor de correo electrónico o hacia el programa de mensajería instantánea?

A lo largo de este subapartado iremos respondiendo a estas preguntas.

#### 4.3.1. El protocolo TCP

Los mensajes que se transmiten por medio de Internet pueden ser cortos (por ejemplo, un "Ok" enviado mediante un programa de mensajería instantánea) o largos (como una canción en formato MP3). En este último caso, el mensaje se deberá dividir en varios datagramas.

Dada la naturaleza de los protocolos de encaminamiento utilizados en Internet, se puede dar el caso de que los datagramas sigan **caminos diferentes** a pesar de que **origen y destino sean idénticos**, en función del momento. También puede suceder que algunos datagramas se **pierdan**. Por lo tanto, cuando se diseñaron los protocolos que darían lugar a Internet, se pensó en un sistema que controlara que la información llegue a destino completa y ordenada.

El **protocolo de transporte TCP**<sup>26</sup> es el encargado de dividir la información en datagramas y garantizar que el receptor de la información los reciba todos.

#### Protocolo TCP

El transporte de la información en Internet se fundamenta, pues, en los protocolos TCP y IP.

<sup>(26)</sup>En inglés, *Transmission Control Protocol*.

Esto se consigue mediante confirmaciones (como habíamos visto antes en este mismo módulo, para el caso del envío de tramas en conexiones punto a punto): el emisor envía un datagrama y, si en un tiempo determinado no ha recibido confirmación, lo reenviará de nuevo. En el destino, se **eliminarán los posibles datagramas repetidos** (que se habrían enviado más de una vez por problemas con la confirmación) y se **ordenará la información** antes de pasarla al programa que la necesita. Los protocolos TCP y UDP se sitúan en el nivel 4 (transporte) del OSI.

#### El UDP

El UDP<sup>27</sup> simplemente añade al datagrama la información sobre los puertos de envío. Por lo tanto, está pensado para servicios de Internet que se basan en envío de mensajes cortos. A diferencia del TCP, el UDP no garantiza que este mensaje corto se entregue;

<sup>(27)</sup>En inglés, *User Datagram Protocol*.

el UDP se suele utilizar para servicios tolerantes a la pérdida de datagramas, como la televisión por Internet.

### 4.3.2. Los puertos

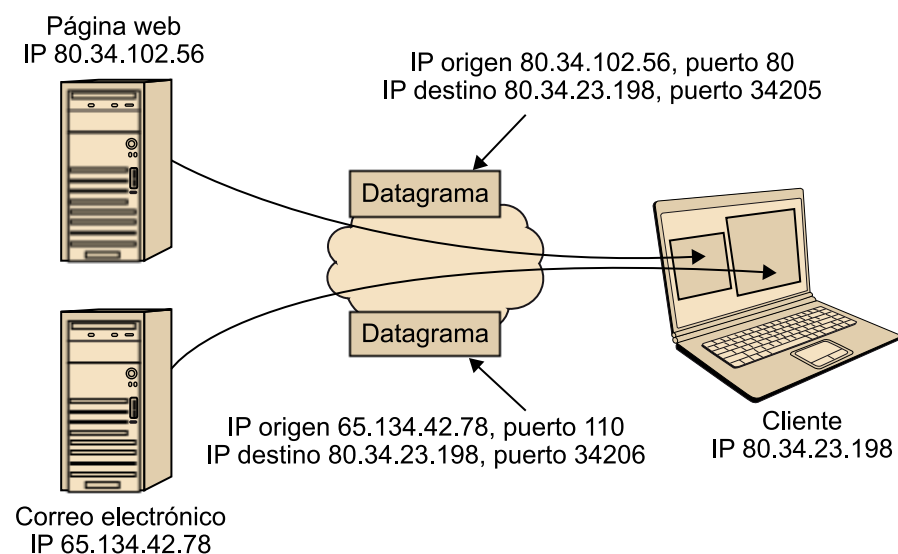
A fin de que el ordenador de destino sepa qué programa está esperando un determinado datagrama, se utilizan los **números de puerto**. Cada programa utilizará uno o varios números de puerto, y así el ordenador, en función del puerto de destino que contenga el paquete, sabe qué programa lo necesita.

El puerto es un número que va del 0 al 65.535. Los primeros 1.024 puertos están asignados a servicios concretos. Por ejemplo, el servicio web utiliza el puerto 80, el correo electrónico utiliza el 25 y el 110, etc. De entre los puertos restantes los hay que tienen su asignación más o menos oficializada, lo que resulta en que algunos puertos están asignados a diferentes servicios y aplicaciones. Según vemos en la figura siguiente, la máquina Cliente, con IP 80.34.23.198, está recibiendo información desde dos servicios ubicados en dos IP distintas. Los datagramas llegan al programa correcto, ya que el Cliente sabe por qué número de puerto destino está recibiendo cada uno.

#### Enlace recomendado

Podéis consultar la asignación de puertos a diferentes servicios de Internet en la web de Internet Assigned Numbers Authority (IANA).

Ejemplo del protocolo TCP



### 4.3.3. El empaquetado

Hasta ahora hemos hablado de tramas (en referencia a cómo se envía la información de un punto a otro punto) y de datagramas. Hemos visto que TCP se encarga de segmentar el mensaje que se debe enviar y crear varios datagramas. Un datagrama, por lo tanto, contiene la información que se ha de enviar de una red a otra. Hemos dicho que cuando se transmite el datagrama, se hará sobre una tecnología del nivel de enlace del OSI y, por lo tanto, lo hará empaquetado dentro de una trama. Así pues, un mensaje que se transmite usando el protocolo IP sobre una red Ethernet:

- Estará dividido en varios datagramas.

- Cada datagrama contendrá información TCP (por ejemplo, el puerto de destino).
- Cada datagrama contendrá información IP (como las direcciones IP origen y destino).
- Cada datagrama irá dentro de una trama Ethernet, que contendrá la información que hemos apuntado en el subapartado 2.2.

Al final del módulo veremos un ejemplo recopilando todos los conceptos.

#### 4.4. Los nombres de dominio

Hemos visto que para especificar el destino de la información utilizamos las direcciones IP. A los humanos nos resulta más fácil utilizar nombres que números a la hora de identificar cosas. Por lo tanto, para facilitar el uso de Internet, se diseñó un sistema para relacionar direcciones IP con nombres que podemos recordar fácilmente (nombres de dominio que identifican al servicio).

##### Enlace recomendado

Podéis consultar los dominios de primer nivel en la web de Internet Assigned Numbers Authority (IANA).

Con el objetivo de facilitar el uso de Internet, el **servicio de nombres de dominio** (en inglés, *Domain Name Service*, DNS) permite asociar nombres con direcciones IP.

A continuación, vamos a ver cómo funciona este sistema. Un nombre de dominio tiene una manera establecida: una serie de nombres separados por punto. Y también hay una jerarquía establecida: cuanto más a la derecha, más importancia tendrá el nombre en la jerarquía.

Por ejemplo, en el dominio `www.uoc.edu` el nombre con más importancia es `.edu`. El nombre de más a la derecha se llama **dominio de primer nivel**. Los dominios de primer nivel están regulados internacionalmente y algunos de ellos están reservados para usos determinados.

##### Ejemplos de dominios de primer nivel

Algunos ejemplos de dominios de primer nivel son:

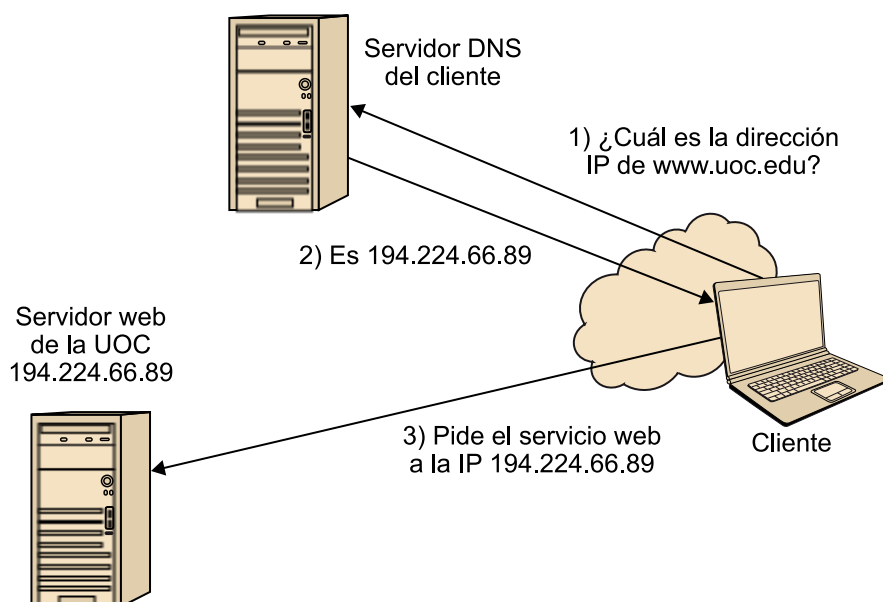
- Dominios geográficos, que identifican países. Por ejemplo `.es` (España) `.fr` (Francia), `.uk` (Reino Unido), etc.
- Otros dominios como `.edu` (para organismos educacionales), `.com` (para servicios comerciales), `.museum` (para museos), etc.

De cada dominio de primer nivel, existe una **entidad gestora del dominio** que asignará los nombres de este dominio de primer nivel. Así pues, si queremos registrar un nombre de dominio para un servicio de Internet, necesitaremos dirigirnos a la entidad en cuestión. Volviendo al ejemplo de `www.uoc.edu`, la UOC tuvo que registrar el nombre `uoc` en la entidad que gestiona los dominios de alto nivel `.edu`. Paralelamente, la UOC se convierte en entidad gestora

del dominio `uoc.edu` y, en consecuencia, puede definir varios nombres dentro de su nivel, por ejemplo, `www.uoc.edu`, `correo.uoc.edu`, `cv.uoc.edu`, `materiales.uoc.edu`, etc.

El funcionamiento del DNS en los ordenadores es muy sencillo y lo encontramos ilustrado en la figura siguiente. Cuando queremos acceder a un servicio web, por ejemplo `www.uoc.edu`, nuestro navegador realiza una consulta a un servicio DNS (mensaje 1 de la figura). Cada ordenador conectado a Internet necesita saber cuál es su servidor DNS, es decir, a qué dirección IP debe consultar para saber la correspondencia entre nombre de dominio y dirección IP. Esto se hace de manera totalmente transparente a los usuarios, de modo que no somos conscientes de que se están realizando estas consultas. Una vez recibamos la respuesta desde el servicio DNS (mensaje 2 de la figura), el ordenador Cliente lleva a cabo una petición al servicio web y abre una conexión en la IP que le ha indicado el servicio DNS (mensaje 3 de la figura).

Funcionamiento del DNS



El procedimiento para llevar a cabo los servicios DNS para conocer todas las correspondencias entre nombres de dominio y direcciones IP es algo complejo. Por lo tanto, está fuera del alcance de estos materiales. En esencia, sin embargo, diremos que la estructura en niveles de los nombres de dominio lo hace posible de una manera eficiente.

## 5. El correo electrónico

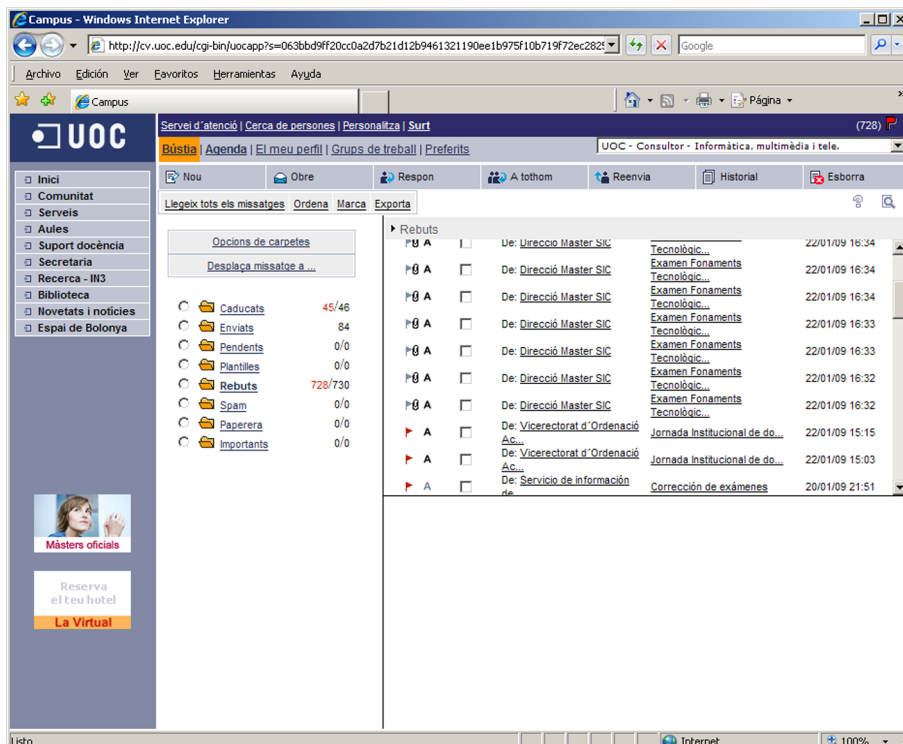
En este apartado, daremos un ejemplo de todo lo que hemos visto anteriormente mediante el estudio de uno de los servicios de Internet más populares: el correo electrónico o *e-mail*.

El objetivo del correo electrónico es el envío y recepción de mensajes de información textual que, adicionalmente, pueden incluir documentos e información multimedia.

### El SPAM

Gran parte de los mensajes de correo electrónico que se mueven por Internet son SPAM o correo basura. Dentro de este concepto, hay diferentes tipos de mensaje, desde anuncios hasta mensajes falsos. El uso del nombre de esta marca de carne enlatada para designar los molestos correos viene de un *sketch* de los Monty Python.

La gestión del correo electrónico puede hacerse o bien desde software instalado en el ordenador (como el Mozilla Thunderbird), o bien desde páginas web, como es el caso del correo web de la UOC.



Gestor de correo electrónico del campus de la UOC

En ambos casos, la aplicación permite la redacción, la lectura y la organización de los mensajes en carpetas, así como la gestión de direcciones de correo.

Sea por el medio de gestión que sea, los protocolos sobre los que funciona el correo electrónico son los mismos. Estos protocolos (y todos los que se utilizan en Internet) son usados por todo el mundo, de manera que es posible enviar correos electrónicos a destinatarios de cualquier país.

En primer lugar, veremos las bases de este servicio y después pasaremos a detallar cómo se envía un correo electrónico mediante Internet.

### 5.1. Direcciones y mensajes

El correo electrónico permite el envío de un mensaje a uno o varios destinatarios. Estos destinatarios están identificados por una dirección de correo con el formato siguiente:

usuario@dominio

#### Ejemplo de dirección de correo

mfernandez@uoc.edu hace referencia a un usuario determinado del servicio de correo electrónico del dominio uoc.edu, atribuible, pues, a la Universitat Oberta de Catalunya. Notad que podría haber una dirección mfernandez@webnetworks.net que no hiciera referencia a la misma persona.

El mensaje de correo electrónico es un **mensaje de texto**, formado por caracteres ASCII. A continuación, ponemos un ejemplo de correo electrónico, que comentaremos a continuación (se han omitido informaciones poco relevantes y se han numerado las líneas para facilitar su comentario):

#### Usuarios de correo electrónico

Dentro de un mismo dominio, no puede haber usuarios de correo electrónico con el mismo nombre.

#### Ved también

Recordad que hemos visto el código ASCII en el módulo "Aspectos tecnológicos de los sistemas informáticos".

```
1: Disposition-Notification-To: amartinezball@uoc.edu
2: Date: Tue, 16 Dec 2008 12:41:00 +0100
3: From: amartinezball@uoc.edu
4: Organization: Universitat Oberta de Catalunya
5: User-Agent: Mozilla Thunderbird 1.5.0.8 (Windows/20061025)
6: MIME-Version: 1.0
7: To: mfernandez@webnetworks.net
8: Subject: Prueba
9: X-Priority: 1 (Highest)
10: Content-Type: multipart/mixed; boundary="--070600070907"

11: This is a multipart message in MIME format.
12: --070600070907
13: Content-Type: text/plain;
14: Content-Transfer-Encoding: 8bit

15: ¿Qué tal? Éste es un mensaje de prueba.

16: --070600070907
17: Content-Type: application/pdf;
18: Content-Transfer-Encoding: base64
```

```
19: filename="documento.pdf"
20: OQo+PgpzdHJlYW0KeJy1WMty2kgU3fMVd+e4yhHdeiFmR4B4qP ...
```

Un mensaje de correo electrónico consta de diferentes partes: el **encabezamiento**, el **cuerpo** y los **elementos adjuntos** al correo electrónico.

El encabezamiento del mensaje del código anterior está formado por las siguientes líneas:

- Una línea para especificar que se pedirá al receptor que envíe una confirmación de lectura (línea 1, *Disposition-Notification-To*)
- La fecha de envío del mensaje (línea 2, *Date*)
- El destinatario (línea 7, *To*) y el remitente (línea 3, *From*) del mensaje. Un mensaje puede tener más de un destinatario. Más adelante haremos referencia a lo que esto implica.
- El programa que se utiliza para enviar el correo electrónico (línea 5, *User-Agent*) En este caso el *Mozilla Thunderbird* bajo el sistema operativo Windows.
- La prioridad del mensaje (línea 9, *X-Priority*). Esta prioridad máxima (con el valor 1 *Highest*) se deberá hacer notar al receptor del mensaje; para ello, sencillamente, se marcará de un color intenso o tendrá un icono con una exclamación, etc. Una prioridad elevada no implica que el mensaje se envíe más rápido que los otros.
- El asunto del mensaje (línea 8, *Subject*). Éste contiene una palabra o pequeña frase elegidas por quien escribe el mensaje, para dar una idea de cuál es su contenido.

El mensaje de ejemplo contiene un adjunto, un fichero PDF llamado `documento.pdf`. Vamos a ver cómo se gestiona la adjunción de ficheros con el correo electrónico.

Los mensajes fueron diseñados para comunicar tan sólo texto. Cuando surgió la necesidad de enviar ficheros como imágenes o documentos anexos, se diseñó un sistema llamado Extensiones Multipropósito para el Correo de Internet<sup>28</sup> (MIME). Si revisamos el ejemplo, entenderemos cómo funciona el sistema MIME:

<sup>(28)</sup>En inglés, *Multipurpose Internet Mail Extensions*.

- La línea 10 indica que el mensaje es **multiparte**, es decir, que está formado por varias partes (en este caso, el cuerpo y un fichero adjunto). Se define una **frontera**<sup>29</sup> que delimitará las diferentes partes de las que consta un mensaje. Se trata de un contenido de texto, los caracteres de frontera, que se elige de manera que no aparezca dentro del contenido del propio adjunto, ya que si no, se encontraría una frontera en medio de la información del adjunto y se produciría un error. La línea 11 indica de nuevo que el mensaje es multiparte.

<sup>(29)</sup>En inglés, *boundary*.

- La línea 12 contiene la frontera que indica que empieza una de las partes del mensaje. La codificación es de 8 bits por carácter.
- La línea 16 marca el inicio de un nuevo contenido, en este caso, un fichero PDF que lleva por nombre `documento.pdf`. El fichero está codificado en base64, que es un sistema que permite representar los bytes que forman este fichero mediante caracteres de texto (recordad que son los únicos utilizables en el correo electrónico).

## 5.2. Envío de correos

Una vez que hemos visto cuál es la estructura de la información que se envía en un correo electrónico, vamos a ver las bases del envío de correo.

Siguiendo con el ejemplo, el usuario `amartinezball@uoc.edu` quiere enviar un mensaje a `mfernandez@webnetworks.net`.

Para enviar correos electrónicos se utiliza un servicio llamado Protocolo Simple de Transferencia de Correo<sup>30</sup> (SMTP) que está en el nivel 7 (de aplicación) del OSI. Veamos cómo funcionaría con el ejemplo:

<sup>(30)</sup>En inglés, *Simple Mail Transfer Protocol*.

- Hay dos servidores SMTP que toman parte de la transmisión del correo: el servidor ubicado en el dominio `uoc.edu` y el servidor ubicado en `webnetworks.net`.
- El software gestor del remitente del mensaje contacta con el servidor SMTP del dominio `uoc.edu` a través de la IP correspondiente a este servidor. Con una serie de mensajes, siguiendo el protocolo SMTP, el correo se transferirá desde el ordenador en el que se ejecuta el software gestor hasta el servidor de correo de la UOC.
- Una vez el mensaje esté allí, el servicio SMTP de la UOC contactará con el servicio SMTP de Webnetworks. Primero hará una consulta al servicio DNS para saber en qué dirección IP se encuentra ubicado el dominio `webnetworks.net`. Si no existiera este servicio, el servidor enviaría un mensaje de correo electrónico a `amartinezball@uoc.edu`, notificando el error.
- Si la conexión se acepta por parte de Webnetworks, el servicio SMTP de la UOC transferirá el mensaje hacia su homólogo.
- El servicio SMTP de Webnetworks guardará el mensaje en su disco, dentro de una carpeta para el usuario `mfernandez`, de manera que esté disponible para cuando el usuario quiera mirar si tiene correo.



Ya hemos visto que el envío del correo electrónico es bastante sencillo. Como complemento, podemos ver qué sucede en una serie de escenarios concretos.

- Por ejemplo, podría pasar que el usuario `amartinezball@uoc.edu` envíe un mensaje al usuario `jcastell@uoc.edu`. En este caso, al pertenecer los dos usuarios al mismo dominio, el servidor ubicado en la UOC guardará directamente el mensaje en la carpeta del disco destinada al usuario `jcastell`.
- Si el mensaje se envía a múltiples destinatarios pertenecientes al dominio UOC, el servidor SMTP de la UOC hará las diferentes copias.
- Si el mensaje se envía a múltiples destinatarios, cada uno perteneciente a dominios diferentes, el servidor SMTP hará las conexiones con los servicios SMTP de los distintos dominios a los que pertenezcan los destinatarios.

El envío de mensajes mediante SMTP se realiza a través de un protocolo que consiste en una serie de mensajes estándar. Estos mensajes permiten establecer un diálogo SMTP entre el emisor y el receptor para poder enviar los correos. Para consultar el correo electrónico, se utiliza otro protocolo que también define un diálogo. Profundizaremos en ello a continuación.

### 5.3. Lectura de un correo electrónico

Con el fin de ejemplarizar cómo se relacionan algunos de los conceptos vistos en este módulo, veremos cómo se produce la lectura de un correo electrónico por parte del usuario. Supondremos que el usuario `mfernandez` utiliza un software gestor de correo electrónico (y no un servicio basado en la web).

Este software utilizará el protocolo oficina de correos (versión) 3<sup>31</sup> (POP3) para facilitar al usuario la lectura de mensajes que tiene almacenados en el servidor de correo del dominio Webnetworks. Este protocolo también está en el nivel de aplicación del OSI.

<sup>(31)</sup>En inglés, *Post Office 3*.

A continuación, damos un ejemplo del funcionamiento de POP3 a fin de que el usuario `mfernandez` lea el correo que le ha enviado `amartinezball`:

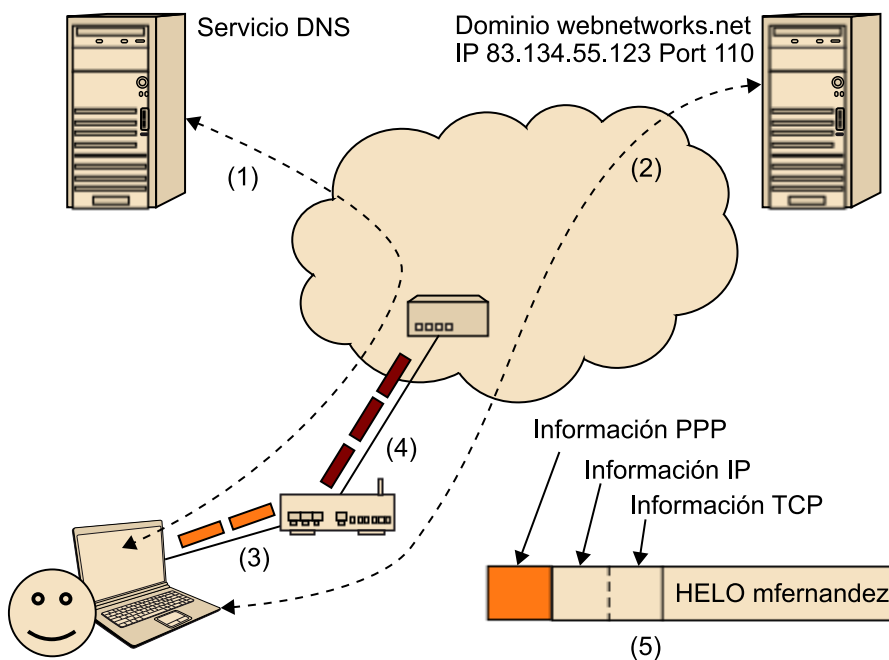
- El usuario `mfernandez` establecerá una conexión TCP en el puerto 110 de su servidor de correo electrónico. Iniciará un diálogo siguiendo el protocolo POP3: enviará el mensaje `user mfernandez` al servidor POP3 del dominio Webnetworks y el POP3 le contestará `OK`.
- Una vez que el usuario se ha identificado, es necesario que éste se valide con una contraseña. Así pues, el usuario envía el mensaje `pass secret`. El POP3 le dirá `OK` si la contraseña es correcta. En caso contrario, le dará un mensaje de error.

- El usuario envía el mensaje `list` para saber cuántos mensajes hay. El POP3 le contestará, por ejemplo, `1 732, 2 514`, indicando que hay dos mensajes: el primero de 732 bytes de tamaño y el segundo, de 514.
- Si el usuario quiere descargar el mensaje 1, enviará al servicio POP3 el mensaje `retr 1`. La respuesta será el propio mensaje de correo, es decir, un texto con el formato que hemos visto en la sección anterior.
- Si el usuario quiere borrar el mensaje del servidor, ya que ya lo ha leído, enviará al servicio POP3 el mensaje `dele 1`.

Normalmente, el software gestor de correo irá pidiendo todos los mensajes al servicio POP3. Por defecto, este software borrará del servidor los mensajes que ya se ha descargado, aunque eso es opcional (si no se quisiera, después del `retr` no habría un `dele`).

Según lo que hemos visto durante este módulo, completamos el ejemplo con la figura siguiente, en la que supondremos que el usuario se conecta a Internet mediante una LAN Ethernet y un *router* ADSL.

Lectura del correo electrónico mediante el protocolo POP3



En la figura anterior, se puede ver lo siguiente:

- 1) Para saber la dirección IP en la que está ubicado el servidor de correo, se hará previamente una consulta DNS sobre el nombre de dominio `webnetworks.net`.

2) El POP3 enviará sus mensajes sobre una conexión TCP que se establecerá entre un puerto de la máquina de origen y el puerto 110 de la máquina de destino ubicada en la IP indicada por el servicio DNS.

3) En general, la conexión TCP dividirá los mensajes que envíe POP3 en varios datagramas: esto será en caso de enviar mensajes grandes, que necesitan diferentes datagramas. Éstos se enviarán mediante diferentes tecnologías de enlace implicando, por lo tanto, diferentes tipos de paquete. Por ejemplo, desde el ordenador en el que se lee el correo hasta el *router* se utilizarán paquetes Ethernet.

4) Desde el *router* hasta la central de la operadora de comunicaciones, se usará el protocolo PPP sobre el medio ADSL en la línea telefónica. Dentro de la red de la operadora se utilizará la fibra óptica, etc.

5) Como hemos visto en el subapartado 4.3 de este módulo, cada paquete empezará con la información de nivel de enlace (por ejemplo, el indicador de inicio de trama, la longitud de la trama, etc.). Dentro contendrá el datagrama y, por lo tanto, también se añadirá información sobre el protocolo IP, como la dirección IP de destino. El datagrama transportará información del TCP, por ejemplo, el puerto de destino. Finalmente, aparecerá el mensaje (o trozo del mensaje en el caso de que éste se haya tenido que fragmentar) correspondiente al POP3.

Finalmente, podemos relacionar los protocolos que han aparecido en este ejemplo con los siete niveles OSI: el protocolo POP3 lo ubicamos en el nivel de aplicación. El TCP lo situamos al nivel de transporte y el IP, al nivel de red. Los protocolos PPP y Ethernet corresponden al nivel de enlace, y conceptos como el ADSL y los protocolos de la red de la operadora que funcionan sobre fibra óptica los situaríamos a caballo entre el nivel físico y el nivel de enlace.

#### Protocolo punto a punto

Los *Point-to-Point Protocol* (PPP) son un sistema de envío de paquetes muy utilizado en conexiones precisamente punto a punto.

## Resumen

En este módulo, hemos estudiado los aspectos tecnológicos que fundamentan las redes de computadores. También hemos analizado cuál es el funcionamiento de Internet y hemos visto un ejemplo concreto de servicio de Internet: el correo electrónico.

En primer lugar, hemos estudiado los diferentes tipos de redes de computadores, según su tamaño y la manera en la que se transmite la información. También hemos presentado a los principales organismos que están relacionados con la estandarización de Internet y las tecnologías de las redes.

En segundo lugar, hemos visto cómo se envía la información, es decir, ceros y unos, a través de un medio físico. Hemos descrito cómo se envía un paquete entre dos ordenadores en una conexión punto a punto y hemos descrito cómo varias transmisiones de información pueden compartir el medio.

Hemos descrito la funcionalidad de las redes de área local, tanto las basadas en tecnología Ethernet como las redes inalámbricas. También hemos apuntado la tendencia de hacer grandes redes corporativas mediante tecnologías de redes de gran alcance. Hemos puesto varios ejemplos para ilustrar su uso.

Hemos dedicado buena parte del módulo a hablar de Internet. Hemos visto que la información se transmite mediante varias redes, cada una de ellas utilizando su propia tecnología de enlace. La información se transmite en datagramas, que son gestionados por el protocolo TCP. Gracias a las direcciones IP y a los *routers*, los datagramas llegan a su destino. Hemos visto que Internet está formado por miles de enlaces punto a punto organizados de manera jerárquica, para conseguir de manera sencilla que se pueda transmitir información por todo el planeta. Finalmente, hemos visto que gracias a los nombres de dominio podemos utilizar los servicios de Internet sin la necesidad de recordar las direcciones IP en la que se ubican.

Finalmente, hemos estudiado el servicio de correo electrónico, lo que nos ha servido para revisar algunos de los conceptos vistos a lo largo del módulo. En concreto, hemos visto cómo es un correo electrónico, cómo se lleva a cabo su envío y cómo se lee desde el software gestor.

## Actividades

1. Averiguad cuáles son las direcciones IP y MAC que tiene vuestro ordenador. Por ejemplo, si estáis en un sistema Windows, podéis abrir un terminal (por medio de Inicia/Ejecuta y la introducción de `cmd`). Una vez dentro del terminal, ejecutad `ipconfig /all`.
2. Si tenéis una pequeña red de computadores en casa o en el trabajo, dibujad un pequeño esquema que la represente. Identificad los elementos que hemos estudiado en este módulo (*router*, punto de acceso, RJ-45, etc.).
3. En la web de la Internet Assigned Numbers Authority (IANA) encontraréis información sobre los finales de los nombres de dominio (.se, .edu, etc.). Buscad entre la información que contiene y decid a qué corresponde el dominio "tv". Dad el nombre a tres dominios "esponsorizados" y decid a qué hacen referencia.
4. ¿Qué sistema de conexión a Internet utilizáis? Id a la web de Internautas.org y, utilizando sus herramientas, decid vuestra velocidad de subida y de bajada. Indicad también cuál es vuestra IP según este servicio. En principio, el valor debería ser diferente: la máquina desde la que os conectáis tiene una dirección IP privada. La dirección que os da el servicio se corresponde con vuestro *router*, y es la dirección que identifica vuestra red.

## Ejercicios de autoevaluación

1. La empresa en la que trabajáis dispone de tres naves separadas por un centenar de metros. Resulta que todos trabajáis con un servidor ubicado en el edificio principal y que, en los tres edificios, hay una LAN. Justificad cuál de las tecnologías estudiadas en el módulo sería más adecuada para unir las tres LAN en una red corporativa que permita trabajar con el servidor.
2. Diferenciad el par trenzado de la fibra óptica como medio físico de transmisión de información.
3. Con el ADSL, ¿cómo es posible que se transmita al mismo tiempo y por un solo cable la conversación telefónica, la información que llega de Internet y la información que va a Internet?
4. En una pequeña LAN conectada a Internet hemos visto que un *router* ADSL, ¿cómo se puede compartir la conexión a Internet entre los ordenadores conectados a la LAN?
5. En casa tenéis un despacho en el que disponéis de un ordenador y una conexión a Internet con ADSL. Decid qué haríais si queréis poner un ordenador en otra habitación y un disco duro multimedia en la sala de estar y conectarlo todo a la LAN.
6. Una fotografía JPEG de 3.096 bytes se descarga de Internet. Suponiendo que estamos conectados a Internet con una red Ethernet, y que el tamaño máximo de las tramas es de 1.500 bytes, decid cuántas tramas serán necesarias para llevar esta fotografía y dibujadlas. Tened presente que la información suplementaria que añade Ethernet son 46 bytes, la que añade IP son 20 bytes y la que añade TCP son 24 bytes.
7. Explicad cómo se consigue que un datagrama que entra en vuestro ordenador sepa si debe ir dirigido hacia el programa gestor de correo electrónico o hacia el navegador web.
8. Describid esquemáticamente cómo sería un mensaje de correo electrónico con dos ficheros adjuntos: una fotografía y un documento de Word.

## Solucionario

### Ejercicios de autoevaluación

1. Utilizaríamos la tecnología Ethernet, en su variante Gigabit Ethernet, que proporciona 1.000 Mbit/s, ya que queremos trabajar con un servidor central sobre el que trabajará mucha gente. Las otras tecnologías que se han visto en el módulo no serían tan válidas. Por ejemplo, las redes inalámbricas no darían bastante ancho de banda; hemos visto que la ATM es demasiado compleja y empieza a estar obsoleta; la XDSI implicaría tener un contrato con la operadora de comunicaciones; el ADSL implicaría tener un contrato con la operadora de comunicaciones. Las tecnologías XDSI y ADSL no nos darían bastante ancho de banda para poder trabajar cómodamente contra el servidor.

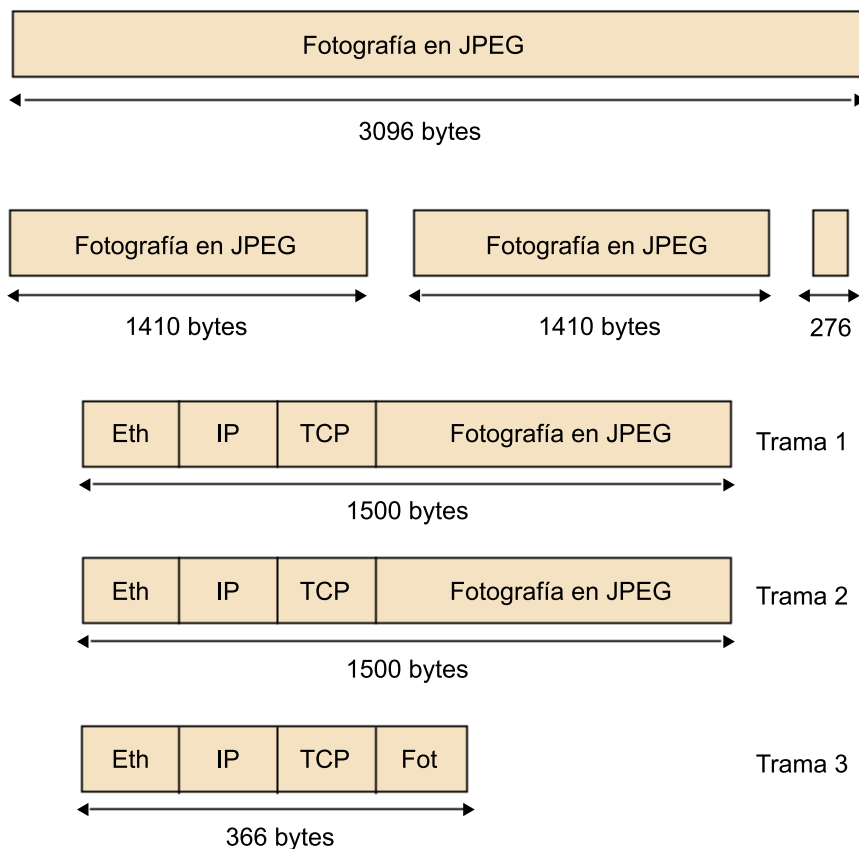
2. Podéis encontrar la respuesta en el subapartado 2.1. de este mismo módulo.

3. Como hemos visto en el módulo, en el subapartado 2.3, el ADSL utiliza un acceso al medio compartido (el cable telefónico), utilizando la multiplexación por división en frecuencia. Por lo tanto, cada tipo de comunicación ocupará una frecuencia diferente y uno no invadirá al otro.

4. El *router* ADSL recibirá datagramas que vendrán de los diferentes ordenadores conectados a la LAN. Los datagramas se irán enviando uno tras otro por la salida ADSL del *router*. Así, diferentes comunicaciones podrán ser enviadas hacia Internet sobre un mismo canal.

5. Una opción sería conectar por medio de cable de categoría 5 el nuevo ordenador y el disco duro multimedia. Esto implicaría pasar el cable por dentro de casa, lo que no siempre es factible. Una solución viable sería pedir un *router* habilitado con un punto de acceso inalámbrico y montar un SSID para conectar los nuevos equipos a la LAN.

6. Cada trama contendrá: 46 bytes de información Ethernet, 20 bytes de información IP –conteniendo, por ejemplo, las direcciones IP de origen y de destino– y 24 bytes de información TCP –conteniendo, entre otros, el puerto destino–. Así pues, si el tamaño máximo de trama es 1.500 bytes, sobrarán  $1.500 - 90 = 1.410$  para poner datos del JPEG. Por lo tanto, necesitaremos  $3.096 / 1.410 = 2,1 = 3$  tramas. Las dibujamos a continuación:



7. Se consigue gracias a que a pesar de tener la misma IP de destino, cada datagrama tiene un número de puerto diferente en función de si debe ir al navegador web o al gestor de correo electrónico.

8. Antes que nada, tendría un encabezamiento con información como el remitente, el destinatario, la fecha de envío o el asunto. También se especificaría que el mensaje es multiparte, a la vez que se definiría una frontera. Esta frontera aparecería para separar el texto del correo del primer adjunto. Volvería a aparecer para separar el primer adjunto del segundo.

## Glosario

**encabezamiento** *m* Porción del paquete destinada a que el protocolo disponga distinta información. Por ejemplo, en el caso del protocolo IP, los datagramas tendrán en el encabezamiento datos como la dirección IP de origen y de destino o el tiempo de vida.

**conmutador** *m* Dispositivo de red utilizado para conectar ordenadores con una topología física de estrella y formar una red de área local. Tiene varios puertos de conexión para conectar cableado de par trenzado con conectores RJ-45.

**datagrama** *m* Paquete que es transmitido por el protocolo IP. Es enviado por diferentes enlaces para ir de una red de origen a otra de destino. El tráfico que circula por los enlaces que forman Internet está constituido por datagramas.

**router** *m* Dispositivo de red que procesa los datagramas que recibe para enviarlos al enlace de salida más adecuada en función de la dirección IP de destino que incluye el datagrama.

**LAN** *f* Red de área local, caracterizada por la velocidad de conexión elevada y la compartición de recursos como uno de los principales objetivos.

**nombre de dominio** *m* Palabra o grupo de palabras que permiten identificar servicios de Internet mediante un nombre y no de una dirección IP.

**PAN** *f* Red de área personal, formada por dispositivos separados por pocos metros.

**par trenzado** *m* Tipo de cable formado por parejas de cables de cobre trenzadas. Utilizado en telefonía y redes de área local.

**trama** *f* Paquete que es transmitido por los protocolos de nivel 2 del OSI, como Ethernet o PPP.

**WAN** *f* Red de área extendida o de largo alcance, que cubre gran parte de un territorio estatal o nacional, como la red telefónica.



## Bibliografía

**Barceló Odinas, J. M. et al.** (2008). *Aplicaciones y protocolos de Internet*. Barcelona: Editorial UOC.

**Kurose, J.** (2004). *Redes de computadoras: un enfoque descendente basado en Internet* (2.<sup>a</sup> ed.). Pearson Educación.

**Rabago, J. F.** (2008). *Redes locales, guía práctica*. Anaya multimedia interactiva.

**Tanenbaum, A.** (2003). *Redes de computadores* (4.<sup>a</sup> ed.). Prentice Hall.



# Seguridad en la sociedad de la información

Antoni Martínez Ballesté

PID\_00150268



Universitat Oberta  
de Catalunya

[www.uoc.edu](http://www.uoc.edu)



Los textos e imágenes publicados en esta obra están sujetos –excepto que se indique lo contrario– a una licencia de Reconocimiento-NoComercial-SinObraDerivada (BY-NC-ND) v.3.0 España de Creative Commons. Podéis copiarlos, distribuirlos y transmitirlos públicamente siempre que citéis el autor y la fuente (FUOC. Fundació per la Universitat Oberta de Catalunya), no hagáis de ellos un uso comercial y ni obra derivada. La licencia completa se puede consultar en <http://creativecommons.org/licenses/by-nc-nd/3.0/es/legalcode.es>

# Índice

|                                                        |           |
|--------------------------------------------------------|-----------|
| <b>Introducción.....</b>                               | <b>5</b>  |
| <b>Objetivos.....</b>                                  | <b>6</b>  |
| <b>1. La seguridad en las redes y en Internet.....</b> | <b>7</b>  |
| 1.1. Los atacantes .....                               | 7         |
| 1.2. Los ataques .....                                 | 8         |
| <b>2. Seguridad en los equipos informáticos.....</b>   | <b>10</b> |
| 2.1. Los virus informáticos .....                      | 10        |
| 2.2. Ataques desde Internet .....                      | 11        |
| 2.2.1. Los secuestradores de navegador .....           | 12        |
| 2.2.2. El software espía .....                         | 12        |
| 2.3. Protección del sistema .....                      | 12        |
| 2.3.1. Software antivirus .....                        | 13        |
| 2.3.2. Programas de limpieza .....                     | 13        |
| 2.3.3. Cortafuegos .....                               | 14        |
| <b>3. Seguridad en la información.....</b>             | <b>16</b> |
| 3.1. Cifrado de la información .....                   | 16        |
| 3.1.1. El cifrado de César .....                       | 16        |
| 3.1.2. El cifrado con raíles .....                     | 17        |
| 3.2. Sistemas actuales de cifrado .....                | 18        |
| 3.2.1. Cifrado de flujo .....                          | 19        |
| 3.2.2. Cifrado de bloque .....                         | 20        |
| 3.3. Sistemas de clave pública .....                   | 21        |
| 3.3.1. Cifrado de clave pública .....                  | 22        |
| 3.4. Firmas digitales .....                            | 24        |
| 3.4.1. Funciones de resumen .....                      | 25        |
| <b>4. Identidad digital.....</b>                       | <b>27</b> |
| 4.1. Comercio electrónico .....                        | 28        |
| 4.2. Correo electrónico seguro .....                   | 30        |
| 4.3. Trámites en línea .....                           | 31        |
| <b>5. Privacidad e Internet.....</b>                   | <b>33</b> |
| 5.1. Privacidad en la navegación web .....             | 33        |
| 5.1.1. El gran hermano en Internet .....               | 35        |
| 5.2. Las redes sociales .....                          | 36        |
| 5.3. El <i>phishing</i> .....                          | 37        |

|                                                        |               |
|--------------------------------------------------------|---------------|
| <b>6. Seguridad en la gestión audiovisual.....</b>     | <b>38</b>     |
| 6.1. Sistemas de control de copia y reproducción ..... | 38            |
| 6.2. Sistemas de marcaje .....                         | 39            |
| 6.2.1. Las marcas de agua .....                        | 40            |
| 6.2.2. Las huellas digitales .....                     | 40            |
| <br><b>Resumen.....</b>                                | <br><b>43</b> |
| <br><b>Actividades.....</b>                            | <br><b>45</b> |
| <br><b>Ejercicios de autoevaluación.....</b>           | <br><b>45</b> |
| <br><b>Solucionario.....</b>                           | <br><b>46</b> |
| <br><b>Glosario.....</b>                               | <br><b>47</b> |
| <br><b>Bibliografía.....</b>                           | <br><b>48</b> |

## Introducción

Los sistemas y protocolos que han dado lugar a Internet y a las redes de computadores no se diseñaron teniendo en cuenta la seguridad en su uso. En los inicios, los ordenadores eran accedidos por un grupo de usuarios que era controlable (se sabía quién accedía a ellos) y estaban cargados de buenas intenciones. Los ataques a los sistemas informáticos eran muy poco frecuentes comparado con hoy en día. Sin embargo, en el año 1969, Lance Hoffman escribió el artículo "Computers and Privacy", para la revista *ACM Computing Surveys*. En él explicaba que si todo lo relacionado con el procesamiento automático de datos no tiene en cuenta aspectos de seguridad y privacidad, el computador podría llegar a ser el malvado de nuestra sociedad. En consecuencia, la informática, uno de los mayores recursos de nuestra civilización, no evolucionaría como debería.

Afortunadamente, en paralelo al uso masivo de las tecnologías de la información y las comunicaciones, y al hacerse patente el acceso a estas tecnologías por parte de usuarios deshonestos, se fueron desarrollando protocolos y sistemas complementarios para proporcionar seguridad. Así pues, aunque los protocolos como IP o TCP no se diseñaron teniendo en cuenta miles de usuarios malintencionados que se podrían conectar a los sistemas informáticos desde todo el mundo, hay una serie de protocolos que hacen que las tecnologías de la información y las comunicaciones puedan desplegarse y usarse de manera segura.

Entender y saber usar las herramientas de seguridad y privacidad es esencial para ver si la sociedad de la información puede confiar suficientemente o no en las tecnologías a las que da apoyo.

En este módulo, estudiaremos los aspectos generales de la seguridad en los ordenadores y las redes. En primer lugar, veremos los ataques a los que están expuestos los equipos, los usuarios y la información que se guarda o se transmite. También veremos qué herramientas, basadas en la criptografía, fundamentan los sistemas y herramientas de seguridad, así como algunos ejemplos en los que se utilizan estas técnicas.

La privacidad de los usuarios de los ordenadores e Internet es un tema relacionado con la seguridad lo bastante importante como para que también veamos sus aspectos más relevantes. Así pues, dedicaremos a ello un apartado. Finalmente, veremos de manera introductoria los aspectos de seguridad en la gestión audiovisual.

## Objetivos

Los objetivos que el estudiante habrá alcanzado al finalizar este módulo son:

1. Conocer los diferentes tipos de ataque que puede sufrir un sistema informático.
2. Conocer las técnicas básicas para proteger un sistema.
3. Asimilar el funcionamiento de los sistemas de cifrado y distinguir sus diferentes tipos.
4. Comprender el concepto y uso de la firma digital, relacionándola con la identidad digital.
5. Conocer ejemplos de uso de la identidad digital.
6. Entender los riesgos que las tecnologías de la información y las comunicaciones comportan a la privacidad de sus usuarios.
7. Conocer de manera básica los sistemas de seguridad en la gestión de contenidos audiovisuales.



## 1. La seguridad en las redes y en Internet

En este apartado, estudiaremos qué tipo de aspectos relacionados con la seguridad nos podemos encontrar como miembros de la sociedad de la información. Una vez vistos estos aspectos, iremos desgranando, durante el resto del módulo, las técnicas y soluciones para dotar de seguridad a las tecnologías de la información y las comunicaciones.

### 1.1. Los atacantes

En primer lugar, veremos cuáles son los diferentes perfiles de atacantes. Seguro que la primera palabra en la que pensamos es *hacker*. A continuación, veremos que existen diferentes tipos de atacantes de los sistemas informáticos, pero este término es el más célebre, ya sea por los medios de comunicación, la literatura o el cine.

El término *hacker* tiene dos significados. Por una parte, se aplica frecuentemente para describir delincuentes informáticos, cuyo principal objetivo es romper las barreras de seguridad de los sistemas informáticos. Por otra parte, este término también se aplica a los informáticos brillantes que han hecho historia por algún hecho importante.

Por lo tanto, la palabra *hacker* puede ser aplicada o bien como elogio o bien por su vertiente peyorativa. De hecho, un *hacker* entrará ilegalmente en un sistema y sencillamente lo hará como demostración de que es un informático muy experto (con la consecuente satisfacción personal), o para evidenciar la falta de seguridad del sistema informático atacado. Si se entra al sistema para dejarlo inutilizado, el término que se aplica es *cracker*. Sea con la finalidad que sea, ser un *hacker* implica tener conocimientos muy avanzados de informática, ser constante e imaginativo.

En este módulo usaremos las expresiones *usuarios deshonestos* o *atacantes* para referirnos a aquellos usuarios que tengan como finalidad atacar a las redes de computadores y los sistemas de información. En este grupo de usuarios incluimos a los espías informáticos: estos usuarios aprovechan el acceso masivo a las tecnologías de la información y las comunicaciones por parte de los usuarios con el fin de obtener información privada.

#### Piratas informáticos

No se deben confundir los hackers con los piratas informáticos. Los piratas hacen copias ilegales a gran escala de software. Por su parte, un pirata también puede desactivar las funciones del registro legal, proporcionar claves de registro, etc.

## 1.2. Los ataques

Una vez que hemos definido quiénes son los atacantes, veremos qué tipo de ataques pueden sufrir las redes de computadores y los sistemas informáticos.

Uno de los ataques más elementales son los **ataques físicos**, que consisten en impedir que el hardware funcione adecuadamente. Para llevarlos a cabo, es esencial que el atacante tenga acceso físico al sistema atacado. El abanico de posibilidades es muy amplio: desde una desconexión de la red eléctrica o de comunicaciones hasta estropear el equipo a base de golpes o incluso robarlo. Evitaremos claramente estos ataques ubicando los equipos que contengan información importante en lugares en los que el acceso esté debidamente controlado.

Los servidores de una red de comunicaciones suelen estar ubicados en las salas de servidores. Estas salas tienen el acceso más o menos controlado y, además, disponen de detectores de humo, sistemas de extinción de incendios y sistemas de refrigeración, para contrarrestar las altas temperaturas a causa del funcionamiento de los equipos.

Unos de los ataques que se producen más a menudo y que tienen resonancia mundial son los ataques de **denegación de servicio**<sup>1</sup>, que consisten en impedir que los usuarios de un sistema puedan recibir servicio. Esta denegación de servicios se produce porque el sistema está siendo víctima de un colapso provocado por un atacante. Estos ataques son populares debido a que algunos grandes servicios de Internet han sido víctimas de este tipo de ataques.

<sup>(1)</sup>En inglés, *Denial of service attack* o *DoS attack*.

### Saturación por demandas

Un ataque habitual de denegación de servicio consiste en que el atacante genera miles de peticiones de conexión a un determinado servicio de Internet. Estas peticiones se dejan a medio hacer, de manera que quedan abiertas. Llega un momento en que el servidor está saturado y no puede atender más peticiones.

Otros ataques que también aparecen mucho en los medios de comunicación son los **ataques de acceso**, es decir, cuando alguien ha entrado de manera ilícita en un sistema informático. Puede suceder que el atacante haya conseguido burlar las barreras de seguridad o entrar como si se tratara de un usuario legal del sistema.

### Ejemplos de ataque de acceso

Un ejemplo de ataque de acceso sería el caso de un atacante que entrara a los sistemas de un órgano gubernamental e hiciera un cambio en su página web principal. Otro ejemplo consistiría en entrar en una red social o gestor de correo electrónico como otra persona con el objetivo de enviar información en nombre del atacado.

Entrar en los sistemas informáticos se puede lograr mediante complejas técnicas, como aprovechar agujeros en la seguridad del sistema operativo para infiltrar un programa que permita obtener el control del equipo. Por otra parte, si para entrar en un sistema se necesita una contraseña, ésta se puede conseguir gracias a las técnicas de **ingeniería social**.

### **Ejemplos de técnicas de ingeniería social**

Veamos a continuación un par de ejemplos de técnicas de ingeniería social:

- Recibimos una llamada telefónica diciendo que son los técnicos de nuestro proveedor de Internet. Éstos nos piden los datos correspondientes a nuestro servicio de correo electrónico, incluida la contraseña.
- Recibimos un mensaje de correo electrónico, supuestamente de nuestra entidad bancaria, y se nos pide introducir las credenciales de acceso al gestor bancario. Este tipo de técnicas de ingeniería social recibe el nombre de *phishing*.

Uno de los ataques históricamente más importantes y para los que se han desarrollado importantes protocolos de comunicaciones son los **ataques sobre la información**. Hay dos tipos de ataques, en función del papel del atacante:

- Por una parte, están los **ataques pasivos**. En este tipo de ataques, el atacante se limita a interceptar información y leer su contenido. Por ejemplo, un usuario de una LAN se dedica a recoger todas las tramas que se producen para intentar encontrar información importante (por ejemplo, contraseñas).
- Por otra parte, en los **ataques activos** el atacante modifica la información. Por ejemplo, se podría modificar el orden de pago de una tarjeta de crédito para que la cuenta corriente del destinatario del ingreso fuera la del atacante.

El éxito de estos ataques es completo si, además, el atacante pasa desapercibido. Por ejemplo, el hecho de que todas las tramas pasen por las manos del atacante, no debería impedir que éstas llegaran a su destino originario. Si no fuera así, alguien se daría cuenta de que hay algo que no funciona del todo bien.

## 2. Seguridad en los equipos informáticos

Los sistemas informáticos están formados por varios componentes. El software de los sistemas informáticos de los primeros ordenadores personales ya era una posible víctima de ataques por parte de virus. Hoy en día, a pesar de la gran cantidad de elementos que intervienen en los sistemas y la gran diversidad de programas existentes en el mercado y en Internet, la gestión de la seguridad de un equipo informático personal es bastante sencilla. Contrariamente, en los sistemas multiusuario, la seguridad implica otros conceptos más complejos, que están fuera del alcance de este material.

Dedicamos esta parte a explicar los conceptos básicos de seguridad en un equipo informático y a ver qué herramientas existen para poder hacerle frente.

Por una parte, veremos el software malintencionado. Se trata de un grupo de programas que tiene como objetivo inutilizar los sistemas o hacer que su uso se convierta en una molestia. Por otra parte, estudiaremos los ataques de acceso a los sistemas.

### 2.1. Los virus informáticos

Los virus informáticos son los programas malintencionados más populares, básicamente porque son los que desde hace más años han ido creando molestias a los usuarios o daños irreparables a sus sistemas informáticos.

Los **virus informáticos** son pequeños programas que se extienden por la ejecución de otros programas infectados. Cuando el usuario ejecuta este programa infectado, el virus queda residente en la memoria e irá infectando a otros programas.

#### El primer virus

El primer virus informático fue creado en el año 1972. Un ordenador infectado emitía aleatoriamente por pantalla un mensaje en tono burlón. Para eliminarlo, se creó, evidentemente, el primer antivirus.

Cuando los ordenadores personales tenían acceso limitado a Internet, el medio por excelencia de transmisión de virus eran los disquetes: un ordenador infectado iba infectando a los programas que contuvieran los disquetes que se introducían en el sistema. Los programas antivirus, capaces de detectar y eliminar virus, eran esenciales si los usuarios no querían ser víctimas de los virus: cuando se introducía un disquete en el ordenador y se desconocía si estaba infectado, se le pasaba el antivirus. Aunque hoy en día los disquetes no sean muy utilizados para pasar información de un sistema a otro, los virus pueden extenderse mediante dispositivos de memoria USB, ficheros descargados de Internet e incluso con ficheros adjuntos en correos electrónicos.

Los virus pueden crear multitud de molestias diferentes. Por una parte, los virus benignos no destruirán nunca la información contenida en el ordenador y se limitarán a mostrar textos de broma, a abrir y cerrar aleatoriamente el DVD del ordenador, a mover el ratón, etc.

Por contra, los virus malignos eliminarán datos del sistema informático o provocarán su cierre, sin dar al usuario la opción de guardar el trabajo que estaba haciendo.

Hay virus que se esconden detrás de programas con apariencia inofensiva: por ejemplo, una presentación de diapositivas para desearnos feliz año nuevo, o en un programa que nos promete una navegación más rápida. Cuando ejecutamos el programa, estamos abriendo un programa malintencionado que intentará dañar nuestro equipo y extenderse a otros. Este tipo de virus reciben el nombre de **troyanos**, en referencia al caballo de Troya que los griegos regalaron a los troyanos. Un método habitual de transmisión de los troyanos es mediante el correo electrónico: el troyano es el fichero adjunto, y para esparcirse se envía por correo electrónico a los contactos de la libreta de direcciones.

#### Ejemplo de virus benigno

Un virus muy popular en los años ochenta fue el "de la pelota" (*ping-pong* en inglés), que hacía aparecer una pelota dando botes por toda la pantalla.

## 2.2. Ataques desde Internet

La infección de un sistema por parte de un virus es, en el fondo, consecuencia de la ejecución de un programa por parte del usuario del sistema informático. Sin embargo, el acceso masivo a Internet ha propiciado la aparición y evolución de diferentes sistemas de ataque por medio del acceso al propio sistema informático por parte de usuarios o software externos.

Un ejemplo de estos sistemas son los **gusanos**, que se consideran una variante de los virus informáticos. A diferencia de estos últimos, los gusanos entran en el sistema directamente desde la Red. Dada la gran capacidad de reproducción de los gusanos, éstos pueden llegar a extenderse por miles de equipos de Internet en cuestión de poco tiempo, y se convierten en un problema serio. Evidentemente, cuando un gusano entra en el sistema, ejecutará su ataque, con los mismos efectos que puede tener cualquier virus.

En general, los gusanos entran al sistema gracias a deficiencias de programación. A menudo, se trata de factores que los programadores del sistema no tuvieron en cuenta y que, una vez descubiertos por la comunidad *hacker*, o son comunicados a los programadores o son aprovechados por los *crackers* para atacar o entrar en los sistemas mediante los **exploits**.

Un **exploit** es un software que aprovecha una deficiencia de programación del sistema operativo para conseguir algún objetivo relacionado con el acceso al sistema rompiendo las barreras de seguridad.

Un *cracker* podría proceder de la siguiente manera: en primer lugar, detectaría que detrás de una determinada dirección IP se esconde una máquina con un determinado sistema operativo. Una vez identificada la versión del sistema y los programas que se están ejecutando, el *cracker* enviará una serie de *exploits*. Si el sistema no está debidamente protegido, se nos podría instalar el *exploit*. A partir de ese momento, el *cracker* podría tener acceso a determinadas partes de nuestro sistema, o tener un acceso total.

#### Actualizaciones del sistema operativo

Notad que si el sistema está actualizado, seguro que tendrá arregladas las deficiencias de programación que se hayan podido detectar con anterioridad.

### 2.2.1. Los secuestradores de navegador

El navegador de Internet es uno de los softwares más utilizados hoy en día. Por ello, hay una parte importante del software malintencionado dedicada a afectar el comportamiento y el uso de los navegadores. Este software recibe el nombre, en inglés, de *browser hijacker*, que podríamos traducir como secuestrador de navegador.

#### Ejemplos de secuestradores de navegadores

Ilustramos con un par de ejemplos lo que podrían llegar a hacer estos programas secuestradores de navegadores:

- Uno de estos programas podría abrir ventanas del navegador que contuvieran publicidad variada, como medicinas de calidad dudosa, casinos en línea de poca confianza e incluso pornografía. Una variante consistiría en que el programa ha modificado la página de inicio del navegador, aquella que se abre automáticamente al iniciar el navegador. A pesar de cambiarlo desde las opciones del programa, el cambio no haría efecto, ya que el secuestrador nos vuelve a cambiar la página.
- Un ejemplo más sofisticado consiste en que el secuestrador modifica los resultados de las búsquedas que hacemos en Internet. Si por ejemplo buscamos información sobre la UOC, los primeros enlaces que mostraría como resultado serían páginas que no tienen nada que ver.

### 2.2.2. El software espía

Finalmente, hacemos una mención al **software espía**<sup>2</sup>. Se trata de un tipo de software malintencionado cuyo objetivo es enviar a un servicio remoto informes de la actividad del usuario, evidentemente sin que éste lo sepa. La información más habitual que se envía es el nombre de los servicios web que se están utilizando, las palabras que se están buscando, etc., aunque también puede haber software espía capaz de enviar contraseñas y datos más importantes.

<sup>(2)</sup>En inglés, el software espía recibe el nombre de *spyware*.

### 2.3. Protección del sistema

La protección de un sistema hacia estos ataques implica, en primer lugar, tener conciencia de la existencia y del peligro o incomodidad que pueden suponer éstos para los usuarios. En segundo lugar, implica tener un grupo de herramientas instaladas en el sistema. Un sistema bien actualizado con las herramientas adecuadas estará mejor protegido que un sistema sin ningún tipo de protección.

Hay que decir que los sistemas operativos actuales son, en mayor o menor medida, susceptibles de sufrir la instalación de software malintencionado, en función del diseño del propio sistema operativo. Los sistemas que potencialmente pueden recibir más ataques son los de Microsoft, dado el gran número de usuarios que lo utilizan y su diseño sin considerar fuertes medidas de seguridad. Las últimas versiones de estos sistemas o incorporan herramientas que ayudan a la protección del sistema o aconsejan su instalación al usuario.

Clasificamos las herramientas esenciales de protección en tres tipos: los programas antivirus, los programas de limpieza y los cortafuegos.

### **2.3.1. Software antivirus**

Los antivirus son programas encargados de detectar virus en el sistema e intentar eliminarlos. Actualmente hay miles de virus informáticos. Cada uno de ellos tiene una huella identificativa propia, es decir:

- Un patrón conocido relacionado con el código del programa que lo implementa. Algunos virus son más complejos, en el sentido de que son capaces de mutar y generar variaciones de esta huella.
- Una serie de información depositada en el registro del sistema operativo, en el sistema de ficheros, etc., que indica que el sistema está infectado.

Los programas antivirus se dedican a comparar la información del sistema informático (registro, sistema de ficheros, procesos en ejecución, etc.) con una base de datos de huellas y otra información que permite identificar si hay un virus. Se buscan virus en la memoria y en los discos del ordenador. Si un virus se detecta, casi siempre se puede eliminar sin dificultad.

Para el buen funcionamiento de este software, conviene que los usuarios estén siempre actualizados: con periodicidad frecuente se actualizan las bases de datos de definiciones de virus para poder afrontar la detección de nuevos virus y mutaciones.

### **2.3.2. Programas de limpieza**

De una manera parecida a los antivirus, los programas de limpieza examinan los equipos para detectar y eliminar otro tipo de software malintencionado, como secuestradores de navegador, y todo aquello relacionado con el software espía. Incluso hay herramientas que son una combinación de antivirus y programas de limpieza.

De la misma manera que en los antivirus, conviene que las herramientas de limpieza estén actualizadas, ya que la diversidad de software malintencionado aumenta con una frecuencia considerable.

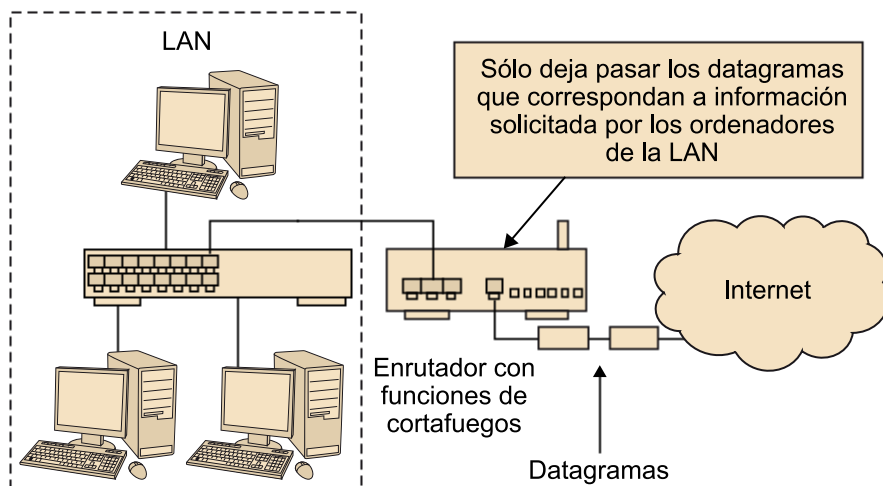
### 2.3.3. Cortafuegos

Los cortafuegos fueron concebidos como elementos para controlar el acceso a las redes de computadores. Un cortafuegos, por ejemplo, podrá controlar las conexiones que vienen del exterior y que tienen como destino un servidor web que se encuentra ubicado en nuestra LAN.

Los **cortafuegos de red** examinan los datagramas que van de una red a otra y, en función de parámetros como los puertos de origen o destino y las direcciones IP de origen y destino, deniegan o permiten el acceso a estos datagramas. La mayoría de *routers* pueden realizar también la funcionalidad de cortafuegos de red.

La función de los cortafuegos se ilustra en la figura siguiente. Una de las configuraciones típicas de los cortafuegos de red es denegar por defecto todos los accesos a la red y dejar pasar sólo aquellas conexiones que estén permitidas. En el caso de la figura, sólo se permite la entrada de los datagramas que se corresponden a información pedida por los ordenadores de la LAN. De esta manera, los datagramas de conexiones iniciadas desde Internet (y que podrían corresponder a ataques) no pasarán por el cortafuegos.

Ejemplo de un *router* con funcionalidad de cortafuegos



Así pues, un sistema sin cortafuegos no tendrá el acceso controlado y, en consecuencia, será susceptible de sufrir más ataques desde el exterior que un equipo con cortafuegos.

Dado el crecimiento de los ataques externos contra equipos informáticos personales, han ido apareciendo varios programas con funcionalidad de **cortafuegos local**. Estos cortafuegos controlan el tráfico de red que entra al equipo o que sale de él: programas que intentan acceder a servicios remotos de Internet, usuarios de la LAN que intentan entrar en nuestro sistema de ficheros, etc.



Con un cortafuegos instalado y bien configurado, el equipo estará mejor protegido contra los gusanos y los ataques mediante *exploits*.

### 3. Seguridad en la información

En esta parte, nos dedicaremos a conocer las técnicas que permiten dotar de seguridad la información que se transmite en una red. Antes de concretar varios aspectos, vamos a definirlos.

Por una parte, estudiaremos cómo se puede conseguir la **confidencialidad** de la información que se transmite mediante una red: es decir, que la información sólo esté disponible para los participantes en la comunicación. También veremos cómo garantizar la **autenticidad** de los datos, es decir, garantizar que la información no ha sido modificada y que proviene realmente de donde se nos indica. Este concepto de autenticidad está fuertemente ligado al concepto de seguridad en la identidad, que desarrollaremos en el apartado 4.

Para tratar la confidencialidad, disponemos de las herramientas de cifrado, mientras que para tratar la autenticidad, disponemos de las herramientas de integridad y las firmas digitales. Las técnicas que estudiaremos en esta parte forman parte de la criptografía.

#### 3.1. Cifrado de la información

El cifrado de la información y la propia criptografía se remontan a épocas muy antiguas. Los primeros sistemas de cifrado, que datan de la época de la Grecia clásica y fueron muy usados durante el Imperio Romano y la Edad Media, se basan en dos técnicas muy sencillas: la sustitución de letras y el cambio de posición.

Los métodos clásicos de cifrado consisten en sustituir una letra por otra y/o cambiar de orden las letras que conforman el texto.

Mediante el estudio de un par de sistemas clásicos de cifrado, aprenderemos una serie de conceptos relacionados con el cifrado y la criptografía en general.

##### 3.1.1. El cifrado de César

Al parecer, Julio César utilizaba este sistema para comunicarse con sus generales: les enviaba un mensaje cifrado y solamente el general a quien iba dirigido podía leer su contenido.

El general en cuestión conocía la **clave** del cifrado, que, evidentemente, también era conocida por Julio César. Si, por ejemplo, la clave era 3 y el mensaje original (técnicamente llamado **texto en claro**) era 'LAVIDAESBELLA', el

mensaje cifrado sería 'ODYLGDHVEHOOD'. Para conseguirlo, se avanzaban tres posiciones del alfabeto para cada una de las letras. Si la letra que se quería cifrar era la 'a', el resultado sería la 'd'.

Para descifrar el mensaje, se realizaba el proceso inverso, es decir, se retrocedían tres posiciones del alfabeto.

Si es interesante ver cómo cifrar la información, también lo puede ser intentar conocer los contenidos de los mensajes sin conocer la clave. Esto es lo que se conoce como **romper un criptosistema**.

Para romper el sistema, podemos optar por un **ataque de fuerza bruta**, consistente en probar todas las claves posibles hasta dar con un texto descifrado que se entienda.

Por lo tanto, un atacante que interceptara al mensajero y le robara el mensaje podría ir probando todas las claves posibles hasta obtener una correcta:

- Si retrocede una posición, obtendría 'NCXKFCGUDGNNC', que es ininteligible.
- Si retrocede dos posiciones, obtendría 'MBWJEBFTCFMMB', que también lo es.
- Al retroceder tres posiciones, obtendría 'LAVIDAESBELLA', que ya se puede leer.

Fijémonos en que, en el peor de los casos, habría que probar tantas veces como claves posibles existen. Notad que el atacante debería conocer también que el sistema de cifrado es César y no otro. Sin embargo, fijémonos en un detalle: el hecho de que haya dos 'O' seguidas (ODYLGDHVEHOOD) podría indicar que la 'L' se ha convertido en 'O' y, por lo tanto, se podría llegar a deducir la clave. Por ello puede resultar interesante combinar este cifrado con uno parecido al que describimos a continuación.

### 3.1.2. El cifrado con raíles

Ahora veremos un sistema basado en el cambio de orden de las letras del texto en claro: el cifrado con raíles<sup>3</sup>. En este sistema, el texto se coloca repartido en raíles, en forma de zigzag (por ello también se conoce con el nombre de zigzag). Suponiendo que la clave sea 2 y utilizando el texto en claro 'LAVIDAESBELLA', se procedería como sigue.

<sup>(3)</sup>En inglés, *Rail Fence Cipher*.

En primer lugar, se colocarían las letras sobre 2 raíles, en zigzag:

|   |   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| L |   | V |   | D |   | E |   | B |   | L |   | A |
|   | A |   | I |   | A |   | S |   | E |   | L |   |

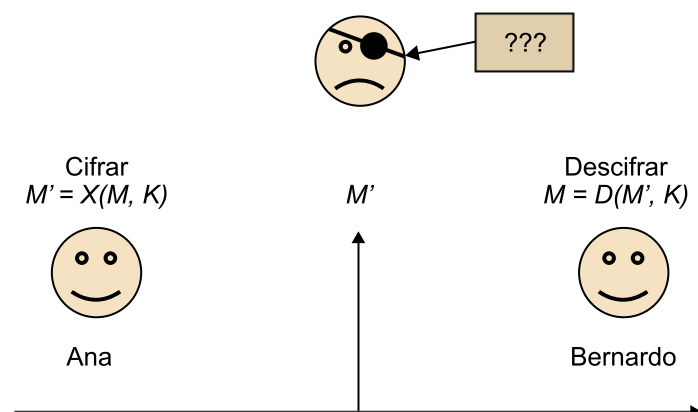
Y se leería el mensaje línea a línea, dando lugar a 'LVDEBLAAIASEL'. Para romper estos sistemas, además de que el atacante sepa que hemos cifrado con este método, habría que probar con todos los números posibles de raíles.

### 3.2. Sistemas actuales de cifrado

Durante siglos se han ido utilizando sistemas similares a los anteriores o combinaciones complejas con ambos sistemas. Debemos tener presente que la aparición de máquinas de cifrado permitían cifrar mensajes de manera compleja con un tiempo bastante corto. Sin embargo, la aparición de los ordenadores en la segunda mitad del siglo XX van a permitir el éxito de los ataques de fuerza bruta en un tiempo factible.

Con la aparición de la informática y el uso de ésta para el cifrado de las comunicaciones, los mensajes no se cifran letra por letra, sino a partir de los bits que los forman. Un ejemplo se podría ver en la figura siguiente. Se podría partir de la traducción con código ASCII de un mensaje de texto para tener la tira de bits que lo representa. Existe un algoritmo  $X$  que cifra un mensaje  $M$  mediante la clave  $K$ . Paralelamente, existe un algoritmo  $D$  que obtiene el texto en claro a partir del mensaje cifrado  $M'$  siempre y cuando se utilice **la misma clave  $K$** .

Ejemplo de aplicación de cifrado con clave compartida. El atacante no puede interpretar el mensaje.



Los ejemplos del cifrado de César y el de raíles forman parte de lo que se llama **cifrado de clave secreta, compartida o cifrado simétrico**. En estos casos, la clave es conocida tanto por el emisor como por el receptor. Por lo tanto, quien no conozca la clave será incapaz de entender la información, a no ser que insista en la ruptura por la fuerza bruta.

Por otra parte, existen dos alternativas a la hora de cifrar con clave simétrica:

#### Ved también

Recordad que hemos tratado el código ASCII en el módulo "Aspectos tecnológicos de los sistemas informáticos".

#### La máquina Enigma

La máquina Enigma es un artilugio electromecánico capaz de cifrar y descifrar mensajes. Inventada en los años veinte es famosa por haber sido utilizada por los alemanes durante la Segunda Guerra Mundial.

- En lugar de utilizar un algoritmo complejo, el mensaje cifrado es el resultado de combinar el mensaje con una clave tan larga como el mensaje. Esta alternativa se llama **cifrado de flujo**.
- Utilizar combinaciones muy complejas de las técnicas vistas en el subapartado 3.1. El texto en claro se divide en una serie de bloques que van entrando en el sistema de cifrado y van produciendo, en bloques, el texto cifrado. A esta alternativa se la conoce como **cifrado de bloque**.

A continuación, analizaremos más a fondo estas propuestas.

### 3.2.1. Cifrado de flujo

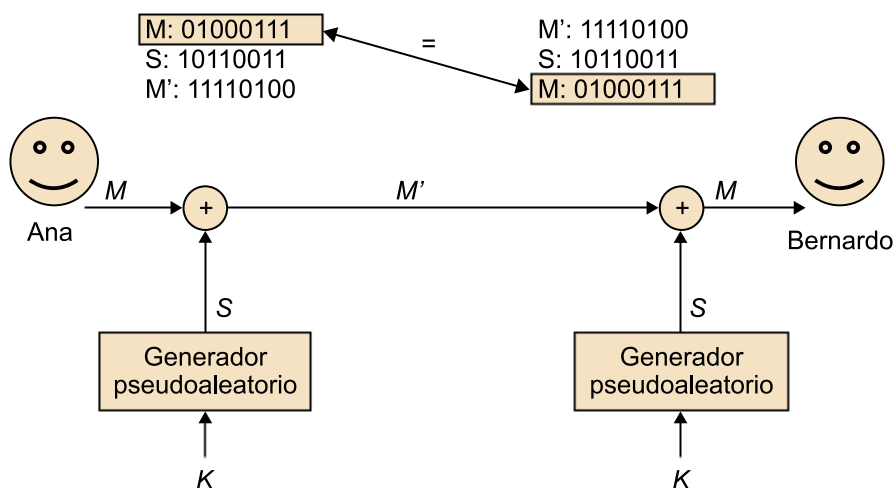
El cifrado de flujo se utiliza mucho en las telecomunicaciones. Por ejemplo, en una conversación de telefonía móvil la voz se digitaliza (es decir, se convierte a un flujo de bits) y se envía cifrada por la red de comunicaciones. Con el fin de no entorpecer la conversación, el proceso de cifrado debería ser lo bastante rápido como para no añadir retraso a la comunicación. Por ello, conviene que la operación de cifrado sea rápida.

La figura siguiente muestra cómo se utiliza el cifrado de flujo. En concreto, cada bit de entrada al sistema de cifrado (el mensaje  $M$ ) se combinará, usando la función lógica XOR, con el bit correspondiente del flujo clave  $S$ , para dar lugar al bit correspondiente al flujo de salida. El receptor hará el mismo proceso de combinación con la XOR para obtener el flujo descifrado.

#### Ved también

En el módulo "Aspectos tecnológicos de los sistemas informáticos", hemos estudiado el concepto de funciones lógicas, como el XOR.

Ejemplo de uso del cifrado de flujo en el que interviene un generador pseudoaleatorio que utiliza una clave compartida



La fortaleza de los sistemas de cifrado de flujo se basa en la clave utilizada para cifrar. Sin entrar en detalles, podríamos decir que se trata de una clave aleatoria y muy larga, a menudo tan larga como la tira de bits que se acabará cifrando.

Ahora bien, ¿cómo podemos tener una clave aleatoria tan larga? Si es aleatoria, ¿cómo la podemos ofrecer al receptor de la información? La solución a estas cuestiones pasa por conocer el concepto de **generador pseudoaleatorio**.

Un **generador pseudoaleatorio** es un algoritmo que, a partir de un mismo valor de entrada o clave, genera el mismo flujo de bits de salida, que tiene el aspecto de secuencia aleatoria.

A fin de que la secuencia parezca aleatoria es necesario que el número de ceros producidos sea similar al número de unos producidos. Sin embargo, habrá un instante en el que esta secuencia se volverá a repetir. Este instante define el período de la secuencia y, cuanto más largo sea este período, más aleatoria parecerá la secuencia. La siguiente secuencia pseudoaleatoria tiene el período en negrita:

**01100110101100110110011001101011001101100110011**

Es evidente que los generadores pseudoaleatorios van generando bits en un tiempo bastante rápido con el fin de no introducir retrasos en las comunicaciones.

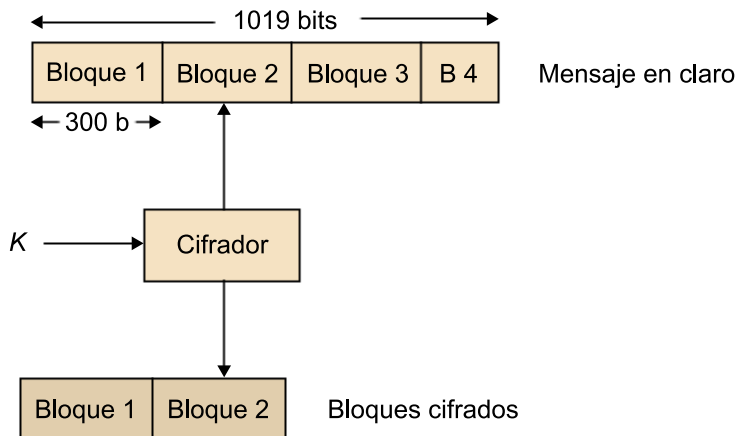
### 3.2.2. Cifrado de bloque

El cifrado de bloque utiliza combinaciones complejas basadas en sustituciones y cambios de posición que se regirán por la clave de cifrado. Estos sistemas son más costosos, tanto a nivel de fabricación de dispositivos como a nivel computacional, que los sistemas de cifrado de flujo (generación de la secuencia incluida). Por contra, con claves relativamente cortas, de 128 o 256 bits, ofrecen una seguridad lo bastante buena contra los ataques de fuerza bruta. El funcionamiento del cifrado de bloque tiene diferentes variantes. La más simple es el libro de códigos electrónico<sup>4</sup> (ECB), que consiste en que la salida correspondiente a un determinado bloque depende de la clave y del propio bloque.

<sup>(4)</sup>En inglés, *Electronic Codebook*.

En la siguiente figura, se muestra la aplicación del cifrado de bloque con la variante ECB. El mensaje inicial, de 1.019 bits, se debe dividir en bloques. Como el cifrador del ejemplo trabaja con bloques de 300 bits, se necesitan tres bloques: los tres primeros de 300 bits y el último de 119.

### Ejemplo de aplicación del cifrado de bloque



Aplicación del cifrado de bloque según el ejemplo del texto, en un estado intermedio en el que se está cifrando el segundo bloque de texto

Uno de los primeros sistemas utilizados en la informática fue el estándar para el cifrado de datos<sup>5</sup> (DES). Este sistema dividía el mensaje de entrada en bloques de 64 bits y utilizaba una clave de 56 bits. A medida que los ordenadores fueron ganando potencia de cálculo, el sistema DES estaba más cerca de quedar inutilizado porque se podría romper, mediante la fuerza bruta, en un tiempo factible (quizá en unas horas). Así pues, se empezó a utilizar una variante, el triple DES, con claves de 192 bits y que consistía en usar varias veces el DES.

<sup>(5)</sup>En inglés, *Data Encryption Standard*.

## Ataques de fuerza bruta

Si un algoritmo tarda un nanosegundo en ejecutarse y queremos probar todas las claves de 256 bits posibles, deberemos ejecutar el algoritmo  $2^{256}$  veces. Por tanto, serán  $2^{256}$  nanosegundos, es decir, tardaremos muchos años en romper el sistema. El conocimiento a fondo de los sistemas de cifrado permite diseñar estrategias de fuerza bruta más elaboradas, que a su vez posibilitan la ruptura de algunos sistemas en "menos años".

Actualmente, el algoritmo que sustituye el DES y el 3DES es el AES<sup>6</sup>, que utiliza bloques de 128 bits y claves de hasta 256 bits.

<sup>(6)</sup>En inglés, *Advanced Encryption Standard*.

## Fortaleza digital

El libro *Fortaleza digital* de Dan Brown trata sobre un escenario hipotético en el que todo el mundo cree que los sistemas de cifrado no se pueden romper. Sin embargo, los servicios de inteligencia saben descifrar los mensajes que el resto de la humanidad (delincuentes incluidos) cree que nadie puede descifrar.

### 3.3. Sistemas de clave pública

Hasta ahora, hemos hablado de diferentes sistemas de cifrado y hemos aprendido algunos conceptos básicos de la criptografía. Sin embargo, no hemos prestado atención al hecho de que los sistemas de clave compartida tienen como punto débil la propia clave:

- La clave se debe compartir, por lo que es necesario un canal seguro de transmisión de claves entre los comunicantes. Una manera de transmitir la clave sería mediante un encuentro presencial entre los participantes en

la comunicación, en la que se pasarían la clave grabada, por ejemplo, en una memoria USB.

- A priori, necesitamos una clave para cada una de las diferentes parejas de participantes que se quieran comunicar.

Todo esto provoca que generalizar un sistema de cifrado de clave compartida en un entorno global como Internet sea poco factible. Afortunadamente, en el año 1976 tres matemáticos, Whitfield Diffie, Martin Hellman y Ralph Merkle, diseñaron un sistema que permite a los dos participantes poder calcular una clave compartida sin la necesidad de pasarse información comprometida, conocido como Diffie-Hellman.

Con este sistema se inicia la **criptografía de clave pública** o **criptografía asimétrica**.

La criptografía de clave pública se basa en el uso de dos valores o claves: uno es la **clave pública**, que todo el mundo la puede conocer; el otro es la **clave privada**, que debe estar custodiada por su propietario.

#### La clave pública

La clave pública resuelve los problemas que tenían los sistemas de clave simétrica en cuanto a la cantidad de claves necesarias para la comunicación con un gran número de usuarios potenciales y la distribución de las claves.

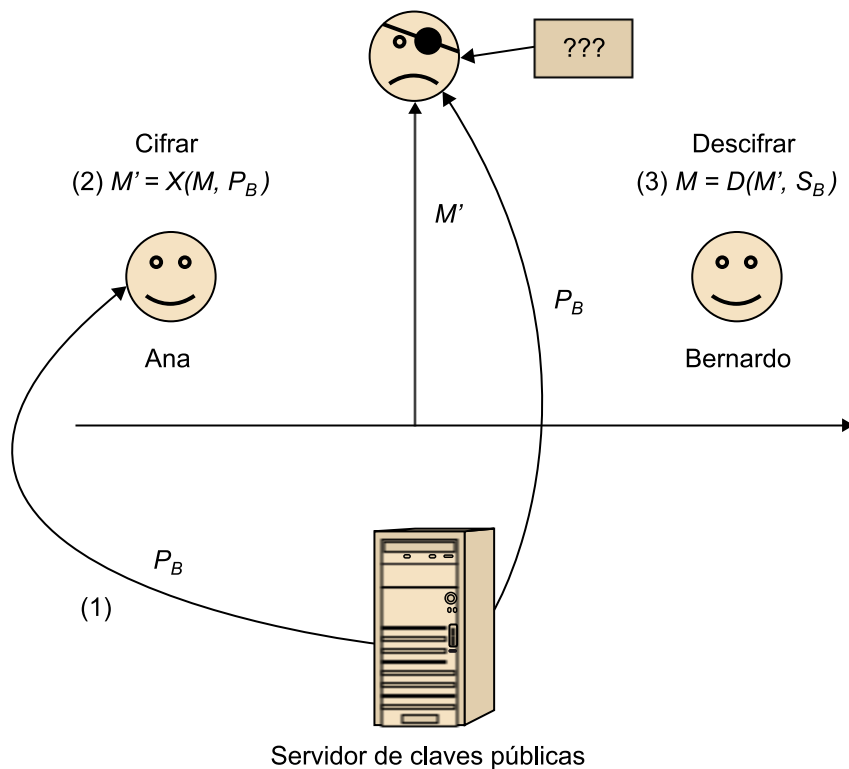
### 3.3.1. Cifrado de clave pública

Para ver qué papel juegan las claves públicas y privadas en el proceso de cifrado, pongamos un ejemplo.

Supongamos que Ana quiere enviar un mensaje secreto a Bernardo. Ana tiene la clave privada  $S_A$  y la clave pública  $P_A$ . Bernardo también tiene su par de claves  $S_B$  y  $P_B$ .



## Proceso de cifrado con un sistema de clave pública



Como se muestra en la anterior figura, se procederá de la siguiente manera:

- Ana obtendrá, mediante un servicio de Internet, la clave  $P_B$  (paso 1 de la figura).
- Ana utilizará un algoritmo de cifrado  $X$  para cifrar el mensaje  $M$  y obtener el mensaje cifrado  $M'$  (paso 2 de la figura).

$$M' = X(M, P_B)$$

- Al recibir  $M'$ , Bernardo utilizará la clave  $S_B$  (es decir, su clave secreta) para ejecutar el algoritmo de descifrado y obtener  $M$  (paso 3 de la figura).

$$M = D(M', S_B)$$

Podéis ver que Bernardo es el único que podrá descifrar el mensaje, dado que es el único que debería tener acceso a  $S_B$ . Así pues, el usuario deshonesto de la figura, a pesar de tener acceso al mensaje cifrado  $M'$  y a la clave pública de Bernardo, no será capaz de descifrar el mensaje.

### Uso de candados

Podemos hacer la analogía de la clave privada con los candados: el candado es la clave pública y la llave que lo puede abrir es la clave privada. Todo el mundo puede utilizar un candado para cerrar una caja, de manera que sólo el propietario de la llave del candado pueda abrirla.

### Protección de la clave privada

Las claves privadas se guardan protegidas con una contraseña o en un dispositivo seguro como una tarjeta inteligente (a la que también se accederá mediante una contraseña).

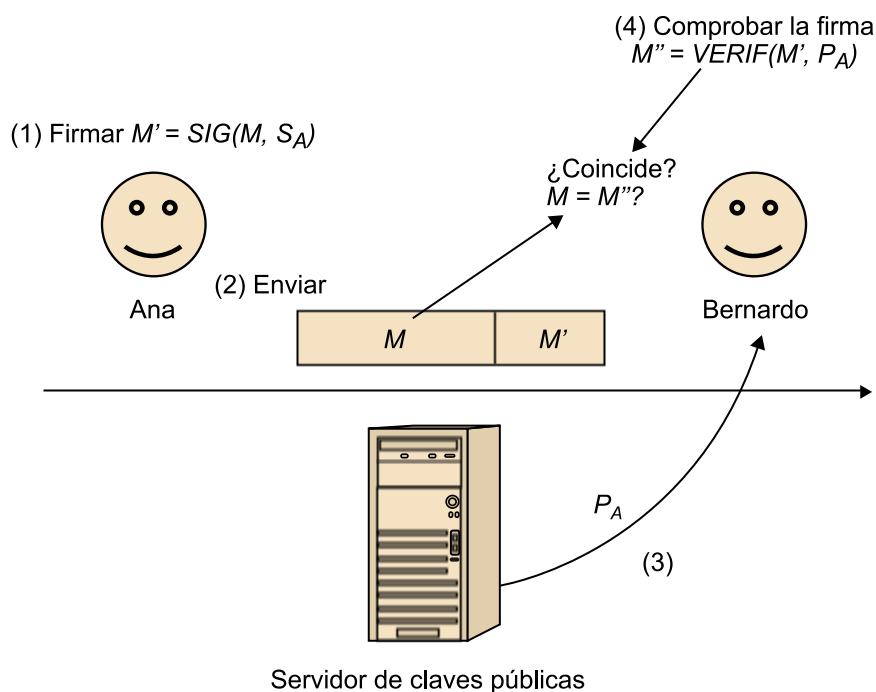
### 3.4. Firmas digitales

Ahora veremos cómo se consigue comprobar la autenticidad de la información con la misma criptografía de clave pública mediante lo que se llaman **firmas digitales**.

Las **firmas digitales** son una aplicación de la criptografía de clave pública que permite dar autenticidad de origen a la información enviada, asegurar la integridad e impedir el rechazo de quien firma.

Seguimos con el ejemplo de Ana y Bernardo y sus claves públicas y privadas. La siguiente figura ilustra este proceso.

Proceso de firma con un sistema de clave pública



Supongamos que Ana quiere hacer una firma digital de un mensaje  $M$  que enviará a Bernardo:

- Ana utilizará un algoritmo  $\text{SIG}$  de firma, con su clave privada  $S_A$ , para producir la firma del mensaje  $M$  (paso 1 de la figura):

$$M' = \text{SIG}(M, S_A)$$

- Ana enviará a Bernardo el mensaje  $M$  junto a su firma  $M'$  (paso 2 de la figura).

Una vez que Bernardo reciba el mensaje y su firma, procederá como sigue:

- Bernardo obtendrá la clave pública de Ana mediante un servicio de Internet (paso 3 de la figura).
- Bernardo usará el algoritmo *VERIF* de verificación, esta vez poniendo la firma y la clave pública de Ana (paso 4 de la figura):

$$M'' = \text{VERIF}(M', P_A)$$

- Si el resultado  $M''$  del algoritmo coincide con el mensaje original  $M$ , querrá decir que el mensaje recibido es auténtico.

Como ya hemos apuntado anteriormente, los sistemas de firma digital aseguran varias propiedades. Vamos a verlas con más detalle con el ejemplo anterior:

- Dado que Ana es la única que tiene acceso a la clave  $S_A$ , ella es la única que puede haber firmado el mensaje. Por lo tanto, **el origen del mensaje es auténtico**, ya que no lo puede haber firmado nadie más. Fijémonos en que una firma "real" sería más fácil de falsificar que una digital.
- Como Ana es la única que conoce su clave, no podrá decir nunca que ella no ha firmado el mensaje. Esta propiedad se llama **no repudio**.
- Finalmente, las propiedades de los sistemas de firma digital provocan que si el mensaje  $M$  se modificara durante el camino, aunque estuviera sólo en un bit, la comprobación de la firma ya no funcionaría. Por lo tanto, las firmas digitales garantizan la **integridad** de la información.

### 3.4.1. Funciones de resumen

Una de las diferencias entre los algoritmos de criptografía de clave compartida y los de clave pública es que estos últimos son más costosos computacionalmente. Por ejemplo, un programa emplea más tiempo en aplicar un algoritmo de clave pública para cifrar un mensaje que al cifrarlo con un método de clave compartida. Además, las claves de los sistemas de clave pública son más largas que las usadas en clave compartida (1.024, 2.048 o más bits).

En el caso de las firmas digitales, con el fin de realizar las firmas más rápidamente no se firma el mensaje, sino un resumen de éste.

Las **funciones de resumen** generan una tira de bits de una longitud determinada a partir de un mensaje de cualquier longitud.

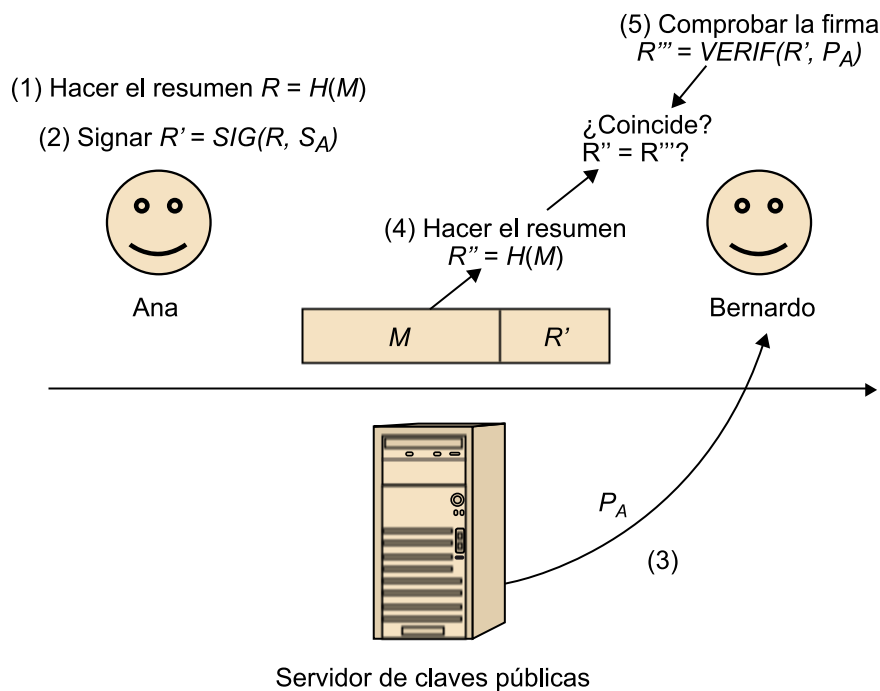
#### SHA

Una de las funciones de resumen más utilizadas es SHA (del inglés *Secure Hash Algorithm*, algoritmo de resumen seguro), que genera resúmenes de 160 bits.

Estas funciones de resumen, para que puedan ser utilizadas en los algoritmos de firma digital, deben cumplir varias propiedades. Una de ellas es que una alteración mínima en el mensaje debe dar un resumen diferente que si el mensaje no se hubiera alterado (cuestión estrechamente relacionada con la propiedad de integridad).

Así pues, tal como vemos en la siguiente figura, Ana primero hará el resumen del mensaje  $M$  utilizando la función de resumen  $H$  (paso 1 de la figura) y acto seguido hará la firma de este resumen (paso 2 de la figura). Como en el caso anterior, Bernardo obtendrá la clave pública de Ana mediante un servicio de Internet (paso 3 de la figura). Acto seguido, Bernardo hará el resumen del mensaje recibido (paso 4 de la figura) y utilizará este resumen para verificar la validez de la firma digital recibida de Ana (paso 5 de la figura). Observad que hacer el resumen no precisa del conocimiento de ninguna clave pública ni privada.

Proceso de firma de clave pública con funciones de resumen



## 4. Identidad digital

Si Ana envía un mensaje cifrado a Bernardo, debe utilizar la clave pública de Bernardo. Por ejemplo, se la descargará de la página web de Bernardo y la guardará en el disco del ordenador para que el software criptográfico de cifrado la pueda utilizar. Fijémonos en que es fundamental estar seguros de que la clave pública de Bernardo es realmente de éste.

La **identidad digital** es todo aquello que hace referencia a asegurar que una determinada clave pública se corresponde con un determinado individuo.

### Ejemplo de identidad digital

Uno de los ejemplos más claros del concepto de identidad digital es el DNI electrónico: éste consiste en una tarjeta inteligente que contiene nuestra clave privada y, al mismo tiempo, nuestra clave pública. Con esta clave nos podemos identificar para realizar varios trámites vía Internet.

Con el fin de asegurar que una clave pública determinada se corresponde con una identidad concreta, se utiliza un sistema de certificados de clave pública.

Un **certificado de clave pública** es un documento electrónico que vincula una clave pública a una identidad.

La información básica que contiene un certificado es:

- La identidad que certifica, por ejemplo una dirección de correo electrónico o el número de DNI de un ciudadano.
- El período de validez, es decir una especie de fecha de caducidad a partir de la cual el certificado no será reconocido como válido.
- La clave pública que se certifica.
- El nombre del emisor del certificado. Se trata de una **autoridad de certificación**, un organismo que puede expedir certificados de clave pública.

Las autoridades de certificación pueden ser órganos regulados por los países, como la Fábrica Nacional de Moneda y Timbre, o empresas certificadoras de gran reputación (como VeriSign). Sin embargo, puede haber autoridades de certificación de ámbito más pequeño (por ejemplo, para hacer certificados para las claves públicas del personal de la UOC).

Las grandes empresas certificadoras internacionales ya son reconocidas por la mayoría de software criptográfico. En este sentido, cuando el software debe utilizar una clave pública certificada por alguna de estas empresas, lo hace sin ningún problema, ya que se trata de una autoridad de confianza.

#### Autoridades de confianza

Las autoridades de certificación pueden hacer una petición a los fabricantes de software para que las incluyan como autoridades de confianza.

Cuando se utiliza una clave pública certificada por una autoridad que no está reconocida por el software, éste pregunta al usuario sobre si se confía en la veracidad de la autoridad. Por lo tanto, en el fondo, son los usuarios quienes deciden si seguir adelante o no. Si se dice que sí, la autoridad pasará a ser reconocida, y el software ya no nos preguntará cuando se utilicen otras claves públicas certificadas por esta autoridad.

El ámbito de la autoridad de certificación tiene mucho que ver con el uso de la clave pública. Por ejemplo, podría ser que una clave pública certificada por una autoridad de certificación dependiente de un organismo autonómico sólo tenga validez para trámites en línea de este organismo. O incluso puede haber reconocimientos entre autoridades: por ejemplo, la clave certificada por un organismo estatal puede utilizarse en trámites del organismo autonómico.

Las autoridades de certificación disponen de un sistema generador y gestor de claves y certificados: las **infraestructuras de clave pública**. Estas infraestructuras también disponen de **listas de revocación** que sirven para especificar qué certificados han dejado de ser válidos antes de que caducaran (por ejemplo, porque el usuario ha perdido la clave privada y otro la podría utilizar).

Hasta aquí hemos estudiado las bases del cifrado y la firma digital. En este apartado, veremos, brevemente, unos ejemplos de uso del cifrado y la firma digital en Internet: el comercio electrónico, el correo electrónico seguro y los trámites en línea.

### 4.1. Comercio electrónico

El comercio electrónico ha supuesto una revolución en la manera de vender y comprar. Desde nuestra casa y utilizando el navegador, podemos comprar entradas de cine, ropa, billetes de avión, hacer reservas de hotel, etc. En cuanto a la seguridad, uno de los puntos más importantes es el momento de hacer el pago de la compra. Por ejemplo:

- Conviene que la información que enviamos sea confidencial, por ejemplo, a la hora de enviar los datos de la tarjeta de crédito.

- También es importante que el servicio al que nos conectamos sea auténtico. Es decir, que cuando damos los datos de la tarjeta de crédito los estamos dando al servicio en línea del banco y no a una web falsa creada por unos falsificadores de tarjetas.

En el comercio electrónico se utilizan sistemas de cifrado simétrico y sistemas de clave pública. Cuando nos conectamos al servicio del banco, lo hacemos sobre una conexión segura SSL<sup>7</sup> (una extensión de las conexiones TCP que garantizan propiedades de seguridad de la información transmitida). Estas conexiones seguras se identifican por medio de un candado que vemos en el propio navegador que mostramos en la siguiente figura.

(7)En inglés, *Secure Socket Layer*.

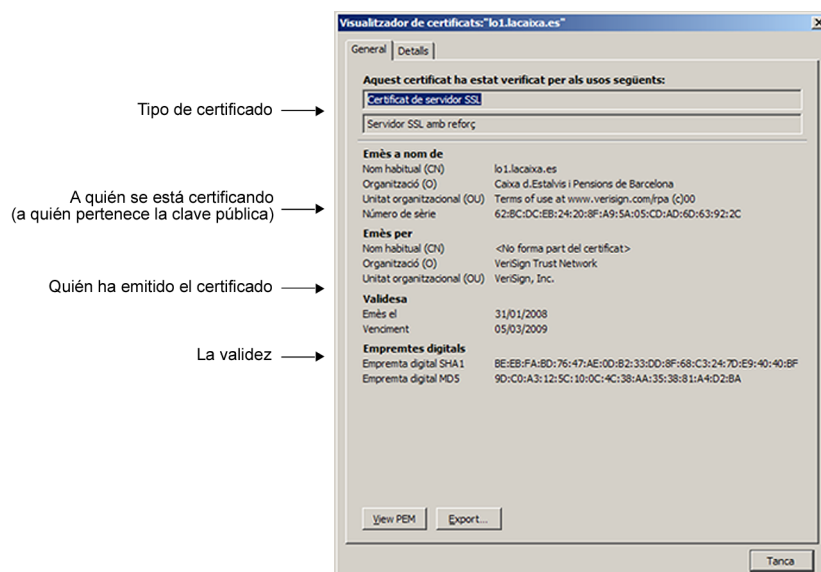
Candado en el navegador web

lo2.lacaixa.es 

El candado en el navegador web indica que se está utilizando una conexión que garantiza la seguridad de los datos.

El navegador también permite visualizar la información que contiene el certificado que nos ha enviado el servicio web. En la siguiente figura aparece la información que nos proporciona el navegador web.

Información sobre el certificado que muestra el navegador web



Durante el proceso de conexión, el servidor envía un certificado. Éste contendrá, entre otros elementos, una clave pública y la autoridad de certificación emisora del certificado. Si nuestro software ya confía en esta autoridad, el proceso seguirá adelante. Si, por el contrario, no confía en ella (no se trata de una autoridad sobradamente reconocida), el navegador nos preguntará sobre si queremos seguir o no con el proceso de conexión segura.

### Seguridad electrónica

La seguridad que ofrecen los sistemas de comercio electrónico a la hora de efectuar los pagos es superior a la de los pagos tradicionales. En el comercio electrónico, el usuario

controla el uso de su tarjeta y los datos que introduce viajan de manera segura hacia el servicio en línea del banco. En cambio, cuando damos la tarjeta a cualquiera no sabemos cuántas veces la pasará o si hará duplicados.

Si se acepta la clave pública, ésta se utiliza para generar lo que se llama una **clave de sesión**. Recordamos que los sistemas de cifrado de clave pública son más lentos que los de clave compartida. Por lo tanto, se elige una clave compartida que será la clave de sesión que se utilizará para cifrar de manera rápida las comunicaciones (los datagramas) que se irán enviando dentro de la conexión segura.

### El pago con tarjeta de crédito

Sin embargo, si la autenticación del servidor y el cifrado de los datos de pago están perfectamente resueltos, la autenticación del cliente no es bastante segura frente a usos ilícitos de la tarjeta de crédito. Aunque cada vez es más habitual que se utilice el teléfono móvil para enviar mensajes de confirmación y que el servicio de pago pueda comprobar que quien efectúa el pago es realmente el poseedor de la tarjeta de pago. Aunque esto no es bastante seguro ante un robo de bolso, por el que el ladrón podrá acceder a la tarjeta de crédito y al móvil.

## 4.2. Correo electrónico seguro

El correo electrónico seguro permite enviar mensajes firmados y/o cifrados. El sistema S/MIME es una extensión de MIME que permite la firma y el cifrado de mensajes. La práctica totalidad de gestores de correo electrónico permiten estas funcionalidades. Vamos a ver qué implica el envío de correo electrónico seguro:

- Para **enviar un mensaje firmado**, en el mismo correo electrónico se envía la firma del mensaje y el certificado que autentica nuestra clave pública. Así pues, el software receptor del mensaje accederá a nuestra clave pública, incluida dentro del propio mensaje, y podrá verificar la validez gracias al certificado también incluido en el mensaje. Se usará la clave pública para verificar la firma digital del mensaje.
- Para **enviar un mensaje cifrado**, será necesaria la clave pública del destinatario. Para obtenerla, será suficiente con pedirle esta clave. Si nos envía la clave (junto con el certificado), la podremos instalar en el equipo para cifrar el mensaje y también para usos posteriores.

El sistema S/MIME utiliza la técnica del **sobre digital** para enviar un mensaje cifrado. El sobre digital que se muestra en la siguiente figura consiste en:

- 1) Elegir aleatoriamente una clave de sesión que se utilizará para cifrar el mensaje utilizando un sistema de cifrado simétrico.
- 2) Adjuntar, con el propio mensaje firmado, la clave de sesión cifrada para cada uno de los diferentes destinatarios que pueda tener el mensaje. Así pues, en el caso de la figura, en la que hay que enviar un mismo mensaje a dos

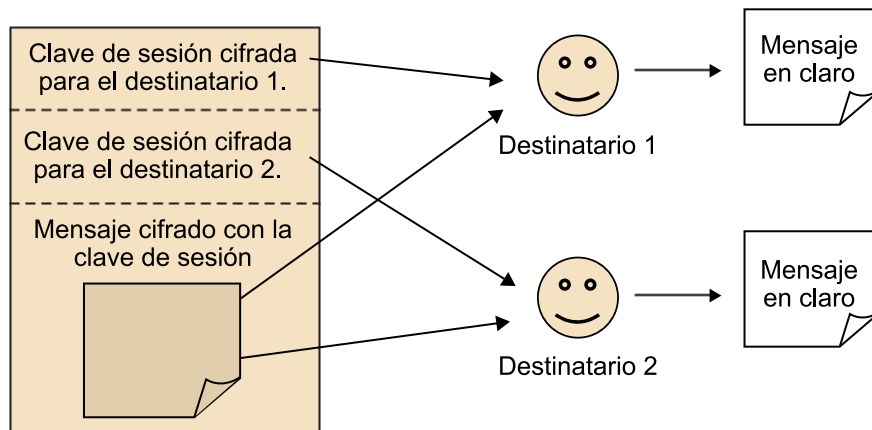
### Correo electrónico seguro

Si previamente recibimos un mensaje firmado por el que ahora será destinatario del mensaje cifrado, dispondremos de su clave pública en nuestro sistema.



usuarios diferentes, añadiremos dos versiones de la clave de sesión: una cifrada con la clave pública del destinatario 1 y otra cifrada con la clave pública del destinatario 2.

Estructura de un mensaje cifrado para dos destinatarios



De esta manera, podemos enviar un único mensaje cifrado para varios destinatarios y, al mismo tiempo, evitamos el coste computacional que implicaría cifrar todo el mensaje (ficheros adjuntos incluidos) con un sistema de clave pública.

### 4.3. Trámites en línea

Los trámites en línea deberían dar más o menos garantías de seguridad en función del riesgo que suponen.

Actualmente, hay trámites en línea que piden un registro al servicio a partir de determinados datos y los hay que piden específicamente que el usuario esté autenticado con un certificado de clave pública. Toda esta heterogeneidad de sistemas de autenticación para los trámites en línea se soluciona por medio de la utilización de la identidad digital ciudadana.

La **identidad digital ciudadana** permite que el usuario esté debidamente autenticado ante las administraciones para realizar cualquier tipo de trámite en línea.

Así como el Estado es quien controla la identidad de sus habitantes, al asignarles un número que los identifica, también puede generar una clave pública con su correspondiente certificado a fin de que se identifique ante los diferentes servicios de Internet.

#### Ejemplo de garantía de seguridad

La obtención de un listado de productos solicitados en una tienda en línea supone un riesgo más bajo que pedir el cambio de datos bancarios para el cobro de la nómina.

#### Ejemplo de registro al servicio

El importe de la factura telefónica para el caso de trámites en línea con la compañía telefónica es un ejemplo de trámite que pide un registro al servicio.

El certificado se encuentra almacenado en un dispositivo seguro que se introduce en el sistema informático del usuario cuando el servicio en línea pide la identificación. Para garantizar más seguridad, el acceso al dispositivo seguro debería hacerse siempre que el usuario introduzca correctamente una contraseña.

### El DNI electrónico español

En el caso del DNI electrónico para el Estado español, el dispositivo seguro es una tarjeta inteligente (en inglés, *smartcard*). Dentro de la tarjeta hay un chip de seguridad que contiene información como claves privadas, claves públicas y certificados.

DNI electrónico para el Estado español

Contactos de acceso al chip



La realización de trámites en línea supone la generación de documentos electrónicos a los que se les puede pedir aplicar una firma electrónica antes del envío mediante el servicio de Internet.

## 5. Privacidad e Internet

Es evidente que hoy en día millones de personas de diferentes edades y niveles educativos utilizan Internet. Como veremos en el módulo "La World Wide Web", estos usuarios, que o bien pueden ser personas con conocimientos mínimos sobre tecnología o bien personas muy expertas, no sólo acceden a Internet para obtener información, sino también para comunicarse y, últimamente, contribuir a añadir información.

Las tecnologías de Internet permiten registrar la actividad diaria de los usuarios, de manera que se puede analizar la información y se pueden llegar a crear perfiles de usuarios: hábitos de búsqueda en los buscadores, historial de compras realizadas, páginas web frecuentadas, etc. Estos perfiles pueden ser usados por los buscadores, para que puedan dar resultados que se ajusten mejor a la personalidad del usuario, como mostrarle anuncios personalizados cuando esté en webs que muestren propaganda, etc. Sea como sea, lo cierto es que por medio de los perfiles hay alguien que podrá saber cómo es nuestra actividad en Internet, lo que en principio invade la privacidad de los usuarios que la utilizamos.

En los ejemplos anteriores, los usuarios quizás no son conscientes de que su privacidad puede ser invadida, ya que utilizar un ordenador en privado da la sensación que no estás siendo controlado.

Aparte de la información privada que se puede obtener de un usuario mediante un virus espía, existen otras maneras de obtener información. En lo que queda de esta parte sobre privacidad, trataremos la privacidad en la navegación web, el tema de la privacidad en las redes sociales y la obtención de datos privados por medio de la ingeniería social.

### 5.1. Privacidad en la navegación web

Cuando visitamos una página web, ésta puede reunir una gran cantidad de información sobre quién accede y con qué frecuencia lo hace. Esta monitorización se realiza mediante las llamadas galletas<sup>8</sup>.

<sup>(8)</sup>En inglés, *cookies*.

Una **galleta** es un pequeño fichero de texto que puede enviar el servicio web cuando un usuario se conecta. Es el único fichero que un servidor web remoto puede escribir en nuestro equipo.

Las galletas se suelen utilizar para recordar la identidad de un usuario que ya ha visitado un servicio web. La primera vez que se conecta, el servicio web comprueba si en el equipo del usuario hay una galleta de este servicio web. Si no es el caso, se pide al usuario la creación de una cuenta de acceso (es decir, un nombre de usuario y una contraseña). Una vez registrado, el servicio web envía una galleta con la identidad del usuario. Cuando en otro momento el usuario vuelva a acceder al mismo servicio web, éste ya encontrará una galleta, con lo que se habrá comprobado la identidad del usuario y se evitará que éste deba introducir de nuevo el nombre de usuario.

Sin embargo, mediante las galletas, el proveedor de un servicio web puede saber con qué frecuencia el servicio es utilizado por cada uno de sus usuarios.

Estos usos de las galletas suponen un riesgo leve para la privacidad del usuario. Esto cambia en el caso de las **galletas de rastreo**<sup>9</sup>. Estas galletas tienen como objetivo rastrear el comportamiento de un usuario a medida que utiliza varios servicios web. Esta información queda recopilada en el sistema de información de quien gestiona estas galletas.

<sup>(9)</sup>En inglés, *tracking cookies*.

Una tercera entidad, dedicada a hacer publicidad mediante Internet, gestionará estas galletas.

### Ejemplo de galletas de rastreo

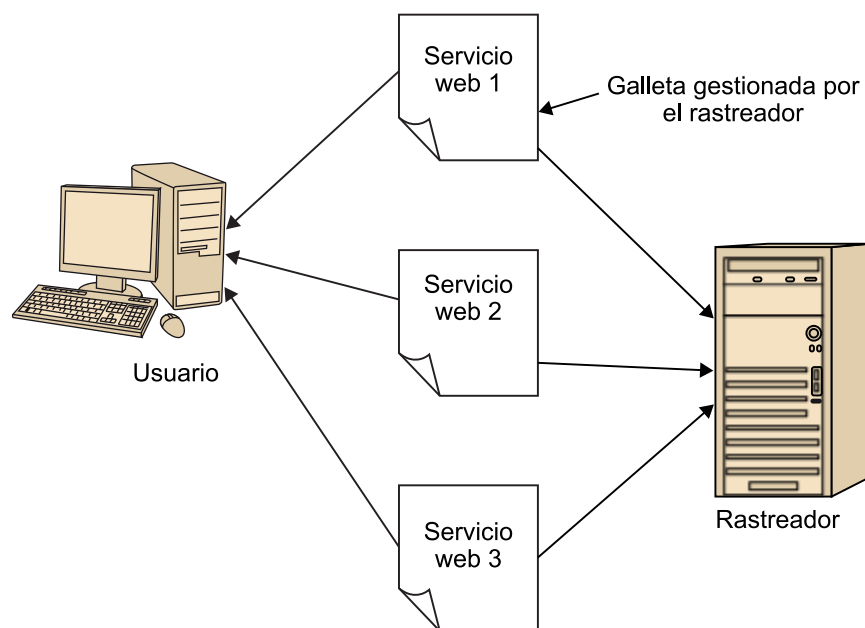
Si realizamos búsquedas sobre motocicletas, buscamos motocicletas antiguas en los servicios webs de subastas, etc., esta tercera parte deducirá que nos gustan las motocicletas. Así pues, cuando vamos a un servicio web que muestre publicidad, la entidad gestora de las galletas de rastreo indicará que la publicidad que se nos debe mostrar tenga algo que ver con las motocicletas.

Hay quien ve las galletas de rastreo como una especie de software espía, ya que la entidad que las gestiona toma conciencia de los servicios web que visitamos, qué hacemos y con qué frecuencia lo hacemos.

### Los programas de limpieza

Los programas de limpieza también detectan y eliminan estas galletas de rastreo de nuestro sistema.

### Rastreo de un usuario durante la visita a tres servicios web diferentes



En este ejemplo, el usuario se va conectando a diferentes servicios web que contienen galletas gestionadas por una tercera parte. Esta toma nota de cuándo el usuario visita estos servicios web y, en consecuencia, sabe por dónde ha navegado y con qué frecuencia.

#### 5.1.1. El gran hermano en Internet

Si bien son muchas las compañías que se dedican a producir software para ordenadores personales e Internet, lo cierto es que sólo dos o tres de ellas acaparan la gran parte del pastel. Por ejemplo, una única compañía es responsable del sistema operativo y el software ofimático que millones de usuarios utilizan habitualmente. Por otra parte, los servicios populares en Internet de búsqueda de información se han convertido en mucho más que eso: hoy en día, sus servicios incorporan herramientas complementarias, como búsquedas en los documentos del ordenador personal, edición en línea de documentos, gestión de citas y calendarios, etc.

Por otra parte, millones de usuarios utilizan a los grandes gestores de correo electrónico de estas compañías a los que acceden vía web. Por lo tanto, su correo electrónico enviado y recibido se guarda permanentemente en servidores compartidos por millones de usuarios en lugar de tenerlos guardados en el disco del ordenador personal. Las tecnologías de la información permiten analizar estos correos electrónicos, no sólo para localizar terroristas y usuarios potencialmente peligrosos, sino también para crear perfiles que ayuden a los buscadores de las propias compañías, entre otras posibilidades.

De todo ello se desprende que un par de grandes compañías tienen acceso a información privada de millones de usuarios de Internet. De manera pública, se sabe que estas compañías reúnen cierta información de sus usuarios para fines relacionados con ofrecer a los usuarios un servicio "más personalizado".

Sin embargo, si esta enorme cantidad de información se utilizara de manera deshonesta, la figura del Gran Hermano de la novela de George Orwell se podría acabar convirtiendo en una realidad.

## 5.2. Las redes sociales

Los servicios de redes sociales permiten que la gente se conecte mediante una comunidad en línea. Existen diferentes tipos de redes sociales, desde aquellas que agrupan usuarios con aficiones muy concretas hasta amplias redes sociales en las que el objetivo es estar conectado con un gran grupo de amigos.

Las grandes redes sociales implican una serie de riesgos asociados a la privacidad. En estos servicios en línea, los usuarios publican su perfil: describen cómo son, indican su fecha de nacimiento, incluyen una fotografía e incluso llegan a poner la dirección personal y el número de teléfono. También se pueden especificar las tendencias políticas, creencias religiosas y la orientación sexual.

### Las redes sociales

Las redes sociales más populares llegan a rebasar los 200 millones de usuarios cada una.

Esta información, de carácter claramente personal, es accesible a los usuarios de la red social, de acuerdo con determinadas restricciones. El usuario final, es decir, quien ha introducido la información de su perfil, es quien decide quién puede acceder a esta información. Por ejemplo, se puede dar el caso de que esta información sea accesible a todos los usuarios de la red social, con los riesgos de privacidad que ello podría implicar. En el caso de querer ser más restrictivos, los usuarios pueden especificar que su información sólo sea accesible por sus amigos. En general, para ser amigo sólo es necesario localizar a un usuario concreto dentro de la red social y enviarle una petición de amistad. Si éste acepta, pasará a formar parte del grueso de amigos.

Sin embargo, y aunque el propio usuario controle quién puede acceder a su información, las redes sociales siguen poniendo en riesgo la privacidad del usuario:

- Por ejemplo, en estas redes sociales se pueden poner fotos. Imaginemos que algún amigo nuestro pone una foto donde, de rebote, aparecemos nosotros en una situación que consideramos comprometida. Además, el sistema permite "etiquetar" a las personas que salen en las fotografías, de manera que si alguien busca nuestra identidad dentro de la red social podría llegar a tener acceso a la fotografía comprometida.
- Por otra parte, un usuario deshonesto podría llegar a crear un perfil falso con nuestro nombre, para hacerse pasar por nosotros o por una persona muy diferente de la que es realmente.

Las políticas de privacidad de estas redes sociales avisan a los usuarios de que la responsabilidad de controlar la información y, de rebote, el riesgo en el que ponen a su privacidad recaen exclusivamente en ellos.

### 5.3. El *phishing*

En el primer apartado de este módulo, hemos descrito en qué consisten los ataques por medio de la ingeniería social. Un caso concreto de la ingeniería social lo hemos ejemplarizado con la recepción de un mensaje de correo electrónico, supuestamente de nuestra entidad bancaria, en el que se nos pide introducir las credenciales de acceso al gestor bancario.

Este caso específico de ingeniería social recibe el nombre de *phishing*, que en español podríamos traducir como "ir de pesca".

El *phishing* es la técnica que consiste en suplantar la identidad electrónica de una organización determinada con el objetivo de convencer a alguien para que revele información confidencial que posteriormente será utilizada con finalidades fraudulentas.

Si el usuario hace caso del mensaje de correo electrónico, irá a la página web de la supuesta entidad bancaria. Conviene darse cuenta de un par de cosas:

- En algunas ocasiones, la dirección en la que se ubica el servicio web no se parece a la real. Por ejemplo, se nos abre el servicio web de `http://mibanco.dangerwebs.ru` en lugar de `http://www.mibanco.com`.
- El servicio web en cuestión no está protegido, es decir, no se verá el candado que indica conexión bajo SSL. También podría pasar que estuviera certificada por una autoridad de certificación que el software de nuestro equipo no reconoce.
- La página está a menudo llena de errores de ortografía o cuando menos de expresiones extrañas, ya que suele estar diseñada por gente de otros países.

Que los ataques de *phishing* tengan éxito depende, en gran medida, de las posibilidades de que el usuario crea que el mensaje es auténtico. Hoy en día existe un conocimiento extendido sobre este tipo de ataques, y es evidente que se desconfiará de un mensaje de esta clase cuando, sobre todo, provenga de una entidad bancaria que no es la nuestra.

## 6. Seguridad en la gestión audiovisual

La entrada de la producción audiovisual (música y cine) en el mundo digital y de Internet se ha traducido en la difusión masiva ilegal de obras, o bien mediante copias directas, en CD o DVD, o bien por Internet. En consecuencia, se han ido desarrollando y utilizando diferentes sistemas cuyo objetivo es dar seguridad a la gestión audiovisual. Esta seguridad es importante tanto para el vendedor como para el comprador del material audiovisual:

- Al vendedor le interesa que el contenido que vende no se pueda copiar sin permiso. También le interesa controlar las copias que ha vendido.
- El comprador busca poder disfrutar del contenido que ha comprado tantas veces como convenga. Por otra parte, conviene evitar que un usuario deshonesto pueda hacerse con el contenido que ha comprado otro y lo distribuya haciéndose pasar por el comprador auténtico.

Las técnicas que se han ido desarrollando se pueden clasificar en dos grupos: las técnicas de control de reproducción y copia y las técnicas de marcaje. Las veremos en este último apartado del módulo.

### 6.1. Sistemas de control de copia y reproducción

El impedimento de copia es un tema tratado desde el inicio de la distribución de audio y vídeo al público general. Los sistemas de vídeo con cinta analógica disponen de un sistema de copia con el que cualquier copia que se realiza de una cinta original (conectando dos vídeos, uno reproduciendo la cinta y el otro grabándola) presenta errores de sincronía que se traducen en una muy mala calidad de imagen y pérdidas del sonido.

Con la llegada de sistemas digitales de audio y vídeo reproducibles sin pérdida de calidad en cualquier ordenador personal, se han ido utilizando diferentes métodos de impedimento de copia. En general, la idea consiste en pervertir el contenido digital que contiene el CD o el DVD, de manera que éste continúe siendo visible en un reproductor doméstico, a la vez que no pueda ser reproducible en un ordenador personal: el contenido erróneo genera errores en el software y detiene el proceso de copia. Sin embargo, cuando el sistema anticopia lleva cierto tiempo en el mercado, siempre hay alguien que crea un software de copia capaz de saltarse esa protección.

#### Copias analógicas

Los sistemas de protección mencionados no tienen en cuenta que el usuario puede realizar una copia analógica del contenido para después volver a pasarlo al ordenador, para



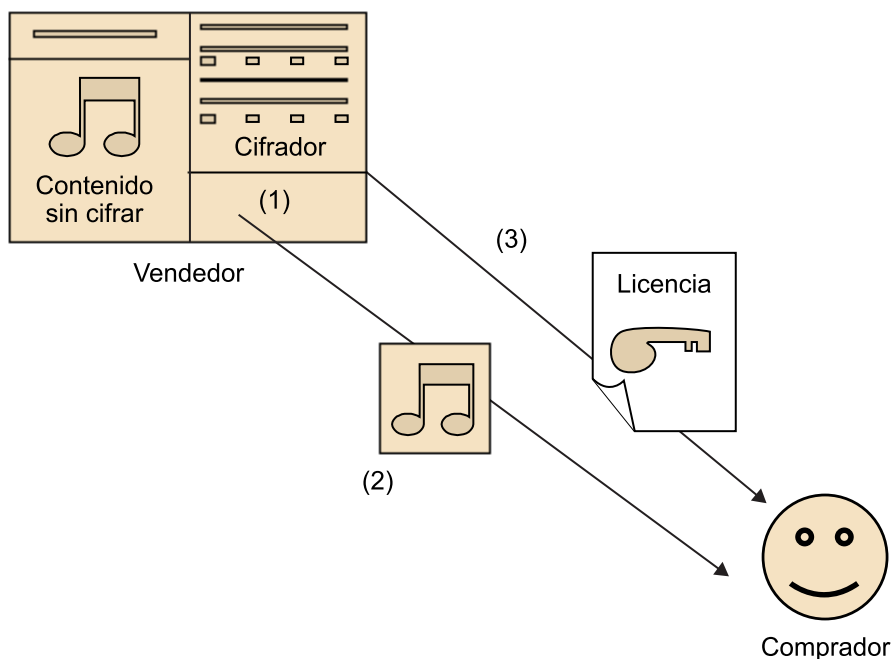
saltarse la protección. A pesar de ser posible, se produce una pérdida de calidad notoria al pasar de digital a analógico.

### Protección de los DVD

De hecho, todos los DVD que encontramos en venta están cifrados. El problema es que los reproductores de DVD contienen las claves que permiten descifrarlos y, en definitiva, ello ha permitido crear software que descifra y copia el contenido de un vídeo en DVD.

Una alternativa interesante consiste en cifrar el contenido y vender la clave de descifrado al comprador del contenido. Esto sólo es factible en contenido comprado mediante Internet y no en los CD o DVD. Lo ilustramos en la siguiente figura.

Ejemplo de protección del contenido por medio de cifrado y licencias



Cuando un comprador adquiere, por ejemplo, una canción, el vendedor realiza una copia y la cifra con una clave secreta (1). El cliente se puede descargar la versión cifrada (2). Cuando éste efectúa el pago, se descarga **una licencia** (3) que contiene la clave de descifrado. Así pues, el software reproductor será capaz de descifrar el contenido y, en consecuencia, hacer sonar la canción las veces que se quiera.

El sistema de licencias es el más extendido, aunque siempre hay algún programa que acaba saltándose estas protecciones y descifrando el contenido a base de ataques específicos para obtener la clave de manera deshonesto.

## 6.2. Sistemas de marcaje

Los sistemas de marcaje pueden llegar a ser una alternativa al control de copia y reproducción. A continuación, estudiaremos en qué consisten las marcas de agua en el contenido audiovisual y veremos algunos ejemplos concretos de su uso.

### 6.2.1. Las marcas de agua

El marcaje de agua digital<sup>(10)</sup> es la técnica que permite ocultar un mensaje (una tira de bits) en un contenido audiovisual. Por ejemplo, un fotógrafo profesional podría disponer de un software de marcas de agua para esconder en sus fotografías un mensaje en el que se indique quién es el autor de la fotografía. Una vez que la fotografía está marcada, ya está lista para vender.

<sup>(10)</sup>En inglés, *digital watermarking*.

Si resulta que el fotógrafo encuentra una fotografía suya en una página web que no tiene los permisos de explotación de ésta, el fotógrafo podría demostrar que él es el autor de la fotografía.

Para que esto sea factible, conviene cumplir una serie de factores:

- El software que oculta la marca de agua utiliza una clave privada. Sólo con esta clave privada se podría recuperar correctamente el mensaje; por lo tanto, el fotógrafo (el único poseedor de esta clave) es el único que puede recuperar la marca.
- El sistema de marcaje debe ser robusto, es decir, debe permitir que la marca se pueda seguir recuperando a pesar de que la fotografía se haga más pequeña, se cambien ligeramente los colores, se recorte una parte, etc. Evidentemente, si no se puede recuperar la marca es porque la fotografía ha quedado tan estropeada que ya no tiene interés comercial.

Aunque hayamos dado el ejemplo con fotografías, lo cierto es que existen sistemas de marcaje bastante consistentes no sólo para imágenes, sino también para música y vídeos.

### 6.2.2. Las huellas digitales

En el caso anterior, todas las copias vendidas de la misma fotografía llevaban escondido el mismo del mensaje: "El autor de la fotografía es...". Sin embargo, si escondemos el nombre del comprador en lugar del nombre del autor, estamos usando la técnica que se llama **huella digital**<sup>(11)</sup>.

<sup>(11)</sup>En inglés, *digital fingerprint*.

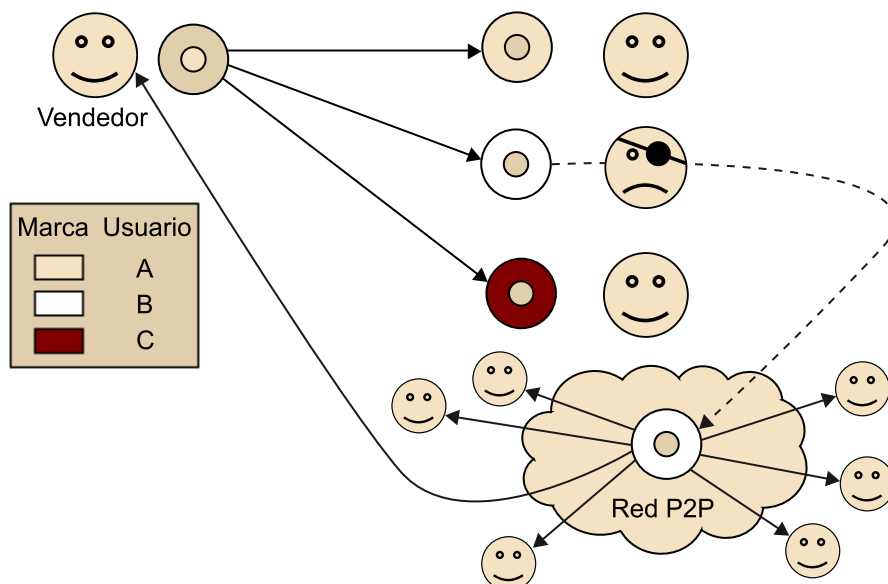
Cambemos de ejemplo ahora para ilustrar el funcionamiento de las huellas digitales. Supongamos que un grupo de música conocido ha terminado la grabación de su nuevo disco. Su sello discográfico quiere promocionar las nuevas canciones un mes antes de poner el disco en venta y, por lo tanto, envía 50 copias del nuevo CD a emisoras de radio, críticos musicales, etc.

Al utilizar la huella digital, se harán un total de 50 copias diferentes del CD. Cada una de ellas tendrá una marca de agua única indicando a quién se entrega la copia promocional.

Resulta que en un par de semanas ya se puede descargar de Internet el nuevo disco, antes de que haya salido en venta; paralelamente, las paradas "top manta" ya están vendiendo copias de los CD. Pues bien, recuperando la marca de agua del contenido pirateado se sabrá quién ha sido el que ha iniciado la distribución ilegal del CD.

En la siguiente figura, se ilustra este hecho: el distribuidor, por medio de la recuperación de la marca, es capaz de saber quién ha puesto el contenido musical en la red de distribución. El vendedor tiene una relación de la marca que se ha puesto en cada una de las copias distribuidas. Uno de estos usuarios pone el contenido en una red P2P. El vendedor, que encuentra el contenido en la red P2P, recupera la marca y sabe, en principio, quién es el responsable de su distribución ilegal.

Ejemplo de identificación del distribuidor ilegal por medio de huellas digitales



### Redes P2P

Las redes P2P (*Peer to peer*, de igual a igual) se diseñaron para el intercambio de archivos. Actualmente, la mayoría de archivos que se intercambian tienen *copyright* que no lo permite, por lo que el intercambio de estos archivos en concreto (por ejemplo, música y películas) es de legalidad dudosa.

Este sistema tiene algunos problemas que pasamos a comentar a continuación:

- El contenido puede ser robado del destinatario original y ser puesto en circulación de manera ilegal. El "culpable" sería el destinatario original, lo que se revelaría en la recuperación de la marca.
- Como cada copia es diferente (precisamente hay diferencia en los bits que sirven para esconder la marca), dos destinatarios podrían comparar sus copias en calidad de bits y así detectar qué bits forman parte de la marca. Entonces cambiarían el valor de estos bits por valores aleatorios, para intentar borrar las marcas que identificarían a los dos destinatarios deshonestos.

A pesar de estos problemas, que cuentan con algunas soluciones que se pueden llevar a la práctica perfectamente, los sistemas de huella digital se están utilizando de manera cada vez más generalizada, por ejemplo en el caso de las copias promocionales.

## Resumen

En este módulo, hemos tratado los temas más importantes relacionados con la seguridad dentro de las tecnologías de la información y las comunicaciones.

En la primera parte, hemos presentado la variedad de ataques posibles que se pueden producir, así como también hemos definido qué es un atacante.

La segunda parte la hemos dedicado a revisar qué tipo de ataques pueden sufrir los equipos informáticos. Hemos visto el problema que causan los virus y sus variantes. Hemos descrito qué son y qué implican los ataques de acceso desde Internet. También hemos visto que los antivirus, los programas de limpieza y los cortafuegos son herramientas fundamentales que hay que tener en el sistema para garantizar, al menos teóricamente, su resistencia a los ataques.

Asimismo, hemos hablado de la seguridad en la información y hemos definido propiedades importantes, como la confidencialidad y la autenticidad. Hemos llevado a cabo una aproximación a los sistemas que permiten garantizar estas propiedades, desde sistemas sencillos de criptografía clásica, que nos han servido para comprender algunos conceptos básicos, hasta las técnicas de firma digital.

Los sistemas de clave pública están estrechamente ligados a lo que hemos llamado identidad digital, gestionada por los certificados y las infraestructuras de clave pública. Hemos visto tres casos de aplicación de las técnicas estudiadas y de la identidad digital: el comercio electrónico, el correo electrónico y los trámites en línea.

La penúltima parte del módulo la hemos dedicado a hablar de un tema relacionado con la seguridad: la privacidad de los usuarios de las tecnologías que sustentan la sociedad de la información. Hemos apuntado cómo los servicios de Internet pueden rastrear el comportamiento de sus usuarios. También hemos reflexionado sobre el hecho de que dos o tres grandes compañías pueden llegar a tener un gran control sobre las actividades que llevan a cabo los usuarios de ordenadores y de Internet.

Finalmente, hemos acabado el contenido teórico del módulo tratando el tema de la seguridad en la gestión audiovisual, donde hemos presentado sistemas basados en impedimento de copia o control de la reproducción y sistemas de marcaje.



## Actividades

1. Buscad información en Internet y explicad qué ventajas puede suponer a la ciudadanía el uso del DNI electrónico.
2. El sistema PGP (*Pretty Good Privacy*, en español, privacidad bastante buena) es un sistema de herramientas de clave pública para gran variedad de aplicaciones: cifrado y firma de correo electrónico, de ficheros, etc. A pesar de todo, su sistema no utiliza certificados de "clave pública" a la hora de garantizar la validez de las claves públicas. Averiguad qué sistema se utiliza y diferenciadlo de lo que hemos visto en el módulo (infraestructuras de clave pública).
3. Conectaos a diferentes portales web seguros (por ejemplo, banca en línea) y mirad qué información tiene el certificado de identidad de servidor (en principio, podéis hacer doble clic en el candado que aparece en el navegador para obtener información).
4. Buscad en las hemerotecas digitales artículos sobre las redes sociales y sus efectos sobre la privacidad de los usuarios. Leedlos y comparadlos con lo que hemos explicado en este módulo.
5. Id a la página web de algún fabricante de antivirus y buscad información sobre cuántos virus diferentes identifican sus productos.

## Ejercicios de autoevaluación

1. Diferenciad los sistemas de criptografía simétrica y asimétrica.
2. Se utiliza el siguiente sistema de autenticación de mensajes entre los usuarios  $A$  y  $B$ : para enviar un mensaje  $M$ , se envía  $\{M, h(M)\}$ , donde  $h$  es la función de resumen SHA-1 de 160 bits de salida. Responded, justificándolo, si se garantiza la detección de que alguien haya modificado el mensaje.
3. Suponed que queréis enviar un mensaje de correo electrónico de manera que el contenido de éste sea secreto para todo el mundo menos para mí y para ti. Explicad:
  - a) ¿Cómo se haría con un sistema de clave simétrica y qué claves necesitaríais?
  - b) ¿Cómo se haría con un sistema de clave pública y qué claves necesitaríais?
4. Ahora quiero enviar un único fichero mensaje dirigido a tres usuarios y que esté cifrado. ¿Cómo lo puedo hacer enviando únicamente un fichero cifrado?
5. Describid qué aplicativos del sistema suelen tener abierto el acceso a Internet mediante el cortafuegos.
6. En función de lo que hemos visto en el módulo, decid por qué la reproducción con licencias no es extensible a los CD. ¿Y lo son los sistemas de huella digital?
7. Una discográfica prepara 50 copias promocionales de un CD que enviará a las radios y a los críticos de música antes de que éste CD se ponga a la venta. A los dos días, una copia en MP3 de uno de los singles se encuentra en eMule. Indicad cómo se habrá podido utilizar un sistema de huella digital para detectar quién (en teoría) ha puesto este contenido en el sistema de distribución P2P.

## Solucionario

### Ejercicios de autoevaluación

1. En la criptografía simétrica se utiliza la misma clave para cifrar y para descifrar; es necesaria una clave para cada pareja posible de comunicantes; comunicar la clave compartida al otro participante en la comunicación puede implicar un problema; los algoritmos son más rápidos de ejecutar para un sistema informático. En la criptografía asimétrica, la clave secreta se utiliza para descifrar/firmar, mientras que la clave pública se emplea para cifrar/comprobar la firma; sólo es necesario un par de claves para cada usuario (la pública y la privada); todo el mundo puede acceder a la clave pública, por lo tanto, su distribución no supone un problema; los algoritmos son menos rápidos de ejecutar.
2. El sistema no garantiza nada: cualquiera que tenga acceso al  $\{M, h(M)\}$  puede elegir el mensaje  $M$ , modificarlo, calcular el resumen haciendo  $h(M)$  y reenviar la información como si nada hubiera pasado.
3. En el caso de clave simétrica, o bien habría que ponerse de acuerdo con una clave común, o bien que eligiera yo una y hacértela saber. Con esta clave cifraría el mensaje y te lo enviaría. Tú utilizarías la clave que te he enviado o hemos consensuado para descifrar el mensaje. En el caso de clave pública, si yo te envío el mensaje a ti, deberé utilizar tu clave pública para cifrarlo, mientras que tú utilizarás la correspondiente clave privada para descifrarlo una vez que lo hayas recibido.
4. Utilizaremos el sobre digital. En primer lugar, el software de cifrado elegirá una clave para cifrar simétricamente el fichero sólo una vez. En el fichero resultante se adjuntará la clave simétrica elegida cifrada por cada uno de los tres destinatarios, es decir, con sus claves públicas.
5. Deberán tener el camino abierto por el cortafuegos los aplicativos más habituales: el navegador web, el gestor de correo electrónico, la mensajería instantánea, etc.
6. La reproducción con licencias implica que el contenido vendido se cifrará de manera específica al usuario. Esto va en contra del proceso de fabricación del CD, mediante el que se realiza un elevado número de copias. Un sistema factible sería cifrar el CD con una clave  $k$  y que para obtener esta clave  $k$  hubiera que comprar una licencia, pero nos encontraríamos con que esta clave circulará por las páginas web de piratería. Lo mismo pasa con los sistemas de huella digital, en los que cada CD se debería marcar de manera específica y, además, habría que gestionar las relaciones entre marca y usuario. Sin embargo, hemos visto que estas técnicas se utilizan para las copias promocionales.
7. Cada copia debería tener incrustada una marca que identificara a quién se ha distribuido el contenido (huella digital o *fingerprint*). Si el distribuidor discográfico encuentra el contenido en una red de distribución, podrá recuperar la marca y saber, en teoría, cuál de los 50 destinatarios lo ha puesto.



## Glosario

**certificado de clave pública** *m* Documento electrónico que relaciona una clave pública con una identidad, como una dirección web o de correo electrónico.

**clave pública** *f* Clave que puede ser puesta a disposición de todos los usuarios que quieran utilizar un criptosistema de clave pública para asegurar la información con su propietario.

**cortafuegos** *m* Herramienta que controla el tráfico que circula entre dos redes, o entre un ordenador y la Red.

**cracker** *m* Hacker que se dedicará a estropear los sistemas que ataca.

**denegación de servicio** *f* Ataque que tiene éxito cuando el servicio atacado es incapaz de dar servicio a sus usuarios.

**firma digital** *f* Técnica que garantiza que un mensaje es auténtico, es decir, que viene de donde dice venir y del que no se ha alterado el contenido.

**función de resumen** *f* Algoritmo que, dado un mensaje de longitud arbitraria, devuelve una tira de bits de tamaño fijo.

**galleta** *f* Fichero de texto que los servidores web pueden dejar dentro de los equipos informáticos cuando se visitan las webs que alojan.

**hacker** *m* Delincuente informático. Por otra parte, define a alguien con elevados conocimientos de informática y seguridad.

**identidad digital** *f* Certificado emitido por la Administración que identifica a los ciudadanos y al mismo tiempo les proporciona una clave pública para poder llevar a cabo trámites en línea.

**ingeniería social** *f* Técnica que permite obtener información confidencial de los usuarios persuadiéndolos.

**marca de agua** *m* Mensaje en forma de tira de bits que se inserta en un fichero sin que se perciba.

**red social** *f* Servicio basado en las nuevas tecnologías web que permiten a usuarios establecer relaciones, comunicarse y compartir información.

**secuestrador de sistema** *m* Software maligno que cambia las propiedades del sistema y su comportamiento.

## Bibliografía

**Gutiérrez, J. D. et al.** (2008). *Seguridad en Redes Locales (Guía Práctica)*. Anaya Multimedia Interactiva.

**Herrera J. y otros** (2003). *Tecnología del comercio electrónico*. Barcelona: Editorial UOC.

**Stallings, W.** (2003). *Fundamentos de Seguridad en Redes*. Pearson Educación.

# La World Wide Web

Gregorio Robles Martínez

PID\_00150271



Universitat Oberta  
de Catalunya

[www.uoc.edu](http://www.uoc.edu)



Los textos e imágenes publicados en esta obra están sujetos –excepto que se indique lo contrario– a una licencia de Reconocimiento-NoComercial-SinObraDerivada (BY-NC-ND) v.3.0 España de Creative Commons. Podéis copiarlos, distribuirlos y transmitirlos públicamente siempre que citéis el autor y la fuente (FUOC. Fundació per la Universitat Oberta de Catalunya), no hagáis de ellos un uso comercial y ni obra derivada. La licencia completa se puede consultar en <http://creativecommons.org/licenses/by-nc-nd/3.0/es/legalcode.es>

# Índice

|                                                                  |           |
|------------------------------------------------------------------|-----------|
| <b>Introducción.....</b>                                         | <b>5</b>  |
| <b>Objetivos.....</b>                                            | <b>6</b>  |
| <b>1. Historia.....</b>                                          | <b>7</b>  |
| 1.1. La web, el servicio estrella de Internet .....              | 7         |
| <b>2. HTTP: cliente y servidor.....</b>                          | <b>9</b>  |
| 2.1. El servidor .....                                           | 9         |
| 2.2. El cliente .....                                            | 12        |
| <b>3. El protocolo HTTP.....</b>                                 | <b>16</b> |
| 3.1. Un ejemplo de comunicación con HTTP .....                   | 18        |
| <b>4. El lenguaje HTML. Nociones básicas.....</b>                | <b>20</b> |
| 4.1. Un ejemplo de página web .....                              | 22        |
| <b>5. La web dinámica o interactiva y aplicaciones web.....</b>  | <b>26</b> |
| 5.1. Buscadores .....                                            | 28        |
| <b>6. Web 2.0.....</b>                                           | <b>31</b> |
| 6.1. Páginas wiki .....                                          | 32        |
| 6.2. Blogs .....                                                 | 35        |
| 6.3. Suites ofimáticas en línea .....                            | 39        |
| 6.4. Otros sistemas web 2.0 .....                                | 40        |
| <b>7. La web multimedia.....</b>                                 | <b>42</b> |
| 7.1. Imágenes .....                                              | 42        |
| 7.2. Audio .....                                                 | 44        |
| 7.3. Vídeo digital y la compresión de datos .....                | 46        |
| 7.4. Sistemas de compartición de imágenes, audios y vídeos ..... | 49        |
| <b>Resumen.....</b>                                              | <b>50</b> |
| <b>Actividades.....</b>                                          | <b>51</b> |
| <b>Ejercicios de autoevaluación.....</b>                         | <b>51</b> |
| <b>Solucionario.....</b>                                         | <b>52</b> |
| <b>Glosario.....</b>                                             | <b>53</b> |

**Bibliografía..... 56**

## Introducción

La **World Wide Web** (o WWW, comúnmente conocida como **web**) es el servicio más utilizado en Internet, hasta el punto de que, aun siendo sólo uno de los servicios existentes, popularmente se confunde todo Internet con él.

En este módulo vamos a adentrarnos en todo lo relacionado con la web. Empezaremos con una perspectiva histórica, desde su concepción hasta su popularización. A continuación, conoceremos el modelo arquitectónico en el que se basa, entendiéndose aquí la arquitectura desde un punto de vista informático. De esta manera, hablaremos de los componentes que necesitamos para que la web funcione:

- un cliente que pedirá información,
- un servidor que la proveerá,
- un protocolo de comunicaciones (HTTP) para poder intercambiar esta información, y
- un lenguaje estándar (HTML) para enriquecer el texto y convertirlo en hipertexto.

Pero la web moderna tiene mucho más, y parte de todo ello se tratará también en este módulo. En particular, hablaremos de la web dinámica o interactiva, las aplicaciones web y la web 2.0. Finalmente, veremos cómo se han integrado en la web elementos multimedia como las imágenes, el audio y el vídeo.

## Objetivos

Tras el estudio de este módulo, el estudiante deberá haber adquirido las siguientes competencias:

- 1.** Conocer la historia de la web. Se estudiará quién creó la web y cómo ha ido evolucionando desde sus comienzos.
- 2.** Tener nociones de cómo funciona la web. Se verá qué elementos intervienen en este servicio y cómo se relacionan entre ellos.
- 3.** Comprender los conceptos básicos del lenguaje HTML. El lenguaje HTML es el lenguaje en el que están escritas las páginas web. El estudiante entenderá los mecanismos básicos de éste y tendrá las nociones suficientes para utilizar sus elementos en páginas web sencillas y en otros lugares donde también pueden ser de utilidad (por ejemplo, en foros).
- 4.** Conocer los fundamentos de la web dinámica y de la web 2.0. La web ha evolucionado desde sus comienzos estáticos e interactúa en la actualidad con el usuario de una manera más sencilla y rica. Esto abre un mundo de posibilidades, como se verá en el caso de la web 2.0.
- 5.** Entender los elementos multimedia y cómo se integran en la web. La web no es sólo texto; también hay imágenes, audio y vídeo. Veremos cómo se componen estos elementos multimedia y cómo se incluyen en las páginas web.



## 1. Historia

Aunque los orígenes de Internet se pueden identificar en la década de los sesenta del siglo pasado, la web es un servicio más bien reciente.

Un **servicio de Internet** resuelve una necesidad de un usuario por medio de Internet.

En los servicios de Internet intervienen varios ordenadores, por lo que se sostienen sobre protocolos y estándares. Normalmente cada servicio tiene su propio protocolo para transmitir la información durante el intercambio de datos.

Aparte del servicio web, existen otros servicios de Internet ampliamente utilizados:

- Correo electrónico.
- Mensajería instantánea.
- Acceso remoto.
- Compartición P2P.
- Juegos multi-usuario.
- *Streaming* de vídeo.
- Telefonía por Internet.
- Vídeo-conferencia.
- Computación masivamente distribuida.

Gran parte del éxito de la web reside en su facilidad de uso, lo que ha hecho que muchos de los servicios de Internet mencionados ofrezcan una interfaz web. Por lo tanto, hacemos uso de ese servicio dentro de un entorno que resulta familiar para el usuario: su navegador web.

### Ejemplo de servicio de Internet a través del navegador web

El servicio de correo electrónico es un ejemplo de servicio de Internet al que se accede mediante un navegador web. Hoy en día existen múltiples plataformas que ofrecen el servicio de correo electrónico vía web, como pueden ser Gmail, Yahoo! mail, Hotmail o el correo de la UOC. Los usuarios pueden abrirse una cuenta y gestionar sus buzones y mensajes de correo electrónico navegando por páginas web.

### 1.1. La web, el servicio estrella de Internet

Como ya hemos comentado, el servicio web no es uno de los primeros que existían en Internet. El servicio web es muy reciente, pero ha evolucionado muy rápidamente.

#### Ved también

Recordad que los protocolos de comunicaciones se explicaron en el módulo "Aspectos tecnológicos de las redes e Internet".

La historia de la web tiene su comienzo en el año 1989, cuando Tim Berners-Lee (conocido popularmente como TBL) propone un nuevo proyecto para facilitar el intercambio de información científica.

Berners-Lee, físico de formación, trabajaba en la Organización Europea para la Investigación Nuclear<sup>1</sup> (CERN, siglas en francés) en Ginebra. Se dio cuenta de que en la Internet de entonces ya existía una gran cantidad de información, pero que estaba vagamente organizada y estructurada. Así, la información estaba disponible y en gran cantidad, pero era difícil acceder a ella, ya que se encontraba en texto plano en múltiples documentos, sin referencias entre sí.

<sup>(1)</sup>Conocido en inglés como *European Organization for Nuclear Research*.

#### Texto plano

El texto plano es texto sin formato.

Por esta razón, Berners-Lee hizo uso del concepto **hipertexto** para lanzar un nuevo proyecto. La idea de hipertexto, conocida desde los años sesenta, se basa en que la información se puede organizar de manera dinámica mediante conexiones y enlaces –que reciben el nombre de **hipervínculos**– entre los documentos. Incluso existen documentos que simplemente sirven de índice, o sea, que son documentos con un listado de documentos enlazados. Cuando un usuario selecciona un hipervínculo dentro de un hipertexto, el programa le mostrará el documento al que hacía referencia ese hipervínculo. Había nacido la web, un tejido en el que los documentos se enlazan unos con otros.

Para llevar a cabo su idea, Berners-Lee ideó y creó lo siguiente:

- un **protocolo de transmisión**, que bautizó como protocolo de transmisión de hipertexto<sup>2</sup> (HTTP). Este protocolo sirve para indicar cómo se ha de pedir un documento de hipertexto y cómo se ha de enviar el documento.
- un **lenguaje de marcado** para enriquecer el texto plano y convertirlo en hipertexto, llamado *HyperText Markup Language* (HTML). La razón por la que se creó el HTML es que se necesitaba añadir información a los documentos de texto plano para que incluyeran los hipervínculos y otros elementos que veremos más adelante.
- un **servidor** en el que alojar las páginas de hipertexto (que llamó CERN HTTPd).
- un **cliente**, conocido como **navegador**, para acceder a las páginas alojadas en el servidor (y que curiosamente recibió el nombre de World Wide Web).

<sup>(2)</sup>En inglés *HyperText Transfer Protocol*.

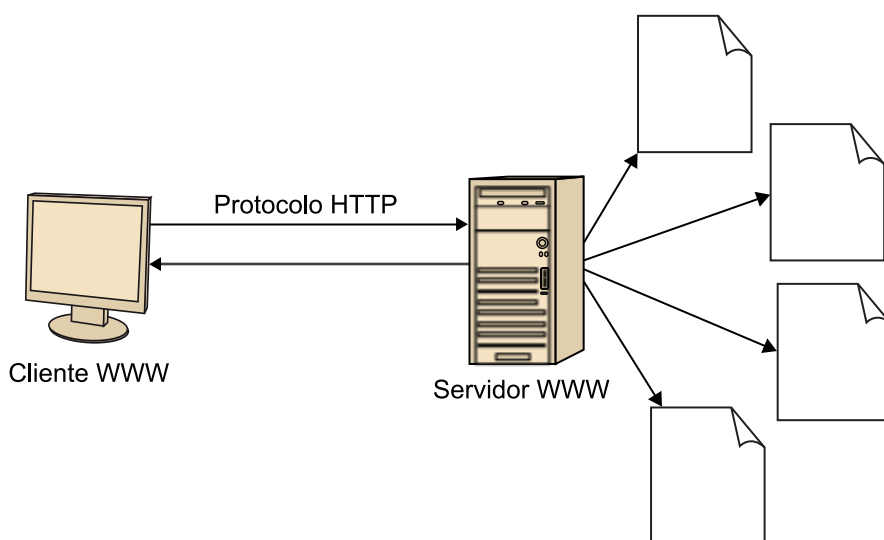
Hacia finales de 1990, con la ayuda de Robert Cailliau, un joven estudiante del CERN consiguió la primera comunicación con éxito entre el cliente y el servidor. Debido a su facilidad de uso, en poco tiempo el servicio WWW se convirtió en popular, lo que desplazó o integró otros servicios existentes en Internet con anterioridad. En 1994 Berners-Lee creó el World Wide Web Consortium (W3C) para regular las tecnologías relacionadas con la web (HTTP, HTML, etc.).

## 2. HTTP: cliente y servidor

La web se basa en el **paradigma cliente-servidor**, una de las arquitecturas que podemos encontrar comúnmente en Internet. En este paradigma, los actores que intervienen no están al mismo nivel ni tienen los mismos requisitos.

Tal y como muestra la figura siguiente, tenemos un nuevo modo de acceder a documentos que se encuentran en otro ordenador (al que llamamos **máquina servidora** o simplemente **el servidor**), y que son transferidos mediante un protocolo específico llamado HTTP. La máquina servidora tiene un software, que se denomina software **servidor web**, cuya tarea se limita a traducir las direcciones que recibe por HTTP en la ruta en local de los archivos para enviarlos al **cliente web** (un navegador) que los ha pedido.

Arquitectura cliente-servidor de un servicio WWW básico



### 2.1. El servidor

El servidor contiene **páginas web**, documentos de texto enriquecidos con hipervínculos y está atendiendo constantemente peticiones por parte de clientes. De esta manera, su labor reside en estar continuamente escuchando peticiones provenientes de clientes, interpretándolas y atendiéndolas, posiblemente respondiendo con la página web (u otros elementos que haya pedido). De este modo, *ad infinitum*. Uno de los requisitos esenciales de un servidor es que esté siempre encendido, porque de lo contrario no podrá atender convenientemente las peticiones que le lleguen. Otro requisito es que posea una dirección IP fija, ya que de alguna manera los clientes han de conocer a quién dirigirse de antemano. El servidor web será un programa en el nivel de aplicación.

#### Ved también

La definición del paradigma cliente-servidor se puede consultar en el subapartado 1.1 del módulo "Aspectos tecnológicos de las redes e Internet".

#### Ved también

Recordad que los niveles OSI se explicaron en el subapartado 1.3 del módulo "Aspectos tecnológicos de las redes e Internet".

Si el número de peticiones que un servidor tiene que atender es muy elevado, entonces será necesario contar con una máquina más potente (o sea, con mayor capacidad de procesamiento, mayor tamaño de memoria y mayor capacidad de almacenamiento en disco duro, por ejemplo). Por esta razón, aunque cualquier máquina pudiera actuar de servidor, cuando se espera mucho tráfico suelen ser generalmente máquinas bastante potentes (y caras).

Pero es muy probable que si la web que aloja es muy frecuentada, no baste con una sola máquina. Entonces entra en escena el concepto granjas de servidores<sup>3</sup>, que permiten escalar un servicio ofrecido a muchas máquinas. Las granjas de servidores actúan de manera transparente al cliente; esto es, el cliente realizará la petición como si sólo hubiera un servidor y no es consciente de cómo se trata su petición, si por una única máquina o por varias.

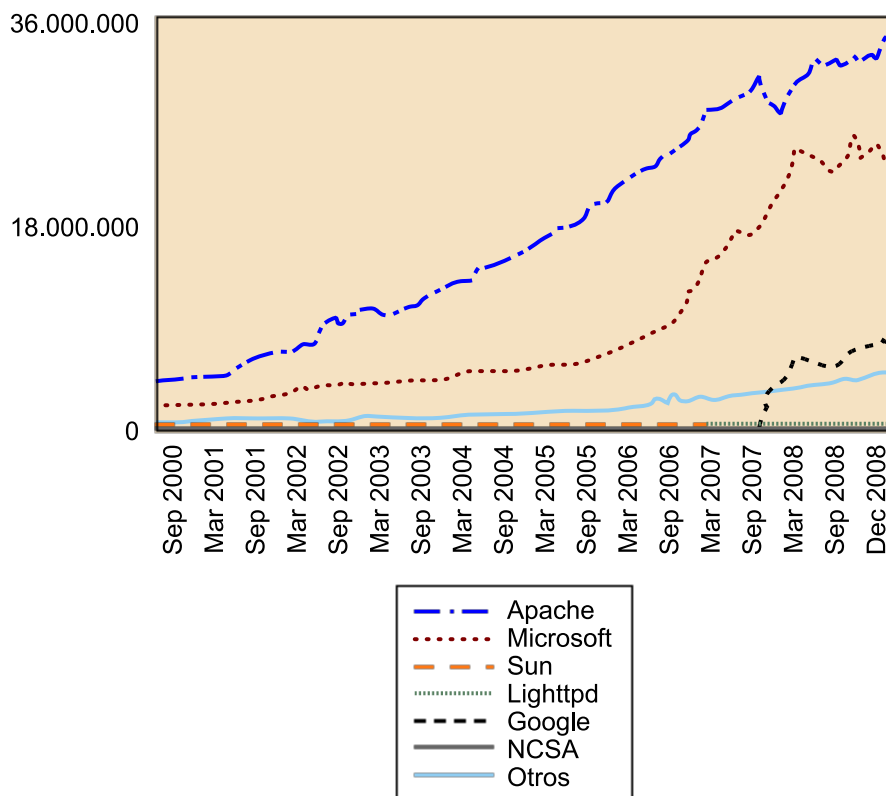
<sup>(3)</sup>En inglés, *server farms*.

Ha de notarse que servidor es una palabra polisémica y que se conoce como servidor web tanto a la máquina (física) que alberga las páginas web, como al propio software (lógico) que procesa y responde a las peticiones de los clientes.

Como se ha comentado con anterioridad, cualquier máquina física puede actuar de servidor y se pueden adquirir máquinas especialmente dotadas para servicios muy frecuentados.

En cuanto al servidor lógico, existen múltiples programas de ordenador en el mercado que pueden realizar sus funciones, aunque hay unos cuantos que históricamente han sido muy utilizados y que creemos conveniente conocer.

Crecimiento en términos absolutos del número de servidores web



(c) Netcraft LTD. Fuente: Netcraft LTD

Existen dos programas que son mayoritariamente utilizados: el servidor web Apache, un programa de software libre, y el Microsoft Internet Information Server (también conocido como *IIS*), una aplicación privativa creada por el gigante tecnológico Microsoft. A gran distancia de estos dos, se pueden encontrar otros software de servidor, como uno creado por la multinacional SUN, e incluso el incipiente crecimiento de servidores basados en el software de Google en los últimos tiempos.

Pero desde el punto de vista histórico, resulta más interesante observar la gráfica en la siguiente figura con la cuota de mercado que cada servidor ha tenido desde que se toman referencias para la encuesta de NetCraft (en agosto de 1995). Se puede observar que mientras que desde el año 2000 los resultados no varían mucho –el servidor web Apache y Microsoft Information Server son los servidores más utilizados–, esto no era así al principio. En 1994, el servidor web más utilizado era el NCSA, un software creado por una compañía llamada Netscape. Esta compañía, además de tener en catálogo este servidor web, ofrecía un navegador, conocido como Netscape Navigator. Mientras que el navegador podía descargarse de manera gratuita, había que pagar por el software de servidor web. Se trataba de una estrategia encaminada en ampliar el mercado de personas que navegaban por Internet (ofreciendo su navegador gratis), pero cobrando a los que ofrecían información y servicios (vendiendo el servidor).

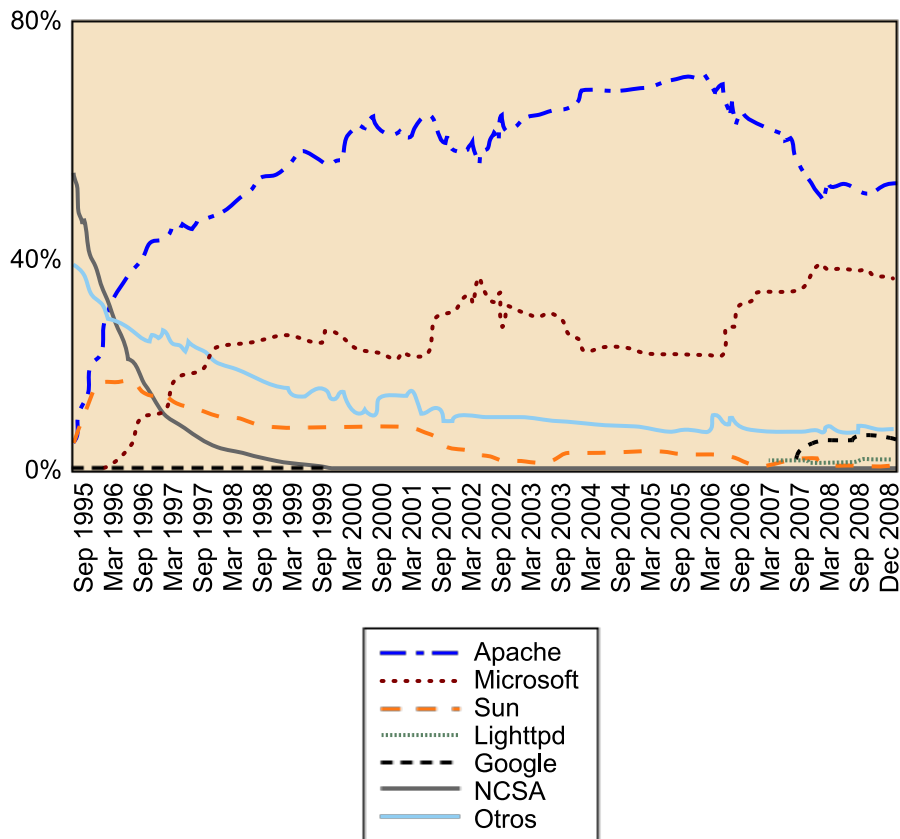
#### NetCraft

NetCraft es una empresa que se dedica a sondear el software que se utiliza como servidor web y que realiza un informe mensual publicado en la web.

#### Ved también

El software libre se describirá en el módulo "La cultura libre".

Cuota de mercado de los servidores web más utilizados en todos los dominios de Internet



(c) Netcraft LTD. Fuente: Netcraft LTD

La figura anterior muestra cómo Microsoft apostó de manera tardía por la web, ya que su servidor empezó a ser importante bien entrado el año 1997, cuando Apache y SUN ya le habían tomado la delantera.

## 2.2. El cliente

Por un lado, hemos visto que tenemos el servidor web, que requiere de una serie de condiciones. Por otro, está el **cliente web**, conocido generalmente como el **navegador**. El cliente web es un programa que se comunica con el servidor, para lo que utiliza una serie de reglas que veremos más adelante. Una de las características más importantes del modelo cliente-servidor es que en todo momento la comunicación sólo existe entre un cliente y un servidor, no habiendo nunca comunicación entre clientes. El cliente web es un programa en el nivel de aplicación y su acometido se divide en dos partes:

- Realizar una **petición**, por ejemplo, una página web alojada en el servidor web.
- Interpretar la **respuesta del servidor**. Esto significa que el cliente web no sólo ha de conocer las reglas para comunicarse con el servidor web (el protocolo HTTP), sino que también deberá saber qué hacer con los contenidos que éste le envía (generalmente utilizando el estándar HTML). Gran

parte de la complejidad de los clientes web actuales reside precisamente en mostrar los contenidos de manera conveniente en pantalla.

A diferencia del servidor, no es necesario que los clientes estén continuamente conectados a Internet y pueden tener IPs diferentes (dinámicas), ya que el servidor responde a la dirección de petición.

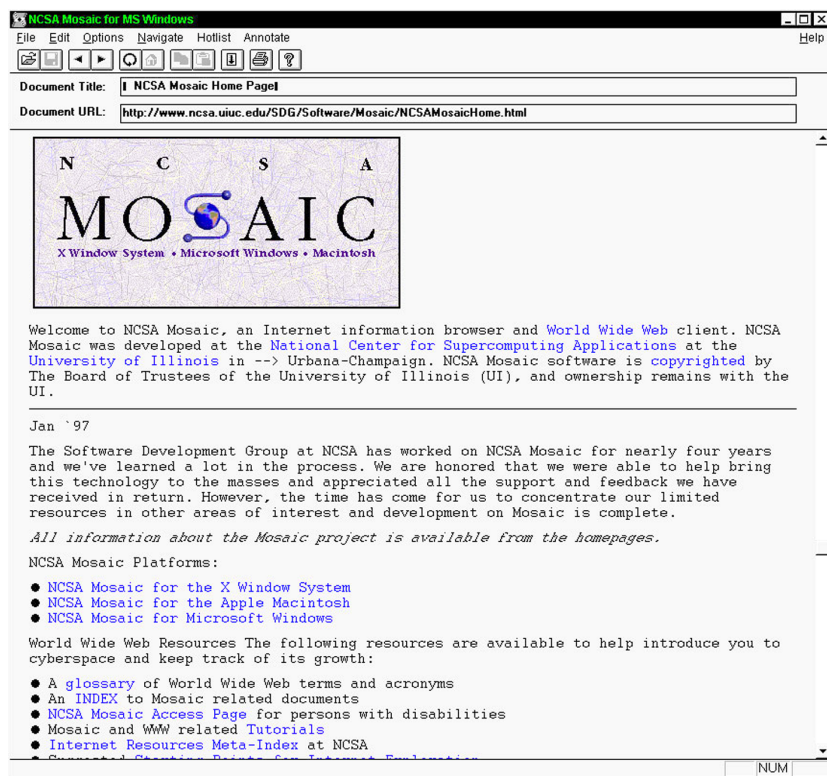
### Operadoras de comunicaciones

Las operadoras de comunicaciones (ISP) tienen un conjunto limitado de direcciones IP que asignan dinámicamente cuando uno de sus clientes se conecta a Internet. Por lo tanto, cada vez que un usuario se conecta a Internet cambia su dirección IP.

### Ejemplo de IP dinámica

Si pedimos una página web desde nuestro portátil en la oficina, nos la enviará a la IP actual (desde la que hemos realizado la petición). Una vez en casa, si pedimos de nuevo la página, el servidor nos volverá a enviar la información a nuestro portátil, que tendrá una dirección IP diferente.

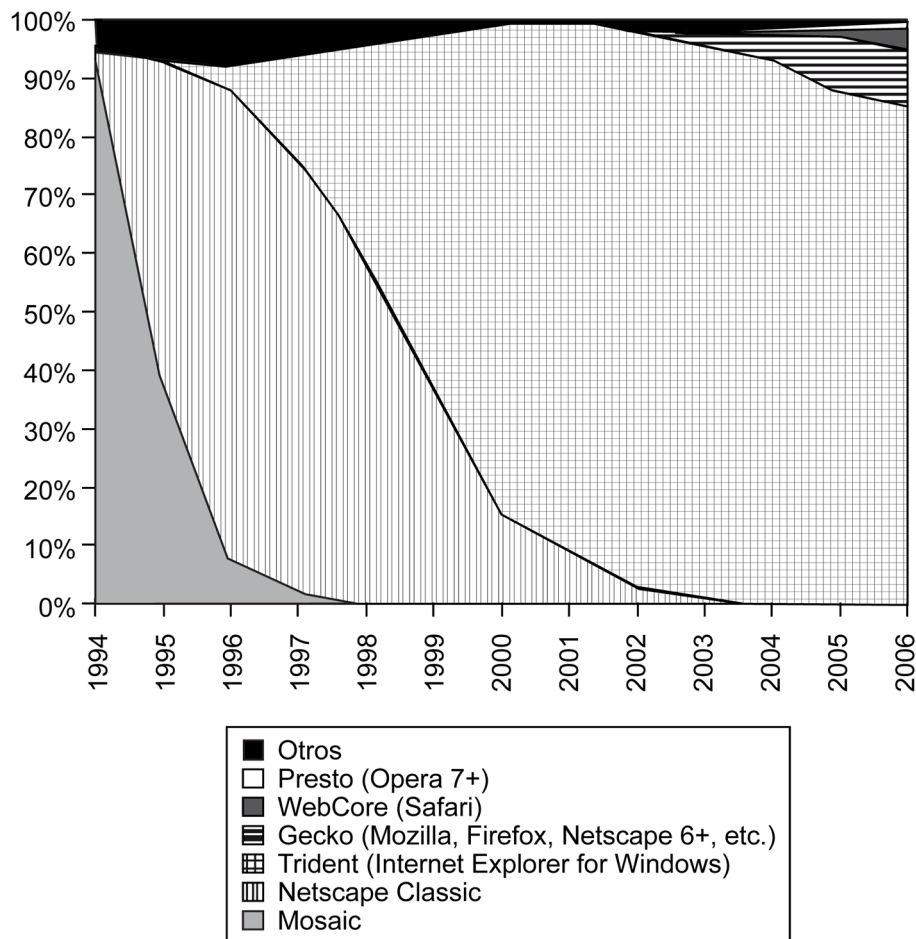
Como ya hemos comentado con anterioridad, el primer cliente de la World Wide Web fue creado por Berners-Lee en el CERN. Sin embargo, el primer navegador realmente popular fue Mosaic, concebido en el National Center for Supercomputing Applications (NCSA) en 1992. Los navegadores actuales, aun siendo más elaborados y completos, tienen gran parecido visual con Mosaic. Tal fue el éxito de Mosaic que sus creadores montaron una *spin-off* llamada Netscape que, como hemos visto con anterioridad, también vendía servidores.



Captura de pantalla del navegador web Mosaic, en su versión 6beta a finales de 1993. Fuente: National Center for Supercomputing Applications/University of Illinois Board of Trustees.

Mientras que el mercado de servidores ha estado fragmentado entre varios competidores, el de navegadores web siempre ha tenido un navegador que ha sido utilizado por la mayoría de los internautas. Hasta 1995 el cliente más utilizado era Mosaic, aunque Netscape Navigator tomó el testigo de manera rápida; al fin y al cabo, eran los continuadores de Mosaic.

Cuota de mercado histórica de los navegadores web más populares desde 1994



Fuente: Wikipedia. Licencia: Dominio Público

Entre 1997 y 1999 se puede observar un período que es conocido como la guerra de los navegadores. Para entonces, el público ya se había acostumbrado a que los navegadores web fueran gratuitos –siguiendo el modelo de negocio de Netscape–, por lo que las compañías que competían en este mercado lo hacían por otras razones. El principal motivo era que la congruencia de todos los servicios de Internet en la web convertía este servicio en un mercado muy apetitoso. Quien consiguiera poner su navegador en el mayor número de ordenadores tendría una posición muy interesante, ya que controlaría la puerta de entrada a la web. Microsoft, aunque tardíamente, se dio cuenta de esto y lanzó un software cliente web llamado Internet Explorer.

#### Ved también

Recordad que el modelo de negocio de Netscape se ha tratado en el subapartado 2.1 de este módulo.

La **guerra** se recrudeció debido principalmente a dos movimientos:

- En primer lugar, Microsoft utilizó como plataforma de lanzamiento de su navegador su sistema operativo Windows. Esto significaba que muchos



usuarios que compraban un ordenador y se instalaban el sistema operativo –o les venía preinstalado– ya tenían un navegador disponible. Por esta razón, se llevó a Microsoft a juicio por abuso de posición dominante. Cuando los jueces fallaron, Netscape había desaparecido –fue adquirido por AOL– y el desarrollo de su navegador se paralizó. Curiosamente, en 1997, en vista de que su cuota de mercado menguaba drásticamente, Netscape decidió liberar su código, ofreciéndolo bajo una licencia de software libre. De esta manera nació el proyecto Mozilla y de ahí el navegador Firefox (Gecko), que aparece en los últimos años como una alternativa a Microsoft Explorer.

- En segundo lugar, Netscape y Microsoft se lanzaron en una carrera por innovar para ofrecer mejores páginas web, más allá del estándar HTML promulgado por el W3C, el organismo encargado de estandarizar HTML. Lo que en un principio puede resultar una situación beneficiosa de innovación, no lo fue tanto, ya que se daba el caso de que los diseñadores web creaban páginas web con información que sólo podía ser vista correctamente por uno de los navegadores. Aunque las dos compañías fueron muy proclives a hacer este tipo de modificaciones propias, en cuanto Microsoft ganó una cuota de mercado importante acentuó estas prácticas. De esta manera, se aprovechaba del llamado efecto red. Un diseñador web tenía muchos incentivos para crear páginas web que llegaran al máximo número de usuarios. Al mismo tiempo, los usuarios utilizaban el navegador web con el que pudieran visualizar correctamente más páginas web. Por el efecto red, el navegador utilizado fue Internet Explorer, ya que la mayoría de los usuarios lo utilizaban.

### **El efecto red**

El **efecto red** se da cuando un bien o servicio tiene mayor valor cuanto más gente lo utilice. En otras palabras, el número de personas que han adoptado antes un servicio es un factor del valor disponible para el próximo adoptante. Un ejemplo clásico es el servicio telefónico: si poca gente tiene teléfono, es menos interesante tener uno que si todo el mundo tiene uno.

### 3. El protocolo HTTP

Una vez conocidos los lados cliente y servidor, vamos a conocer cómo se intercambia información. Para ello, se utiliza el protocolo HTTP creado por Tim Berners-Lee.

El protocolo utilizado en el servicio web tiene como nombre **protocolo de transferencia de hipertexto<sup>4</sup>(HTTP)**. Las primeras versiones del protocolo fueron diseñadas por Tim Berners-Lee, pero después ha pasado a ser coordinado por el World Wide Web Consortium (W3C) y el Internet Engineering Task Force (IETF). El protocolo viene especificado por un RFC. El RFC de la versión actual de HTTP, la versión 1.1, es el RFC 2616 y fue publicado en junio de 1999.

El protocolo HTTP es un protocolo basado en texto, por lo que puede ser entendido fácilmente por humanos. Existen dos tipos de mensajes HTTP:

- Los mensajes de **petición**, realizados por el cliente al servidor.
- Los mensajes de **respuesta**, enviados del servidor al cliente.

Un mensaje de petición contiene los siguientes elementos:

- La línea con la petición, en la que se especifica un **método** y un **objeto**. El método es la acción que se debe realizar, mientras que el objeto es, como su propio nombre indica, el objeto de la acción. Existen varios métodos de peticiones diferentes como se muestra en la tabla siguiente. Entre los métodos HTTP podemos distinguir métodos seguros e inseguros. Los métodos seguros reciben ese nombre porque no cambian el estado del servidor y sólo realizan peticiones de información. Son HEAD, GET, OPTIONS, CONNECT y TRACE. El resto de métodos, POST, PUT y DELETE, sí cambian el estado del servidor o realizan acciones de envío de información al servidor. Evidentemente, con estos últimos métodos hay que tener más cuidado en el servidor, ya que pueden poner en riesgo su seguridad.
- Ninguna, una o varias **cabeceras**. Las cabeceras permiten enviar información adicional sobre la petición, como el navegador utilizado o el lenguaje preferente. Esta información puede ser útil para el servidor. Por ejemplo, si el servidor ofrece los contenidos en varios idiomas, puede ver cómo hemos configurado nuestro cliente a partir de las cabeceras y mandar la información en el idioma seleccionado.
- Una línea en blanco.
- El cuerpo del mensaje opcional. Aquí irán los contenidos que enviamos al servidor. Generalmente, una petición de página web no tiene conteni-

#### Ved también

Recordad que los protocolos de comunicaciones se explicaron en el módulo "Aspectos tecnológicos de las redes e Internet".

<sup>(4)</sup>En inglés *Hypertext Transfer Protocol*.

#### Los RFC

Recordad que los RFC (*Request For Comments*) son acuerdos de la IETF con relación a un protocolo o estándar.

dos. Pero esto no es el caso cuando enviamos un formulario o subimos un archivo.

Métodos HTTP más importantes utilizados para realizar peticiones a un servidor web

| Método HTTP | Uso                                                                                                                                                          |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| GET         | Pide al servidor un recurso (página web, imagen, archivo de sonido, etc.).                                                                                   |
| POST        | Envía datos al servidor para ser procesados.                                                                                                                 |
| HEAD        | Pide un recurso (página web, imagen, archivo de audio, etc.), pero sin que se le envíe el cuerpo de la respuesta, de manera que sólo recibirá las cabeceras. |
| PUT         | Manda el recurso (página web, imagen, archivo de sonido, etc.) dado por la dirección al servidor.                                                            |
| DELETE      | Borra el recurso.                                                                                                                                            |
| OPTIONS     | Pide al servidor los métodos que pueden ser utilizados con el recurso.                                                                                       |
| TRACE       | Método utilizado para realizar pruebas.                                                                                                                      |
| CONNECT     | Es un método reservado para realizar túneles en proxies.                                                                                                     |

Los mensajes de respuesta enviados por el servidor tienen la misma estructura que los de petición, aunque difieren en la primera línea. Así, en vez de enviar un método y un objeto, el servidor devuelve un código numérico indicando el **estado de la petición**. Al código numérico se le añade una breve descripción textual sobre su significado. En la siguiente tabla se muestran algunos de los códigos de respuesta HTTP más comunes, su descripción y su significado.

Códigos HTTP de respuesta (y su descripción) más utilizados

| Código de estado | Descripción       | Significado                                                                    |
|------------------|-------------------|--------------------------------------------------------------------------------|
| 200              | OK                | La petición ha sido realizada con éxito.                                       |
| 301              | Moved permanently | El recurso pedido tiene una nueva dirección permanente.                        |
| 302              | Found             | El recurso pedido tiene una nueva dirección temporal.                          |
| 403              | Forbidden         | El servidor ha entendido la petición, pero no la ejecutará.                    |
| 404              | Not Found         | El servidor no ha encontrado nada por esa dirección.                           |
| 500              | Server Error      | El servidor ha encontrado un error interno que le impide procesar la petición. |

Una de las ventajas del protocolo HTTP es que al ser texto plano, podemos leerlo y entenderlo con facilidad. Por desgracia, no sólo el cliente y el servidor lo podrán hacer, sino que cualquiera en un *router* intermedio de Internet podría hacer lo mismo sin muchos problemas. Para ciertos usos, la facilidad de lectura se convierte en una desventaja. Podríamos estar interesados en que

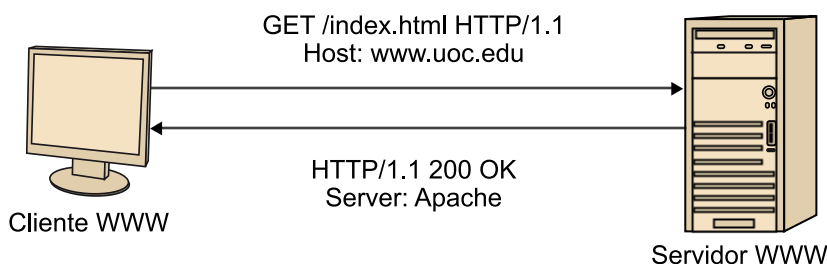
nadie leyese nuestras peticiones o las respuestas del servidor, como al realizar una transferencia bancaria, pagar con nuestra tarjeta de crédito o mirar nuestros datos fiscales en la web del ministerio correspondiente.

Para solucionar este problema, se propuso el **protocolo HTTPS** (HTTP seguro), estandarizado en el RFC 2818, que es una combinación del protocolo HTTP con técnicas de cifrado. Por ello, cuando accedemos a la página en línea de nuestro banco, lo hacemos pidiendo una página cuya dirección web empieza por "https" y no "http". Mediante este mecanismo, no sólo se asegura la confidencialidad de la transmisión, sino que también se evita la suplantación.

### 3.1. Un ejemplo de comunicación con HTTP

Una petición HTTP de la página principal de la página de la UOC podría ser tal y como se muestra de manera gráfica (y resumida) en la figura siguiente.

Ejemplo de intercambio de mensajes HTTP entre un cliente un servidor



La petición completa que envía el cliente al servidor web de la UOC es la siguiente:

```
GET /index.html HTTP/1.1
Host: www.uoc.edu
User-Agent: Mozilla/5.0 (X11; Linux i686; en-US; rv:1.9.0.5)
```

Se puede observar que en la petición aparece información sobre el cliente que estamos utilizando (en el caso del ejemplo se trata del navegador Mozilla Firefox 1.9.0.5, que utiliza el motor Mozilla 5.0), el sistema operativo (basado en Linux), el entorno gráfico (en este caso, X11, un interfaz de usuario gráfico para Linux) e incluso el idioma (en-US) en el que tenemos configurado el navegador.

La respuesta del servidor HTTP de la UOC a nuestra petición anterior sería la siguiente:

```
HTTP/1.1 200 OK
Server: Apache
Last Modified: Fri, 14 Dec 2007 10:16:44 GMT
Content-type: text/html
Content-length: 1371
```

#### Ved también

Recordad que en el subapartado 4.1 del módulo "Seguridad en la sociedad de la información" se habla de identidad digital.

#### Las direcciones web

Las direcciones web son también conocidas como URL o *Uniform Resource Locator*, localizador uniforme de recurso.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ca">
<head>
[...]
```

Se puede observar que en la primera línea se especifica tanto el protocolo (y su versión) como el código de respuesta (200 OK). A continuación, en líneas sucesivas y hasta una línea en blanco, recibimos información mediante cabeceras. Se puede comprobar qué software de servidor se está utilizando (Apache), la fecha de la última modificación del documento, el tipo de contenido que se está enviando (en este caso, una página de texto en lenguaje HTML) y, finalmente, la longitud de la página en bytes (1.371 bytes). Luego viene el contenido, que es una página web en lenguaje HTML. Por su excesiva longitud, en la figura se muestra sólo las cabeceras de la página web.

## 4. El lenguaje HTML. Nociones básicas

Como ya se ha comentado con anterioridad, uno de los aspectos fundamentales de la popularidad de la web es el concepto de hipertexto. El texto plano es una buena manera de contener información, pero adolece de elementos estructurales y de apariencia. Hasta cierto punto, nos gustaría añadir además información sobre el texto en sí y referencias a otros textos. En este apartado veremos cómo conseguirlo con el lenguaje de marcado HTML.

Berners-Lee pensó que la manera de conseguir esa riqueza adicional en texto plano es añadiéndole **etiquetas**. Si queremos que un texto aparezca en negrita, debemos crear un **elemento HTML** del siguiente modo:

Esta es una frase con `<b>texto en negrita</b>`.

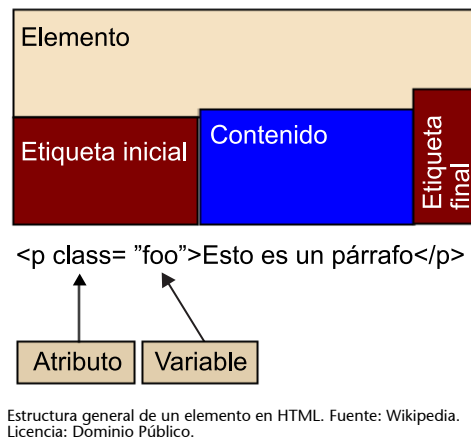
Nótese que lo que estamos añadiendo es información al texto plano para que el navegador la interprete de manera conveniente. Y lo más interesante es que esta información que estamos añadiendo, ¡la estamos añadiendo también en texto plano! Los elementos HTML vienen dados por etiquetas de principio y de final. El hecho de que las etiquetas se encuentren entre signos de menor y mayor es precisamente para que el navegador sepa distinguir entre la información que queremos representar y la información sobre cómo la queremos representar.

Las etiquetas y la manera de utilizarlas vienen especificadas en un estándar (en realidad, varios estándares, porque ha habido varias versiones) y han sido publicados como un estándar del W3C. Este lenguaje recibe el nombre **lenguaje de marcado de hipertexto**<sup>5</sup>(HTML).

<sup>(5)</sup>En inglés, *Hypertext Markup Language*.

En general, los **elementos HTML** se forman de la siguiente manera:

- Una etiqueta marcando el principio de un elemento.
- Opcionalmente, se aceptan **atributos** con sus valores (variables) dentro de la etiqueta.
- El contenido del elemento (que puede ser texto y/u otros elementos).
- Una etiqueta final.



Se da el caso de elementos que son vacíos, por lo que no precisan de contenido. El ejemplo más conocido es el del salto de línea (<br>). En tales casos, la etiqueta final no es necesaria. En la siguiente tabla se pueden ver algunas de las etiquetas HTML más utilizadas.

Etiquetas HTML más frecuentes

| Etiqueta   | Significado/uso                                                                                                                                                                       |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| html       | Inicio de una página HTML.                                                                                                                                                            |
| title      | Título de la página HTML.                                                                                                                                                             |
| head       | Cabecera de la página HTML. Aquí se incluye metainformación sobre la página, como el título, autoría, palabras clave, etc.                                                            |
| meta       | Metainformación sobre la página, generalmente incluida en la cabecera.                                                                                                                |
| body       | Cuerpo de la página. Es donde viene el contenido de ésta.                                                                                                                             |
| p          | Etiqueta de párrafo.                                                                                                                                                                  |
| br         | Etiqueta de salto de línea.                                                                                                                                                           |
| b          | Etiqueta de texto en negrita.                                                                                                                                                         |
| i          | Etiqueta de texto en itálica.                                                                                                                                                         |
| hr         | Línea horizontal.                                                                                                                                                                     |
| a          | Etiqueta de enlace. Cuando se trata de un enlace a otra página, requiere al menos un atributo, href, que indica la dirección web de destino.                                          |
| center     | Etiqueta para centrar contenido.                                                                                                                                                      |
| img        | Etiqueta de imagen. Requiere al menos dos atributos: src para indicar la dirección de la imagen y alt para mostrar texto alternativo en el caso de que no se pueda mostrar la imagen. |
| table      | Etiqueta de tabla.                                                                                                                                                                    |
| td         | Etiqueta de elemento de una tabla.                                                                                                                                                    |
| tr         | Etiqueta de fila de una tabla.                                                                                                                                                        |
| h <i>n</i> | Etiqueta de encabezado. El <i>n</i> da el nivel de encabezamiento y va desde 1 (más grande) a 6 (más pequeño).                                                                        |

## 4.1. Un ejemplo de página web

A continuación, se muestra un ejemplo del código de una sencilla página web (creada a partir de las etiquetas que se muestran en la tabla del apartado 4). Se ha utilizado el espaciado horizontal para que se pueda determinar visualmente qué elementos son contenidos en otros.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN"
"http://www.w3.org/TR/1998/REC-html40-19980424/loose.dtd">

<html>
  <head>
    <title>Página de ejemplo: Fundamentos Tecnológicos</title>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  </head>

  <body>
    <h2>Bienvenidos a mi primera página web</h2>

    <p>Ésta es la <b>primera página web</b> para la que leo el código.
      Y lo hago porque estoy cursando un máster en la
      <a href="http://www.uoc.edu">Universitat Oberta de Catalunya</a>.
    </p>

    <hr>

    <p>A continuación, vamos a introducir como imagen el logo de la W3C, el
      organismo encargado de estandarizar el HTML. Para ello, vamos a hacer
      referencia al logo que podemos encontrar en su página principal
      (tomando su dirección, o URL). Aquí va:</p>

    <hr>

    <p>Pero podríamos ir más allá y combinar varias etiquetas, por ejemplo,
      para conseguir que la imagen fuera a la vez un enlace y estuviera
      centrada. ¿Cómo? Véase a continuación:</p>

    <center>
      <a href="http://www.w3c.org">
        
      </a>
    </center>

    <hr>
```



```
<p>Vamos a probar a introducir una <i>tabla</i> en HTML.</p>

<table border="1">
  <tr>
    <td>Elemento 1.1</td>
    <td>Elemento 1.2</td>
    <td>Elemento 1.3</td>
  </tr>
  <tr>
    <td>Elemento 2.1</td>
    <td>Elemento 2.2</td>
    <td>Elemento 2.3</td>
  </tr>
</table>

</body>

</html>
```

La primera línea indica el tipo de documento del que se trata: un documento que sigue el estándar "HTML versión 4.0 Transitional"; asimismo, se adjunta la dirección de la especificación técnica del estándar.

El resto del documento empieza y termina con una etiqueta `html` que indica que se trata de un documento HTML y que contienen el resto de elementos. Así, si `html` es la etiqueta raíz (o de primer nivel), podemos observar que en el segundo nivel contamos con dos etiquetas: `head` y `body`.

En la cabecera, dentro de `head`, podemos encontrar información sobre la página web en sí; en nuestro ejemplo, simplemente hemos indicado el título de la página web utilizando la etiqueta `title`. Además, mediante una etiqueta `meta`, se informa de que el tipo de contenido es una página de texto, en particular HTML, y que el tipo de codificación de caracteres es UTF-8. Si quisiéramos se podría añadir mucha más información, como viene indicado en el estándar HTML: autor de la página web, palabras clave, etc.

En el cuerpo, entre las etiquetas `body`, encontramos el contenido de la página web. En nuestro caso, empezamos con un texto que queremos que sea más grande de lo habitual, por lo que utilizamos la etiqueta `h2`. Recordemos que si quisiéramos que este texto fuera todavía más grande, deberíamos utilizar en su lugar la etiqueta `h1`.

Justo a continuación viene el primer párrafo (que se puede identificar fácilmente al estar entre etiquetas `p`). Este primer párrafo contiene otras etiquetas: primera página web aparecerá en negrita (**bold**) al encontrarse etiquetado como `b`, mientras que `Universitat Oberta de Catalunya` es un enlace

a la página principal de la UOC. Los enlaces vienen dados por la etiqueta `a` y requieren de un parámetro adicional, en este caso `href`, para saber cuál es la dirección de destino.

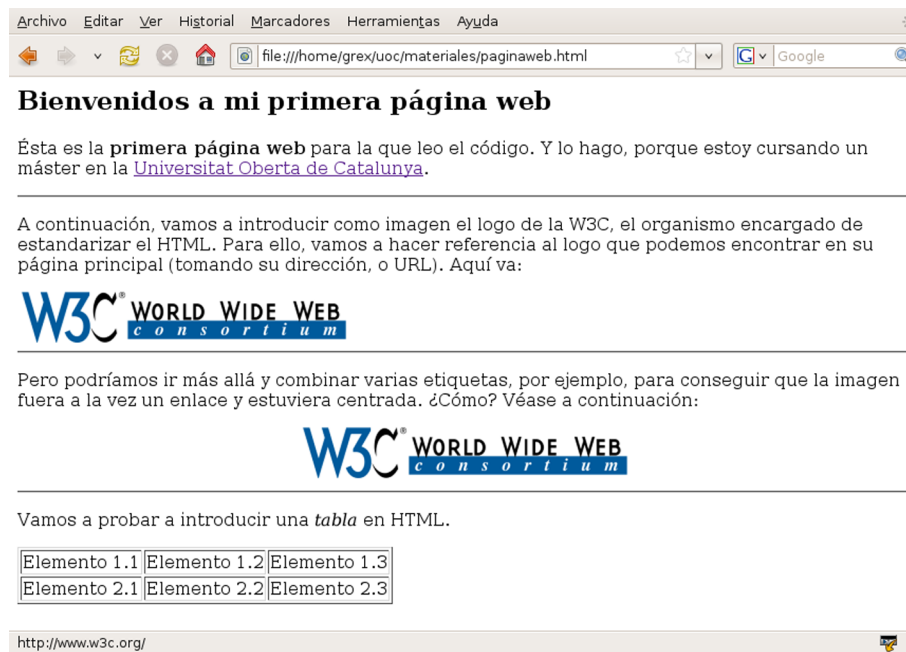
Algo parecido ocurre en el caso de querer insertar una imagen, tal y como hacemos en el siguiente párrafo. Para incluir imágenes, la etiqueta que debemos utilizar es `img` y el número de parámetros necesarios son dos: la dirección de la imagen (`http://www.w3.org/Icons/w3c_main`) y un texto alternativo (Logo W3C), utilizando el atributo `alt`. Este texto aparecerá en caso de que no se encuentre la imagen, de que el navegador no sea un navegador gráfico o incluso de que sea un navegador para invidentes (en cuyo caso se leerá este texto alternativo por los altavoces).

Se pueden encadenar varias etiquetas para conseguir el efecto agregado de todas ellas. Así, el siguiente párrafo contiene una imagen centrada, que es a su vez un enlace a la página web del W3C. Ya conocemos todos los elementos de este enlace, salvo el atributo `border`, que está puesto a `0`. Si no se especifica este atributo, a las imágenes se les añade por defecto un borde de color azul para que visualmente se pueda reconocer que se trata de una imagen que enlaza otra página.

Finalmente, se muestra cómo incluir una tabla en HTML. La etiqueta `table` es la que indica que vamos a construir una tabla, y le pasamos como parámetro `border` con valor a `1` para que se puedan ver los bordes de la tabla. La tabla tiene dos filas (dos etiquetas `tr`) de tres columnas (por eso hay tres etiquetas `td` en cada fila).

Para crear esta página web, sólo es necesario copiar y pegar el texto en un editor de texto (como el "bloc de notas" o el "Wordpad") y guardarlo en el disco con el nombre `miprimeraweb.html`. Haciendo un doble clic sobre el fichero se abrirá la página web en el navegador. En la siguiente figura, se muestra cómo es interpretado por un navegador web.

Visualización con el navegador de la página web que hemos creado



En el momento de la captura, el puntero del ratón (que no aparece en la captura) estaba situado sobre la segunda imagen. Esta imagen, centrada, era además un enlace a la página principal del W3C. Por ello, al pie de la ventana del navegador, se puede ver la dirección a la que apunta: <http://www.w3c.org>. Si hacemos clic con el ratón, iremos a parar a esa página.

El lenguaje HTML es un estándar que tiene varias versiones. Una vez que hemos terminado de hacer nuestra página web, podemos ir al servicio de validación web del W3C, que nos indicará si hemos hecho un uso correcto del estándar. En caso contrario, nos dirá los errores que hemos cometido, por ejemplo, si falta algún atributo o si no hemos cerrado alguna etiqueta.

#### Servicio de validación del W3C

El servicio de validación de la W3C se puede encontrar en <http://validator.w3.org/>.

## 5. La web dinámica o interactiva y aplicaciones web

Hasta ahora, y así fue en los primeros años de existencia del servicio web, hemos considerado que un servidor web ofrece una serie de documentos que los navegadores pueden pedir mediante el protocolo HTTP. Estos documentos son unos documentos de texto enriquecidos con etiquetas HTML que hacen no sólo que podamos darle estructura y mostrarlos visualmente de una manera muy rica, sino que también permitían estar enlazados entre sí. Esto nos permitía pasar fácilmente de uno a otro, y se ha venido a llamar "navegar" por la web.

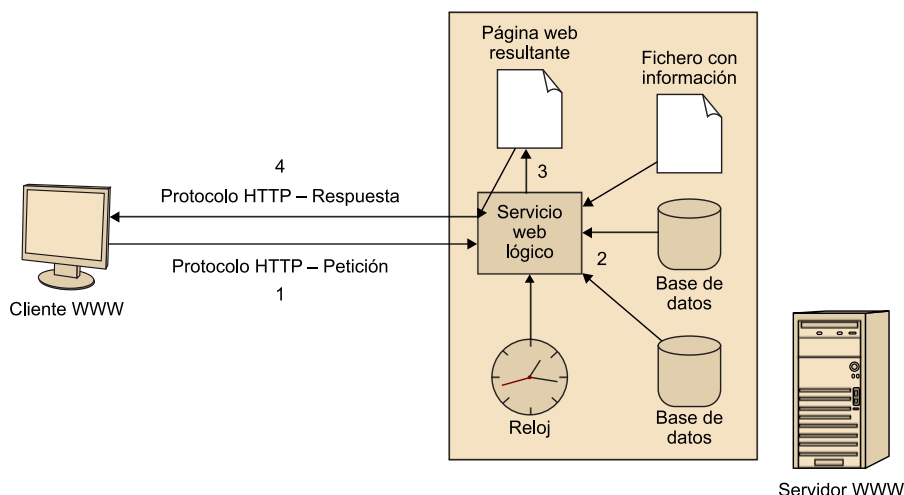
Sin embargo, pronto muchos vieron que este sistema, aun siendo muy simple y de gran provecho, tenía sus limitaciones. Una de las limitaciones más importantes era que la interactividad se reducía precisamente al hecho de navegar por páginas en los servidores. En esta situación, la información ya se encontraba de antemano en el servidor y no se podía modificar, añadir o corregir de manera sencilla. Y otra limitación es que esta información es estática, que no interactúa con el usuario. En otras palabras, en la web que hemos presentado hasta ahora los servidores son los contenedores de información y los clientes los que la piden. El usuario era un mero consumidor de información. Por otro lado, esta información no se genera con la petición del cliente, sino que debía existir con anterioridad en el servidor.

En este apartado estudiaremos cómo la web evolucionó de un marco estático, como el descrito, a otro dinámico, en el que el servidor interactúa de manera mucho más estrecha con el cliente. Ofreceremos una perspectiva de cómo se consigue esta interacción individualizada con el usuario y veremos que de esta tendencia nacieron las aplicaciones web.

El caso es que pronto muchos se dieron cuenta que el software servidor podría hacer más cosas que simplemente identificar el documento que se ha pedido y enviarlo al cliente. Así, por ejemplo, se podría crear ese documento dinámicamente o "al vuelo" y enviarlo al cliente. O se podría construir una página web a partir de datos almacenados en una o varias bases de datos. O muchas otras posibilidades, ya que el software de servidor puede realizar una gran cantidad de tareas. Esto da una enorme versatilidad a la web.

La siguiente figura intenta mostrar esto de manera comprensible.

Esquema de un servicio web que ofrece contenidos generados dinámicamente de varias fuentes de información



Por un lado, a la izquierda, tenemos al cliente, que es el que realiza peticiones; en este caso, una página web dinámica. En el otro lado, a la derecha, se encuentra el servidor web. Al recibir la petición HTTP del cliente (paso 1), el servidor web lógico (que es un programa que se ejecuta dentro del servidor físico y que hemos representado por un rectángulo) toma información posiblemente de varias fuentes de información (paso 2). Así, se podría dar el caso de que esas fuentes de información fueran una o más bases de datos, archivos en el disco duro e incluso información propia de ejecución (como, por ejemplo, la hora). Esta información se combina con etiquetas HTML de la manera que se desee para crear una nueva página web (paso 3). Finalmente, la página web es enviada de vuelta por parte del servidor web lógico mediante un mensaje de respuesta HTTP (paso 4). El aspecto crucial de todo este proceso es que la página web resultante estará codificada en el lenguaje HTML que entienden los navegadores, y que no se encuentra almacenada como tal en el sistema de archivos del servidor. Mediante esta simple idea, hemos creado un modo de obtener contenidos dinámicos que interactúen con el cliente. Un mundo nuevo de posibilidades se abre ante nuestro navegador.

Precisamente, el hecho de que la web sea considerada como la **puerta de entrada a Internet** y que muchos confundan Internet con la web es debido a esta versatilidad. Al fin y al cabo, si el servidor web se conecta al servidor de correo electrónico y ofrece sus contenidos al usuario mediante páginas web, lo que estamos haciendo es acceder a un servicio de Internet clásico como el correo electrónico mediante la web. El usuario no tiene ni siquiera noción de que está utilizando otro servicio; para él sigue siendo la web. Si a esto le añadimos que "usar" la web es muy sencillo, entenderemos que la **convergencia hacia la web** es un hecho natural, lo que hace caer en el olvido el resto de servicios de Internet.

Las **aplicaciones web** son aquellas aplicaciones a las que los usuarios pueden acceder mediante el navegador web.

Hemos añadido tanta complejidad al programa servidor web que éste se puede asemejar a una aplicación como las que tenemos instaladas en nuestro propio disco duro. Con la salvedad, evidentemente, de que una aplicación web se encuentra almacenada y se ejecuta en la máquina servidor. La manera de interactuar con la aplicación web es mediante páginas HTML que visualizaremos en nuestro navegador. Esto constituye una gran ventaja, ya que el interfaz del navegador y el concepto de hiperenlaces es conocido por el usuario. Además, los creadores de una aplicación web se pueden olvidar de los problemas de instalación en los ordenadores de sus usuarios. No es necesario detectar qué sistema operativo utiliza el usuario en su ordenador personal, ya que con que éste tenga un navegador es suficiente.

### Ejemplos de aplicaciones web

Existen muchas aplicaciones web, la mayoría muy populares. Entre las aplicaciones web más utilizadas podemos contar con:

- Servicios de correo-web (o *webmail*): ofrecen acceso a nuestro buzón de correo electrónico mediante el navegador. Los servicios modernos de correo-web ofrecen gran capacidad de almacenamiento, libreta de direcciones, filtros frente al correo no deseado (SPAM), etc.
- Periódicos en línea: las páginas web de los periódicos modernos en línea rara vez existen como tal en el disco duro de un servidor. Generalmente se crean "al vuelo" incluyendo cabeceras, la noticia o bloques laterales (noticias relacionadas, noticias más leídas, etc.).
- Centrales de reserva de viajes: permiten acceder a las grandes bases de datos de compañías aéreas y promotores de viajes, de manera que se pueda organizar un viaje desde el ordenador de casa.

Más ejemplos de aplicaciones web lo constituyen los buscadores, las páginas wiki, los blogs, los sitios de redes sociales, etc.

## 5.1. Buscadores

Las aplicaciones web de busca de páginas web son un servicio indispensable en la actualidad, pero no ha sido siempre así. En este subapartado, además de la evolución histórica de los buscadores, introduciremos brevemente cómo funcionan.

Los **buscadores** son aplicaciones web que nos permiten encontrar información en Internet de manera rápida. Mediante la introducción de una o varias palabras clave, nos devuelve una página índice que enlaza páginas web con información relacionada.

Con el auge de la web y toda la información que había en ella, también surgió el problema de cómo encontrar la información deseada. Los primeros esfuerzos, allá por 1993, se dedicaron a crear páginas web que enlazaban con otras páginas web, generalmente de un mismo tema.

Así, surgieron los primeros **índices temáticos**, sistemas de búsqueda por temas como Lycos (en 1994) o Yahoo! (en 1995). Estos sitios tenían una jerarquía de temas por los que se podía navegar y que contenían páginas web o subtemas. Los índices temáticos contaban con personas que se encargaban de identificar, clasificar y listar las páginas web según temas o categorías. Se trataba de una tarea laboriosa y con sus limitaciones, ya que era realizada por humanos, aunque –precisamente por ser revisada– eran de gran calidad. Los índices temáticos tenían, además de su índice navegable por temas o categorías, un formulario que permitía realizar búsquedas dentro sus propias páginas. Si buscabas "fútbol", te llevaba a la página del índice temático sobre fútbol con enlaces sobre el tema; todo lo que no estaba incluido en el índice no era accesible.

En la actualidad, los índices temáticos han pasado a un segundo plano por cuestiones tecnológicas, ya que se han conseguido formas automáticas y eficientes de clasificar y mostrar la información. Aun así, existe un proyecto llamado **Open Directory Project** (ODP o DMOZ de [directory.mozilla.org](http://directory.mozilla.org); en castellano se podría traducir como Proyecto de Directorio Abierto) que ofrece este servicio con la colaboración de una comunidad de editores voluntarios.

 open directory project

[about dmoz](#) | [suggest URL](#) | [help](#) | [link](#) | [editor login](#)

[advanced](#)

[Arts](#)  
[Movies](#), [Television](#), [Music](#)...

[Business](#)  
[Jobs](#), [Real Estate](#), [Investing](#)...

[Computers](#)  
[Internet](#), [Software](#), [Hardware](#)...

[Games](#)  
[Video Games](#), [RPGs](#), [Gambling](#)...

[Health](#)  
[Fitness](#), [Medicine](#), [Alternative](#)...

[Home](#)  
[Family](#), [Consumers](#), [Cooking](#)...

[Kids and Teens](#)  
[Arts](#), [School Time](#), [Teen Life](#)...

[News](#)  
[Media](#), [Newspapers](#), [Weather](#)...

[Recreation](#)  
[Travel](#), [Food](#), [Outdoors](#), [Humor](#)...

[Reference](#)  
[Maps](#), [Education](#), [Libraries](#)...

[Regional](#)  
[US](#), [Canada](#), [UK](#), [Europe](#)...

[Science](#)  
[Biology](#), [Psychology](#), [Physics](#)...

[Shopping](#)  
[Autos](#), [Clothing](#), [Gifts](#)...

[Society](#)  
[People](#), [Religion](#), [Issues](#)...

[Sports](#)  
[Baseball](#), [Soccer](#), [Basketball](#)...

[World](#)  
[Deutsch](#), [Español](#), [Français](#), [Italiano](#), [Japanese](#), [Nederlands](#), [Polska](#), [Dansk](#), [Svenska](#)...

[Become an Editor](#)

Help build the largest human-edited directory of the web



Copyright © 1998-2006 Netscape

La segunda generación de sitios de búsquedas, que datan de finales de los años noventa del siglo XX, viene dada por la aparición de los **motores de búsquedas**.

Los motores son programas que trabajan en tres etapas:

- La primera etapa consiste en la descarga de páginas web. Para ello, los motores de búsqueda se sirven de programas que se denominan **arañas**, se descargan las páginas web y siguen los enlaces que encuentran en esas direcciones para seguir descargando otras páginas. Así, una página aparecerá en un motor de búsqueda si, y sólo si, es enlazada por otra página web a la que el motor de búsqueda tenga acceso. Aunque existen páginas web que son visitadas por las arañas con mayor frecuencia, por ejemplo los sitios de noticias o algunos foros, en general, durante un mes se recorren todas las páginas web. Es por ello que si subimos una página web al servidor web no aparece automáticamente en un motor de búsqueda; debemos esperar a que una araña visite nuestra página web para que aparezca. También ha de notarse que, al descargarse la página web, el motor de búsqueda está guardando una copia en sus discos duros. Algunos servicios de motor de búsqueda ofrecen la posibilidad de ver esa copia (conocida como la copia *caché*).
- El segundo paso del proceso consiste en la indexación de la página web. Para ello, el motor de búsqueda utiliza una serie de algoritmos para extraer las **palabras clave**. Las palabras que aparecen en el título de la página suelen tener mayor peso que las que aparecen en el cuerpo. Asimismo, los motores de búsqueda almacenan metainformación sobre la página que pudiera ser de interés (como intentar la identificación del idioma en el que está escrita la página web o si ésta tiene una licencia de uso específica).

### El algoritmo de Google

Google tiene un algoritmo llamado PageRank que ofrece una valoración de una página a partir del número de páginas que la enlazan y del PageRank de estas páginas. Así nuestra página personal tendrá mayor PageRank si nos enlaza a muchas páginas con un PageRank alto. El razonamiento que podemos encontrar en este algoritmo es que la importancia de la propia página web va en función de la importancia de las páginas web que la enlazan.

- El último paso tiene como objetivo mostrar la información de las bases de datos, de manera que sea útil para el usuario. Parte de esta tarea consiste en optimizar el tiempo de búsqueda para que la espera no sea muy larga.

### Ejemplos de motores de búsqueda

El más conocido de los motores de búsqueda es Google, aunque Yahoo! Search, AllTheWeb o Bing, por ejemplo, funcionan de una manera similar.



## 6. Web 2.0

Como se ha podido comprobar de la lectura de los últimos apartados, la web ha ido evolucionando desde un repositorio de información estática a principios de los años noventa hasta un servicio interactivo con el usuario, dinámico en su esencia y altamente cambiante. El nuevo escenario se ha venido a denominar **web 2.0**. Es importante entender que aunque el concepto de web 2.0 esté fuertemente vinculado con tecnologías, la web 2.0 en sí no es una tecnología, sino un modo de entender y usar la web. Por lo tanto, a pesar de que el nombre parece indicar que se trata de una nueva versión de la web, esto no es así, ya que no existe ningún RFC o documento similar que lo estandarice ni lo describa. La web 2.0 es, en realidad, una manera nueva de entender la web, mucho más centrada en el usuario final.

Este apartado pretende arrojar un poco de luz sobre todo esto.



Compendio de conceptos y fundamentos asociados con la idea de web 2.0. Fuente: Wikipedia. Licencia: Creative Commons Attribution ShareAlike 2.0.

## Ejemplos de webs 2.0

La red social Facebook, el archivo fotográfico Flickr, el servicio de blogs de Blogger, la cartografía de Google Maps, el correo electrónico etiquetado de Gmail, la opinión de los clientes acerca de los libros de Amazon, la realidad interacción generadora de confianza por parte de vendedores y compradores de eBay o la compartición de marcadores con páginas interesantes de Del.icio.us, son ejemplos de webs 2.0.

Dentro de los conceptos importantes que se atribuyen a la web 2.0 están los siguientes: plataforma, web de lectura y escritura y controlado por los usuarios.

La web 2.0 está formada por una serie de **plataformas tecnológicas**, que no dejan de ser aplicaciones web complejas. Al HTML original se le han añadido tecnologías que permiten introducir dinamicidad en el navegador y peticiones

asíncronas de nuevos datos. Recuérdese que antes, para obtener nuevos datos, había que realizar una nueva petición HTTP al servidor; ahora, existen tecnologías que permiten descargarse nueva información sin la petición explícita del usuario.

En los inicios de la web, la información que existía en el servidor no podía ser modificada, ahora, con la web 2.0 esto no es así. Un ejemplo de ello son los wikis, que permiten precisamente cambiar las propias páginas web, o los blogs, que suelen ofrecer la posibilidad de comentar las historias.

Finalmente, lo importante es el usuario final y no los contenidos. Esto se debe a que lo realmente difícil es captar nuevos usuarios a los sitios de Internet (el efecto red del que ya hemos hablado con anterioridad), mientras que la información interesante o útil se ha visto que es más fácil de crear.

### Ejemplo de efecto red

Un ejemplo paradigmático del efecto red viene dado por las redes sociales, en las que la Red gana en valor con cada nuevo usuario que se apunta; al fin y al cabo, uno no se apuntaría a una red social que no utilizara nadie de sus conocidos.

## 6.1. Páginas wiki

En este subapartado introducimos el wiki, un ejemplo de aplicación web desarrollada dentro de la web 2.0. En el esquema clásico de páginas web almacenadas en el disco duro del servidor, el cliente recibía contenidos estáticos. Mediante las páginas wiki, el cliente puede editar la propia página, modificándola y almacenándola de nuevo en el servidor.

Una **wiki** es una página web que permite ser editada mediante el navegador web.

La historia de las páginas wiki se remonta a 1995, cuando Ward Cunningham produjo la primera versión de un servidor WikiWiki.

Las ventajas de este sistema de edición "al vuelo" son evidentes: cualquiera que visita la página puede editarla, y permitir así la elaboración, la corrección y la mejora en común. Además de permitir la modificación de la página, los wikis permiten –salvo que esta funcionalidad esté inhabilitada o el usuario carezca de permisos para llevarla a cabo– la creación de nuevas páginas wiki. Por lo tanto, existe un mecanismo para que dentro de la página wiki podamos crear un enlace a una nueva página wiki.

### WikiWiki

El nombre WikiWiki viene de que en la lengua de las islas Hawaii *wiki* significa rápido.

Las páginas wiki cuentan con lo que se ha venido a denominar **vistas**, que no es otra cosa que varias representaciones de la propia página wiki o de información relacionada. El usuario puede elegir entre las diferentes vistas, utilizando normalmente las pestañas que existen en la parte superior de la página.



Página de la Wikipedia sobre el wiki. Fuente: Wikipedia: Licencia: GFDL y Creative Commons Attribution ShareAlike 3.0.

Las vistas más comunes son:

- La **vista tradicional** es la del artículo, en el que se muestra el contenido de la página.
- La **vista discusión** está pensada para la edición de wikis cooperativos, en particular de enciclopedias. Es un espacio en el que los que aportan contenidos pueden discutir y comentar la página wiki, por ejemplo, sobre la veracidad del contenido, si sigue la guía de estilo, etc.
- En la **vista de código fuente** se puede editar la página wiki. Se trata de un simple formulario en el que podemos introducir el texto que deseemos. Al pulsar el botón de envío, pasaremos a la primera vista y podremos ver de manera inmediata los cambios que hemos realizado.
- Finalmente, contamos con la **vista de historia**. Mediante la historia podemos trazar los cambios que ha sufrido la página, incluyendo información sobre quién los ha realizado, cuándo fueron realizados y qué líneas fueron cambiadas. Además de poder ver cómo ha ido evolucionando la página wiki, permite deshacer cambios y volver a versiones anteriores de la página wiki.

Como se ha comentado con anterioridad, en la vista de edición de una página wiki podemos modificar la propia página. En los inicios, y todavía en algunos wikis, el código de las páginas wiki se escribía en HTML. Sin embargo, pronto se alzaron muchas voces en contra del uso del HTML y a favor, de una manera más sencilla de editar. Esto es debido a que el estándar HTML, aunque amplio, a veces es farragoso y ciertamente complejo para usuarios no técnicos. A su vez, el HTML permite algunas cosas que van en contra de la consistencia visual de la página wiki y, aunque de manera moderada, también introduce algunos riesgos a la seguridad.

En consecuencia, se ideó una sintaxis específica para los wikis, más sencilla de utilizar. El problema es que cada aplicación web de wikis utiliza sus propias convenciones ligeramente diferentes, aunque en los últimos tiempos podemos encontrar algunos esfuerzos por unificar las convenciones, de manera que las reglas sintácticas sean iguales. En la tabla siguiente se pueden ver algunas de las convenciones utilizadas en Wikimedia, la aplicación web utilizada por la Wikipedia. Como se puede observar, aun así existen algunos códigos que son HTML "puro".

Etiquetas wiki más frecuentes

| Estilo                                                          | Significado/uso   |
|-----------------------------------------------------------------|-------------------|
| <code>'''negrita'''</code>                                      | negrita           |
| <code>''itálica o cursiva''</code>                              | itálica o cursiva |
| <code>&lt;u&gt;subrayado&lt;/u&gt;</code>                       | subrayado         |
| <code>&lt;span style="color:green"&gt;verde&lt;/span&gt;</code> | verde             |
| <code>&lt;center&gt;Centrado&lt;/center&gt;</code>              | Centrado          |
| <code>[[Enlace]]</code>                                         | Enlace            |
| <code>[[Enlace Texto visible]]</code>                           | Enlace            |
| <code>== Sección 1 ==</code>                                    | Sección           |
| <code>=== Sección 1.1 ===</code>                                | Subsección        |
| <code>==== Sección 1.1.1 ====</code>                            | Apartado          |
| <code>[[Imagen:Isaac.Asimov02.jpg Isaac Asimov.]]</code>        | Imagen            |

Fuente: Wikipedia

El wiki más conocido es la enciclopedia **Wikipedia**, iniciada en el 2001 con el objetivo de crear una obra de consulta cooperativa en múltiples idiomas y con contenidos libres.

La Wikipedia funciona como un proyecto autogestionado y tiene una serie de normas consensuadas por la comunidad.

#### Guía sobre la edición de páginas wiki

Una guía detallada sobre la edición de páginas wiki se puede encontrar en la ayuda correspondiente de la Wikipedia ([http://es.wikipedia.org/wiki/Ayuda:Cómo\\_se\\_edita\\_una\\_página](http://es.wikipedia.org/wiki/Ayuda:Cómo_se_edita_una_página))

#### Ved también

Veremos lo que son contenidos libres en el módulo "Cultura libre".

Al ser un wiki, cualquiera puede editar en la Wikipedia pasando a ser un **wikipedista**; de hecho, salvo excepciones, lo puede hacer de manera anónima o siendo usuario (o sea, creándose una cuenta). Existe una serie de usuarios, conocidos como **bibliotecarios**, que tienen permisos especiales para borrar artículos o bloquear páginas, pero no tienen mayor poder decisorio que los wikipedistas y deben seguir las normas adoptadas por la comunidad. Asimismo, existen **bots** (diminutivo de robot), programas que realizan tareas tediosas como la creación de páginas de manera automática a partir de datos del censo, de manera que todo pueblo incluido en el censo tenga una página en la Wikipedia.

Se ha debatido mucho sobre la calidad de los contenidos de la Wikipedia, en especial en comparación con las enciclopedias tradicionales elaboradas por especialistas en las materias que se presentan. Algunos estudios han demostrado que la calidad de la Wikipedia en términos generales es similar o mejor que los de la Enciclopedia Británica. Sin embargo, la posibilidad de que cualquiera pueda editar una página de la Wikipedia tiene también sus aspectos negativos. Uno de ellos es la existencia de **vandalismo**, que es como se llama cuando alguien introduce modificaciones incorrectas, inapropiadas u ofensivas. Si el vandalismo se sucede repetidamente, los bibliotecarios tienen la potestad de bloquear la página wiki, evitando que sea modificada.

## 6.2. Blogs

Un **blog** es un sitio web donde se incluyen **historias** o **entradas**<sup>6</sup>, artículos o textos que pueden incluir imágenes y otros elementos multimedia, en orden cronológicamente inverso.

<sup>(6)</sup>En inglés, *posts*.

### Blog

Hay personas que utilizan el término bitácora en castellano, pero el anglicismo se utiliza hoy en día ampliamente.

El concepto de blog viene de **weblog**, donde *log* significa diario en inglés. El carácter de diario es el que ha dado una cierta informalidad a los contenidos publicados en los blogs. En general los blogs ofrecen la posibilidad de comentar esos contenidos, de manera que los lectores de las historias pueden interactuar con el autor (o autores). Por esta razón, los blogs han tenido amplia aceptación más allá del ámbito personal, siendo utilizados profusamente en periodismo y por instituciones y empresas para comunicarse con sus lectores y clientes. Aunque generalmente los blogs suelen incluir textos, hay blogs de fotos (los **fotoblogs**) o de vídeos (los **videoblogs**).

A día de hoy cualquiera puede abrirse un blog en Internet y llevar su diario personal allí.

### Ejemplos de alojamientos blog

Hay muchos sitios que ofrecen alojar blogs de manera gratuita en Internet, como por ejemplo:

- Blogger
- WordPress

- Bitacoras.com
- LiveJournal

Aunque no hay normas sobre cómo escribir en un blog (al fin y al cabo, se supone que se trata de un espacio personal donde uno escribe sobre lo que quiere y cómo quiere), sí hay una serie de recomendaciones que uno debería seguir si quiere tener un blog interesante. Así, es importante darle pausas al lector, utilizando convenientemente párrafos y espacios, y usar listados y elementos gráficos. Un texto compacto y complejo, con párrafos grandes "invita" al lector a no seguir leyendo. La claridad y la simplicidad son reglas básicas de un buen blog.

La longitud de las historias en los blogs ha de ser necesariamente corta, salvo casos especiales como manuales por ejemplo. El tiempo de lectura de la pantalla al que está habituado el lector es muy breve, especialmente en un entorno como el de la web donde hay múltiples enlaces para navegar por páginas sin fin. Por eso es importante ir a la esencia rápidamente. Enlazar otras páginas web es una buena práctica para ser concisos y a la vez originales. No es buena práctica copiar párrafos de otra página web, incluyéndolos en nuestro blog. Lo recomendable es crear un hipervínculo a esa página, para que el lector si así lo desea la visite, y mostrar nuestra postura o comentario en nuestro blog.

La mayoría de los blogs ofrecen la posibilidad de utilizar el lenguaje HTML a la hora de elaborar historias o comentarios, permitiendo utilizar al menos un subconjunto de las etiquetas del estándar. Finalmente, aún siguiendo los anteriores consejos, el mayor hándicap de los blogs –como diarios que son–, es que requieren actualizarse con cierta frecuencia para ser interesantes.

Generalmente la página que vemos del blog es la unión de los contenidos creados por el autor, a los que se les da forma para que tengan un aspecto visual atractivo y práctico. Véase el caso de la figura siguiente, que muestra la página principal de un conocido blog.

Página principal de un conocido blog sobre tecnología y sistemas de información



Fuente: <http://www.enriquedans.com>. Licencia: Creative Commons Reconocimiento 2.5 España.

En aras de una mejor visualización, los títulos de las historias se encuentran en otro tamaño y color, incluyen una imagen con el nombre del blog en la cabecera de la página, e información adicional en el pie, entre otras muchas cosas. En un lateral de la página, podemos ver que la mayoría de los blogs cuentan con alguna manera de navegar por historias pasadas, de buscar en los contenidos del mismo o de enlaces a otros blogs que el autor considera interesante.

Uno de los aspectos más interesantes que ofrecen los blogs es la posibilidad de **exportar sus contenidos**. Todos los elementos decorativos son eliminados a la hora de exportar los contenidos de un blog. Recordemos que los contenidos no dejan de ser el texto (generalmente en HTML) escrito por autor para escribir una historia y algunos metadatos, como pudiera ser la dirección de la historia, el autor, etc.

Para facilitar el intercambio de historias, se toman los contenidos y se ofrecen de manera pública siguiendo un formato estandarizado. Hay varios de estos formatos, como **RSS** o **Atom**. La mayoría de los blogs ofrecen un enlace a una página con sus contenidos en estos formato.

En la anterior figura, el enlace está arriba a la derecha como "RSS" junto a un icono cuadrado de color naranja (el logotipo de RSS). Si el lector lo visita, podrá observar que se trata de una página de texto plano con etiquetas similares, pero ligeramente diferentes, a las que conocemos del HTML, como puede verse en la figura siguiente.

## RSS del blog de Enrique Dans sobre tecnologías y sistemas de información

```

<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" media="screen" href="/~d/styles/rss2spanishfull.xsl"?><?xml-stylesheet type="text/xsl" media="print" href="/~d/styles/rss2spanishprint.xsl"?>
<channel>
  <title>El Blog de Enrique Dans</title>
  <link>http://www.enriquedans.com</link>
  <description>Investigación y opinión acerca de los Sistemas y Tecnologías de Información</description>
  <pubDate>Tue, 07 Apr 2009 07:41:38 +0000</pubDate>
  <generator>http://wordpress.org/?v=2.7.1</generator>
  <language>en</language>
  <sy:updatePeriod>hourly</sy:updatePeriod>
  <sy:updateFrequency>1</sy:updateFrequency>
  <creativeCommons:license>http://creativecommons.org/licenses/by/2.0/</creativeCommons:license>
  <item>
    <title>Los nuevos influyentes, de Paul Gillin</title>
    <link>http://feedproxy.google.com/~r/ElBlogDeEnriqueDans/~3/iTY6yEjy6PQ/los-nuevos-influyentes-de-paul-gillin.html#comments</link>
    <comments>http://www.enriquedans.com/2009/04/los-nuevos-influyentes-de-paul-gillin.html#comments</comments>
    <pubDate>Tue, 07 Apr 2009 07:41:38 +0000</pubDate>
    <dc:creator>Enrique Dans</dc:creator>
    <guid isPermaLink="false">http://www.enriquedans.com/?p=6325</guid>
    <description><![CDATA[Los nuevos influyentes de Paul Gillin es la edición española de "Los nuevos influyentes" de Paul Gillin. El libro fue un éxito en su edición original: era uno de los primeros que se aventuraba a entrar en la experiencia de escribir el prólogo fue muy interesante, porque el libro produce en la mayoría de los bloques de la historia de los nuevos influyentes es un brutal ejercicio de democracia: para cada uno de los nuevos influyentes se ha escrito un capítulo. EL libro está disponible en LID Editorial" href="http://www.lideditorial.com/EL-libro-esta-disponible-en-LID-Editorial"]></description>
    <content:encoded><![CDATA[<p><a title="Los nuevos influyentes - LID Editorial" href="http://www.lideditorial.com/EL-libro-esta-disponible-en-LID-Editorial">Los nuevos influyentes</a> de Paul Gillin es la edición española de "Los nuevos influyentes" de Paul Gillin. El libro fue un éxito en su edición original: era uno de los primeros que se aventuraba a entrar en la experiencia de escribir el prólogo fue muy interesante, porque el libro produce en la mayoría de los bloques de la historia de los nuevos influyentes es un brutal ejercicio de democracia: para cada uno de los nuevos influyentes se ha escrito un capítulo. EL libro está disponible en LID Editorial" href="http://www.lideditorial.com/EL-libro-esta-disponible-en-LID-Editorial"]></content:encoded>
    <wfw:commentRss>http://www.enriquedans.com/2009/04/los-nuevos-influyentes-de-paul-gillin.html#comments</wfw:commentRss>
    <feedburner:origLink>http://www.enriquedans.com/2009/04/los-nuevos-influyentes-de-paul-gillin.html#comments</feedburner:origLink>
  </item>
  <item>
    <title>Dominios .tel, curva de difusión y valor añadido</title>
    <link>http://feedproxy.google.com/~r/ElBlogDeEnriqueDans/~3/Ref7pLnqSks/dominios-tel-curva-de-difusion-y-valor-anadido.html#comments</link>
  </item>
</channel>

```

RSS es, por tanto, otro lenguaje de marcado. Su filosofía es similar a la del HTML, aunque su finalidad es diferente. Con los conocimientos que ya tenemos sobre HTML, podemos ver que no es muy difícil entender el RSS de la figura anterior. Se pueden identificar fácilmente etiquetas para el título (<title>) del blog, para la dirección del mismo (<link>), el lenguaje en el que está escrito (<language>), la licencia (<creativeCommons:license>), así como los títulos (<title>) y contenidos de las historias (<content:encoded>).

Actualmente, algunos navegadores web ofrecen la funcionalidad de gestionar los RSS de los blogs que un usuario quiera leer con frecuencia. Además, existen páginas que también aglutinan los contenidos de varios blogs. Esto es muy útil si se quiere tener en un sitio centralizado blogs que comparten temática, como blogs de empleados de una empresa, de compañeros de clase o de un proyecto de software libre. El funcionamiento de estos sitios, conocidos como **planetas de blogs**, es el siguiente: un software toma los contenidos de los blogs del planeta exportados en RSS o Atom e integra las historias de todos los blogs como si fueran de un único blog, mostrándolas en orden cronológicamente inverso. La unión de la informalidad de escribir en un blog con la posibilidad de compartir tus experiencias y conocimientos con otros blogueros provoca que se creen comunidades virtuales muy interesantes, y muchos proyectos y empresas fomentan su uso porque potencian la difusión de información, que es difícil que se propague por los mecanismos habituales.

#### Ved también

Se hablará de las licencias Creative Commons en el módulo "Cultura libre".



### 6.3. Suites ofimáticas en línea

Las **suites ofimáticas en línea** están compuestas por los programas más comunes de las suite ofimáticas, principalmente un procesador de textos, una hoja de cálculo, un gestor de presentaciones, pero accesibles y operables mediante un navegador web.

El hecho de que la aplicación se ejecute en el servidor es muy interesante, ya que no sólo supone un ahorro al no tener que comprar el software, sino que también permite que varios usuarios editen un documento a la vez. Asimismo, el documento queda almacenado en el servidor y no en los discos duros locales de los usuarios. Esto tiene la ventaja de que se puede acceder a aquél desde cualquier ordenador con el sencillo requisito: que éste se halle conectado a Internet y tenga un navegador.

Pero también hay críticas a estos servicios. Los más escépticos argumentan que su uso es inseguro: al hecho de que se pueda perder la conexión o de que el servidor se pueda desconectar, debemos añadir que si estamos realizando una tarea crítica –y un informe para nuestra empresa lo puede ser–, lo estamos realizando con infraestructura ajena, con todos los problemas que esto puede traer (privacidad). Por otro lado, está la velocidad: ejecutar aplicaciones en el ordenador local suele ser mucho más rápido que hacerlo en remoto.

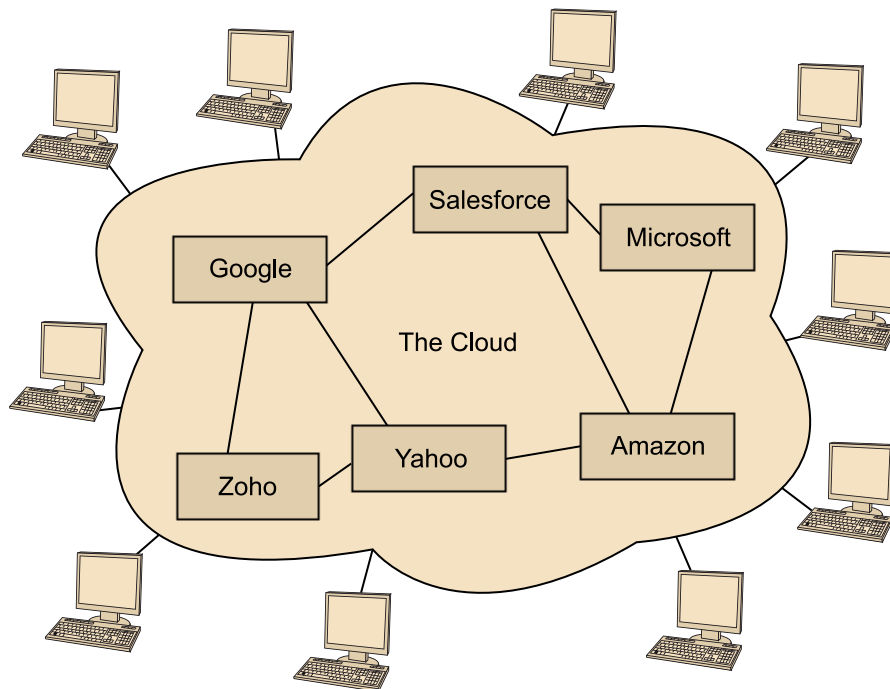
Las suites ofimáticas en línea son el ejemplo más popular de **la nube**<sup>7</sup>. El nombre se debe a que en los diagramas de redes de ordenadores se utiliza la metáfora de la nube para Internet, como se puede comprobar en la siguiente figura. La idea detrás de la nube es que se pueden utilizar servicios externos accediendo a ellos mediante un navegador web. Y dentro de estos servicios también podemos encontrar tareas que generalmente realizamos en nuestros ordenadores en local, como escribir un documento. En el caso de la nube, la infraestructura de estos servicios, su creación, puesta a punto y mantenimiento son llevadas a cabo por otros y no por nosotros mismos (o por nuestra empresa), por lo que el ahorro de costes puede ser importante.

#### Google Docs

El servicio más conocido de suite ofimática en línea es **Google Docs**. El servicio **Google Docs** se puede encontrar en <http://docs.google.com>.

<sup>(7)</sup>En inglés, *the cloud* o *cloud computing*.

### Una perspectiva de la nube



Fuente: Wikipedia: Licencia: Creative Commons Attribution ShareAlike 3.0 Unported.

## 6.4. Otros sistemas web 2.0

Además de las aplicaciones explicadas en los anteriores subapartado, existen otros ejemplos de sitios web 2.0. En este subapartado referenciamos los más relevantes. Notad que todas estas aplicaciones no son mutuamente excluyentes: puede haber webs que las incluyan todas, cada una para una finalidad distinta.

- Los **sistemas de gestión de contenidos** son aplicaciones web utilizadas para crear, editar, buscar y publicar textos y multimedia en Internet. Estos sistemas funcionan con una base de datos que almacena los contenidos y datos asociados a éstos. Un contenido podría ser una noticia, mientras que los datos asociados pueden ser comentarios a la noticia, información sobre el autor, la fecha, etc. Los sistemas de gestión de contenidos tienen funcionalidad para gestionar los contenidos de manera sencilla. Esto se debe a que, entre otras cosas, permiten editar los contenidos de manera fácil mediante páginas web, ofrecen la posibilidad de establecer un flujo de trabajo (indicando que el contenido es un borrador o definitivo) e incluyen políticas para tener varios roles (autor, revisor, etc.). La diferencia entre los sistemas de gestión de contenidos y los wikis radica en que mientras los primeros están pensados como un lugar para ofrecer información en la web, los segundos están abiertos a que cualquiera pueda modificarlos de manera sencilla. Así, los sistemas de gestión de contenidos son interesantes cuando se quiere ofrecer mucha información y en constante cambio (por ejemplo, un periódico o la web de un ministerio), mientras los wikis tienen como objetivo la creación de información de manera cooperativa.

- Los **sistemas de docencia electrónica** permiten crear y gestionar cursos en línea. Estos sistemas ofrecen funcionalidades como calendario, foros, entrega de trabajos, gestión de alumnos, gestión del currículum, etc. También pueden integrar otros elementos como los que se han presentado hasta ahora, como los wikis y los blogs. Desde un punto de vista formal, los sistemas de docencia electrónica se pueden considerar como sistemas de gestión de contenidos específicamente creados para entornos docentes.
- Los **servicios de redes sociales** permiten a los usuarios apuntarse, indicar quiénes son sus "amigos" (o contactos) y compartir con ellos mensajes, fotos, música, noticias, juegos y comentarios.
- A caballo entre las redes sociales y los blogs, podemos encontrar el servicio **Twitter**. Twitter permite compartir con tus contactos mensajes cortos de hasta 140 caracteres. Los usuarios generalmente lo utilizan para comentar sus experiencias y pensamientos y compartirlos con sus seguidores, que es el nombre que reciben aquéllos que se apuntan como contacto. Debido a lo limitado del espacio de cada mensaje y del carácter marcadamente personal e informal que suelen tener, se habla de **micro-blogging**.

#### Ejemplo de sistema de docencia electrónica

El Campus Virtual de la Universitat Oberta de Catalunya es un sistema de docencia electrónica, y uno de los más populares es **Moodle**, que tiene una licencia de software libre.

#### Ejemplo de servicios de redes sociales

Los servicios de redes sociales más populares actualmente son **Facebook** y **LinkedIn** –una red social orientada a contactos profesionales–, aunque hay muchas más.

## 7. La web multimedia

Hasta el momento, hemos visto que la web ha evolucionado mucho y que ha cambiado sensiblemente desde los tiempos en los que la información que recibía un cliente era un documento estático en un servidor. En el apartado dedicado al lenguaje HTML, hemos podido comprobar que se pueden incluir otro tipo de elementos que no sea texto en las páginas, en particular imágenes. Del uso diario de la web sabemos que existe otro tipo de elementos que también se pueden encontrar, como pueden ser vídeos o audio. En este apartado, vamos a conocer un poco más a fondo estos elementos.

**Multimedia** viene de la unión de *multum* (muchos) y *medium* (medios) y denota el uso de distintos medios para la información: audio, gráficos, vídeo, animación, etc.

### 7.1. Imágenes

Sabemos cómo se representa la información, en especial, la información en formato texto. Podríamos resumir que el método consiste en que cada una de las letras tenga una secuencia asociada de unos y ceros, o sea bits. De esta modo tenemos una codificación de caracteres, como el ASCII.

La cuestión que vamos a abordar ahora es cómo representar la información gráfica. Tomemos una imagen y supongamos que está compuesta de minúsculas partículas de colores únicos. La unión de todas las partículas nos dará la imagen. Cada partícula es lo que conoceremos como **píxel** y representará un color. La siguiente figura pretende clarificar esta idea: la imagen global está compuesta por píxeles de forma cuadrada, como se puede observar al aumentar una parte de la imagen. Cada píxel vendrá dado por un color. Desde lejos, la capacidad integradora del ojo humano provoca que la imagen no parezca estar compuesta por píxeles.

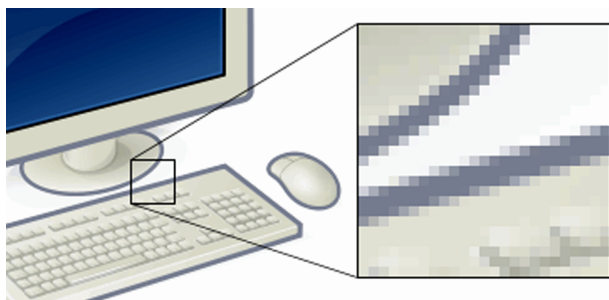
#### Ved también

La representación de la información y la codificación ASCII se explican en el módulo "Aspectos tecnológicos de los sistemas informáticos".

#### Pixel

*Píxel* es la contracción de *picture element*, elemento del dibujo.

Imagen con una parte aumentada



En la parte aumentada se pueden ver los píxeles de manera cuadrada que componen la imagen. Fuente: Wikipedia. Licencia: GFDL, y Creative Commons by-sa-2.5.

Supongamos, para simplificar, que estamos tratando una imagen en blanco y negro. Entonces, podríamos convenir que un píxel tendría un valor de 1 cuando fuera negro (N), y de 0 cuando fuera blanco (B). Cada píxel ocuparía un bit y la secuencia 101010 correspondería a NBNBNB. Si quisiéramos tener más colores, necesitaríamos darle a cada píxel más bits. Esto es conocido como la **profundidad de color**. La regla general es que con  $n$  bits podemos obtener  $2^n$  colores. En el caso de blanco y negro,  $n$  es igual a 1, ya que solo tenemos dos colores.

Con esta regla, podríamos codificar todos los colores que existen con un código único, como hacíamos con las letras. El problema es que el número de colores es infinito, por lo que no tendríamos bits suficientes. Por suerte, el ojo humano no es capaz de distinguir entre todos los colores. Llega un momento en el que un pequeño cambio en la tonalidad de un color es inapreciable para el ojo humano. Se conoce como **color verdadero** la codificación con 24 bits con la que podemos representar 16.777.216 colores.

Para terminar con la representación del color, queda una cuestión más. Todos los colores pueden representarse por una mezcla de tres colores, conocidos como **colores primarios**, que son el rojo, el verde y el azul. A partir de los tres colores primarios, podemos, por ejemplo, obtener el amarillo (sumando rojo puro y verde puro), el cyan (sumando verde puro y azul puro) y el magenta (sumando rojo puro y azul puro). La suma de los tres colores puros nos daría el blanco.

Si en vez de tomar los colores puros elegimos tonalidades, por ejemplo un rojo más débil, y lo sumamos con el verde, el amarillo que obtendremos será más débil que el amarillo puro. Con este mecanismo se puede obtener cualquier color a partir de los tres colores primarios y sus tonalidades.

El color de un píxel se puede dividir en tres partes de 8 bits, una parte por cada color primario. Con 8 bits podemos representar 256 tonalidades (recuérdese que  $2^8 = 256$ ). En otras palabras, tenemos 256 tonalidades de rojo, 256 tonalidades de verde y 256 tonalidades de azul, que podemos mezclar entre sí. Si quisiéramos obtener el azul puro, lo que deberíamos es tomar la tonalidad más pura del azul y nada del rojo ni del verde. Para el cyan, necesitaríamos las tonalidades más puras de verde y azul, pero nada de rojo. Sin embargo, si quisiéramos un cyan algo más claro, entonces habría que añadir algo de rojo.

Para obtener la imagen completa, tendremos una secuencia de píxeles, tantos como vienen dados por la **resolución**. La resolución es una matriz de A columnas por B filas de píxeles. Como las imágenes suelen ser rectangulares, la resolución se suele indicar en los siguientes términos  $A \times B$ , como  $800 \times 600$ . En otras palabras, la imagen consta de 800 píxeles de ancho por 600 píxeles de alto. En total, tendríamos 480.000 píxeles en la imagen, que es el resultado de

multiplicar el ancho por el alto. En las cámaras digitales la resolución viene dada en megapíxeles, que no es otra cosa que millones de píxeles. Así, una cámara de 5 megapíxeles puede crear fotos de hasta 2.560x1.920.

Finalmente, el **tamaño de la imagen** será el resultado de multiplicar la resolución por la profundidad de color:

$$\text{Tamaño imagen} = \text{Resolución} \times \text{Profundidad de color}$$

donde la profundidad de color nos da el tamaño en bits de cada píxel, mientras que la resolución indica el número de píxeles de nuestra imagen. Así, una foto de una cámara de 5 megapíxeles a color verdadero ocupa  $2.560 \times 1.920 \times 24$  bits = 117.964.800 bits o unos 14 megabytes.

Sin embargo, las fotos en el ordenador no suelen ocupar tanto. En el subapartado 7.3, al tratar de vídeo, hablaremos de una técnica conocida como **compresión**, que es la responsable de que una imagen ocupe menos de lo que hemos contado aquí. En la siguiente tabla se presentan algunos de los formatos de imágenes más habituales, indicando sus características más importantes.

Formatos de imagen

| Formato    | Características más importantes                                                                                                                                                     |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Raw        | Imagen "cruda" sin compresión.                                                                                                                                                      |
| BMP        | Formato de imagen de los iconos de Microsoft Windows. Admite compresión sin pérdidas.                                                                                               |
| TIFF       | Formato popular en escáneres (puede tener compresión con pérdidas).                                                                                                                 |
| GIF        | Formato muy popular en la web. Es de 8 bits (256 colores máximo) y permite crear animaciones.                                                                                       |
| PNG        | Formato diseñado para reemplazar el formato GIF. Los colores pueden codificarse con 8 bits (256 colores), 24 bits (4 millones de colores) y 48 bits (unos 256 billones de colores). |
| JPG o JPEG | Utilizado para fotografías e imágenes con muchos colores. Es un formato comprimido con pérdida de calidad ajustable.                                                                |

## 7.2. Audio

El **audio digital** es la codificación en digital de una señal eléctrica obtenida a partir del sonido.

Los humanos sólo tenemos capacidad auditiva en un rango específico, que va desde 20 hasta 20.000 Hz. Para obtener audio digital a partir de una señal eléctrica es necesario realizar dos pasos: el muestreo y la cuantificación.

El **muestreo** consiste en tomar muestras de una señal eléctrica con regularidad, tal y como se puede ver en la figura siguiente. Así, cada período de tiempo  $T$ , tomamos una muestra de la señal  $S(t)$  (básicamente, nos quedamos con el

### El sonido

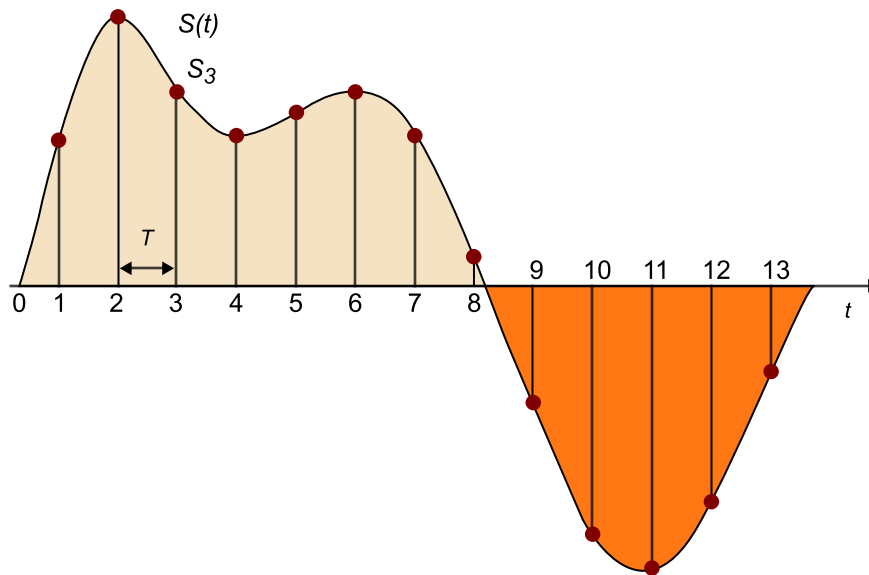
El sonido es en realidad una onda de presión.

### Hercio

Un hercio, o 1 Hz es equivalente a una variación de la onda de presión por segundo.

valor de la señal  $S(t)$  en ese instante de tiempo). Por ejemplo, en  $t = 3$ , el valor de la señal es  $S_3$ . La regularidad necesaria viene dada por la condición de no perder información, ya que en caso contrario la codificación es de baja calidad. Matemáticamente, se ha demostrado que es suficiente con muestrear con una frecuencia el doble de rápida que la frecuencia más alta de la señal eléctrica. Esto quiere decir que si los humanos escuchamos hasta 20.000 Hz, deberíamos muestrear como poco a 40.000 Hz. Por ello, la frecuencia de muestreo de los CD de audio es de 44.100 Hz.

Muestreo de una señal de audio



Fuente: Wikipedia. Licencia: Dominio Público.

El segundo paso es la **cuantificación**. Esta consiste en asignar un valor concreto al valor de la onda que se ha muestreado. Cuantos más bits usemos para representar este valor, más precisión habrá y, por lo tanto, más calidad. Supongamos que una señal de 20.000 Hz se cuantifica con dos bits, el 0 y el 1. Para los valores menores de 10.000 Hz se asigna un 0 y para los valores mayores de 10.000 Hz, un 1. Así, se podrían cuantificar todos los valores obtenidos mediante el muestreo del paso anterior. El problema es que con tan pocos bits la cuantificación sería muy mala, porque estaríamos dando un 0 tanto a un valor de unos pocos hercios como a otro de 9.999 Hz

La solución consiste en dedicar más bits a la cuantificación. ¿Cuántos? Pues depende de la calidad que deseemos obtener. Se ha comprobado que con 16 bits (con lo que obtenemos  $2^{16}$ , 65.536 valores de cuantificación diferentes) se puede obtener una calidad más que suficiente para una persona común. La cuantificación de 16 bits se utiliza en los CD de audio tradicionales, que almacenan el audio en formato digital. Al utilizar 16 bits, tenemos suficientes valores para muestrear los 20.000 Hz.

El **tamaño de archivo de audio digital** se calcula multiplicando el tiempo que dura por la frecuencia de muestreo por la cuantificación:

$$\text{Tamaño archivo audio} = \text{Duración} \times \text{muestras/segundo} \times \text{bits/muestra}$$

Mientras que la frecuencia de muestreo nos da el número de muestras por segundo, la cuantificación nos da el número de bits que utilizamos para cada una de las muestras. Con la multiplicación de estos dos factores, se obtiene el número de bits que ocupa un segundo de audio. Los CD de audio se suelen grabar en estéreo (o sea, que tenemos dos canales que muestrear y codificar), por lo que 1 segundo de audio ocupa 1.411.200 bits o 176.400 bytes.

Con un sencillo cálculo se puede comprobar que en un CD se puede grabar unos 70 minutos de música. Una sencilla operación matemática, que se deja como ejercicio, demuestra que el audio ocupa algo más de 700 megabytes. Precisamente lo que se podría grabar si se utilizara ese mismo CD para grabar datos.

La siguiente tabla presenta los formatos de archivos de audio más comunes y sus características más importantes. El lector ha de tener en cuenta que si graba canciones de audio en un CD de audio caben hasta un máximo de 80 minutos de canciones, pero que si las graba en formato MP3 le entran muchísimas más (como diez veces más). Esto también se debe a la compresión, que se explica en el siguiente subapartado. Además del popular formato MP3, existe otro formato de compresión de audio ampliamente utilizado: OGG. La principal diferencia entre uno y otro radica en que utilizan técnicas de compresión diferentes, siendo las del formato OGG un estándar libre y abierto (que no es el caso del MP3).

Formatos de audio

| Formato | Características más importantes                       |
|---------|-------------------------------------------------------|
| WAV     | Formato de audio sin compresión.                      |
| MP3     | Formato de audio con compresión con pérdidas.         |
| OGG     | Formato (libre) de audio con compresión con pérdidas. |

### 7.3. Vídeo digital y la compresión de datos

El **vídeo digital** viene dado por una sucesión de imágenes junto a una secuencia de audio. El número de imágenes (denominadas **frames** en la jerga) necesario para conseguir la ilusión de una imagen en movimiento debe ser, según algunos estudios, mayor de 15. En el cine, la velocidad es de 24 *frames* por segundo (fps), mientras que en la televisión –dependiendo del estándar utilizado– va de 25 a 30 fps.

El tamaño de una imagen de vídeo vendrá dado, por lo tanto, por el tamaño de cada uno de los *frames*, por el número de frames por segundo, por el tamaño del audio por segundo y todo ello multiplicado por la duración del vídeo.



$$\text{Tamaño archivo vídeo} = (\text{frames/segundo} \times \text{bits / frames} + \text{bits audio/segundo}) \times \text{duración del vídeo}$$

Si tenemos un frame de 1.280x720 píxeles, 25 fps, en color verdadero, y el audio es de una calidad como la del CD-audio (16 bits, 44.100 Hz de muestreo y estéreo). Un segundo de vídeo ocuparía más de 66 megabytes, ya que 1 segundo de vídeo ocupa debido al audio 0,17 megabytes, y debido a las imágenes, casi 66 megabytes. En un DVD de capa simple, cuya capacidad es de 4,7 gigabytes, apenas cabrían 74 segundos de vídeo.

Para reducir el volumen de datos aparece la **compresión de datos**. Para entender la compresión, se debe tener en cuenta que la información puede ser de tres tipos:

- **Redundante.** Información que se repite o que se puede inferir aunque faltara. Sea una imagen monocolor o de múltiples colores, el número de píxeles (resolución) y el número de bits para cada píxel (profundidad de color) no cambian. Sin embargo, al utilizar técnicas de compresión podríamos indicar de alguna forma que la imagen tiene píxeles repetidos y reducir tamaño.

#### Ejemplo de información redundante

El ejemplo más clásico de información redundante es una imagen que tiene todos los píxeles en blanco. Sin técnicas de compresión, esta imagen ocuparía lo mismo que cualquier otra imagen, ya que como hemos visto el tamaño de una imagen depende de su resolución y de la profundidad de color.

- **Irrelevante.** Se trata de información que no podemos apreciar.

#### Ejemplo de información irrelevante

Un ejemplo típico de información irrelevante se da en el audio, donde hemos visto que se suele muestrear para captar frecuencias de hasta 20 kHz. Y aunque existen personas con el oído tan fino como para escuchar variaciones a 20 kHz, lo usual es que la mayoría no llegue más allá de los 16 kHz. Si se eliminaran estas frecuencias, irrelevantes en general, el archivo ocuparía menos.

- **Básica:** Es la información que ni es redundante ni irrelevante.

Dependiendo del tipo de información que se utiliza, existen dos tipos de compresión: compresión sin pérdidas y compresión con pérdidas.

- La compresión **sin pérdidas** es aquella para la que podemos recuperar toda la información original a partir del archivo comprimido. Para realizar este tipo de compresión, se trabaja y elimina la información redundante.
- Por otro lado, la compresión **con pérdidas** es aquella para la que no podemos recuperar la información original a partir del archivo comprimido. Dentro de la compresión con pérdidas, debemos tener en cuenta que –aun habiendo pérdidas– puede que el receptor no se dé cuenta de éstas. Es lo que se conoce como **compresión con pérdidas, pero subjetivamente**

#### Ejemplo de compresión sin pérdidas

En el formato *zip* se utiliza la compresión sin pérdidas. De hecho, imaginemos un texto que, tras haber sido comprimido, tuviera pérdidas; sería ilegible.

**sin pérdidas:** estaremos eliminando la información redundante y la irrelevante.

- Si la compresión fuera con pérdidas y se eliminara también parte de la información básica, hablaríamos de **compresión subjetivamente con pérdidas**. Puede parecer algo raro un escenario en el que se pierda información relevante, pero esto es algo muy común, particularmente en el vídeo digital, debido a su gran tamaño. La razón detrás de una compresión tan fuerte es que muchas veces preferimos ver algo, aunque sea sin mucha nitidez pero de manera inmediata, a tener que esperar un rato para verlo con mayor calidad. Un vistazo a los vídeos del portal de compartición de vídeos YouTube convencerá a los más escépticos: la imagen de los vídeos no será la mejor, pero se valora más el poder verlos "al vuelo".

#### Ejemplo de compresión con pérdidas

El caso del audio con la eliminación de frecuencias por encima de 16 kHz es un ejemplo de compresión con pérdidas.

Dentro de la compresión de vídeo, se pueden diferenciar dos formas de comprimir:

- La primera es **compresión en el mismo frame**. Estas técnicas se asemejarían a la compresión utilizada en una imagen, aunque suelen ser más fuertes (entiéndase, con mayores pérdidas), ya que una imagen que sólo vamos a ver una fracción de segundo no necesita contener muchos detalles.
- La segunda es **compresión entre frames**. La razón de esta compresión es que un *frame* y el siguiente suelen tener mucho en común. Por ello, en vez de enviar dos frames diferentes, utilizando las técnicas de compresión en un mismo *frame*, se podría enviar un primer *frame* de mayor calidad y sucesivos *frames* con las diferencias con el primero. A este tipo de técnicas de compresión se las conoce también como compresión diferencial.

En la siguiente tabla se pueden ver algunos formatos de vídeo. Todos ellos son formatos con compresión con pérdidas, porque sino los vídeos ocuparían un gran volumen de datos.

Formatos de vídeo digital

| Formato   | Características más importantes                                                             |
|-----------|---------------------------------------------------------------------------------------------|
| DivX      | Formato de vídeo con compresión con pérdidas. Los ficheros a menudo tienen la extensión AVI |
| MPEG      | Formato de vídeo con compresión con pérdidas                                                |
| QuickTime | Formato de vídeo con compresión con pérdidas                                                |
| FLV       | Formato de vídeo en flash con compresión con pérdidas                                       |

## 7.4. Sistemas de compartición de imágenes, audios y vídeos

Uno de los hitos más importantes de la web 2.0 ha sido la aparición de sitios en los que los usuarios pueden compartir sus imágenes, sus canciones y sus vídeos.

Dentro del mundo de las imágenes –especialmente fotografías–, el lugar más popular actualmente es **Flickr**.

Flickr va más allá de ser un repositorio de fotografías con funcionalidades típicas de un sistema de gestión de contenido (añadir comentarios, puntuar las fotos, etc.). También se puede añadir información acerca de las fotografías (esta información sobre la información se denomina **metainformación**) mediante el uso de **etiquetas**<sup>8</sup>. De este modo, se pueden realizar búsquedas por palabras clave en Flickr que mostrarán fotografías con etiquetas relacionadas. Las etiquetas pueden incluir información del lugar, la fecha y la licencia de uso (en particular, en Flickr se puede indicar si la fotografía tiene una licencia Creative Commons).

<sup>(8)</sup>En inglés, *tags*.

### Ved también

Trataremos las licencias Creative Commons en el módulo "Cultura libre".

En lo que se refiere al audio, existen múltiples proyectos de radio en Internet que emiten "en vivo". En la jerga informática, esto se conoce como emitir mediante **streaming**: no se descarga un archivo para ser escuchado posteriormente, sino que se emiten datos que son reproducidos mientras éstos llegan. Existen múltiples radios en Internet que ofrecen un servicio similar a la radio convencional. Pero fuera de las radios, una comunidad interesante es **Jamendo**, un portal en el que los músicos se pueden dar a conocer ofreciendo su música bajo las condiciones de las licencias Creative Commons.

Finalmente, están los sitios de compartición de vídeo. Aunque hay muchos, el más conocido es **YouTube**. Estos sitios permiten subir vídeos en una serie de formatos (MPEG, AVI, etc.) que los transforman para poder ser vistos a baja calidad, alta calidad, incluso para móviles. Generalmente, la transformación consiste en bajar la resolución (o sea, hacer los frames más pequeños) y aplicar técnicas de compresión con pérdidas. La gran ventaja es que un usuario sin una gran conexión de Internet es capaz de ver los vídeos "al vuelo" sin tener que esperar mucho. Asimismo, los vídeos se pueden integrar de manera sencilla, mediante la inserción de una serie de etiquetas HTML que se presentan en un cuadro, en otras páginas web y en blogs. Por contra, en YouTube la inclusión de vídeos con contenidos de terceros ha provocado muchos problemas con los derechos de autor.

## Resumen

En este módulo hemos introducido la web, el servicio de Internet más conocido y usado. Para ello, se ha ofrecido una perspectiva histórica de su evolución, así como de su arquitectura cliente-servidor. También se han mostrado brevemente los protocolos más importantes sobre los que se asienta la web, básicamente HTTP –para la petición y respuesta de páginas web entre el cliente y el servidor– y el estándar HTML, mediante el cual se elaboran las páginas web.

Tras ver las limitaciones de las primeras aproximaciones de la web, se han presentado algunas nociones sobre la web dinámica e interactiva y cuáles son los fundamentos de las aplicaciones web. De entre las aplicaciones web, hemos visto en detalle varios tipos: los buscadores, los wikis y los blogs. De esta manera, el módulo finaliza presentando el concepto de web 2.0, mucho más centrado en el usuario, considerando interactividad y participación.

## Actividades

1. Cread un blog en alguno de los sitios indicados en el texto. Probad a editar una entrada de presentación en HTML, incluyendo imágenes y estilos de carácter (negrita, itálica, subrayado, etc.).
2. En los últimos tiempos hemos podido presenciar una amplia polémica entre los blogs y el periodismo tradicional. Recopilad argumentos a favor y en contra de ambas posturas.
3. Crearos una opinión sobre la siguiente afirmación: "la web 2.0 no aporta nada nuevo sobre lo que ya existía", y defended vuestra postura.
4. Discutid la aportación y riesgos (de funcionamiento, legales, etc.) para una empresa de utilizar aplicaciones web en vez de aplicaciones instaladas en sus ordenadores.
5. Cread una página wiki en WikiOle. Intentad introducir código HTML y código estructurado. Comprobad la diferencia entre el código estructurado de WikiOle y el utilizado en Wikimedia, que es el presentado en este texto.
6. Buscad en DMoz por páginas sobre los Juegos Olímpicos de Pekín 2008 sin utilizar el formulario de búsquedas (o sea, navegando por la jerarquía de conceptos).
7. Utilizad la búsqueda avanzada de Google para encontrar páginas web que hablen de la sección de baloncesto Fútbol Club Barcelona, pero no de la de fútbol. Utilizad para ello elementos de búsqueda avanzada, tales como los descritos en la Ayuda de Google.
8. Cread una página web personal en HTML utilizando un editor de textos. La página ha de incluir vuestro nombre como título, una foto y unos dos párrafos de currículum. Una vez creada la página, aprobadla con el servicio de validación del W3C.

## Ejercicios de autoevaluación

1. Nos hemos comprado un marco electrónico de los que nos permiten mostrar varias fotos digitales en nuestra mesilla de noche. Las dimensiones de las fotos son de 2.000x1.500 píxeles. En las instrucciones leemos que si pasamos fotos en formato BMP (con color verdadero) por el puerto USB, el software integrado en el marco automáticamente las comprime en JPEG y pasan a ocupar una décima parte de su tamaño original. ¿Cuántas fotos caben si la capacidad de la memoria incluida en el marco es de 120 megabytes?

2. Queremos crear un sintetizador de sonido sencillo para utilizarlo en las maquetas de nuestro grupo. El sintetizador será capaz de producir las siete notas musicales (do, re, mi, fa, sol, la, si) en 8 octavas (o sea, que podrá crear ocho veces las siete notas). ¿Cuántos bits necesitáis para codificar las notas? Si cada segundo de nuestro sintetizador puede emitir hasta 1.000 notas diferentes, ¿qué tamaño en kilobytes tendría el fichero de una canción de 20 segundos? Un amigo nos comenta que podríamos guardar esos mismos datos en un fichero de menor tamaño sin que se pierda calidad. ¿Es eso cierto? ¿Se os ocurre alguna manera de hacerlo? ¿Cómo?

3. En nuestro viaje de interrail pasamos por la bella ciudad de Praga, y para inmortalizar la visita nos hemos llevado nuestra flamante cámara que hace fotos de hasta 10 megapíxeles (con color verdadero) y que cuenta con una memoria de hasta 1GB. En un principio, pensamos en hacer fotos de 10 megapíxeles y no utilizar compresión a la hora de almacenar nuestras fotos. ¿Cuántas fotos podríamos hacer de esta manera?

Vemos que el número de fotos que se pueden hacer de esta manera es muy pequeño para una ciudad tan bonita como Praga. Uno de nuestros compañeros de viaje nos comenta que podemos hacer dos cosas:

- Utilizar compresión al almacenar las fotos. Miramos las instrucciones y vemos que nuestra cámara permite compresión de hasta un factor de 10 creando imágenes en formato JPEG.
- Configurar la cámara para que el número de megapíxeles de las fotos sea menor. Vemos que para obtener un número de fotos igual que con la otra solución, sería necesario bajarlo a 1 megapixel.

¿Cuál de las dos soluciones es mejor? ¿Por qué?

## Solucionario

### Ejercicios de autoevaluación

1. Para calcular cuánto puede ocupar una foto como máximo, debemos calcular primero el número de píxeles, luego multiplicarlo por el tamaño que ocupa cada píxel (a partir de la profundidad de color) y finalmente aplicarle el factor de compresión.

$$2.000 \times 1.500 = 3.000.000 \text{ de píxeles}$$

$3.000.000 \text{ de píxeles} \times 4 \text{ bytes/píxel (color verdadero)} = 12.000.000 \text{ bytes} = 12 \text{ megabytes}$  (hemos tomado para eso la aproximación de que un millón de bytes es un megabyte).

Si aplicamos el factor de compresión, cada foto puede llegar a ocupar 1,2 megabytes.

Si tenemos 120 megabytes de capacidad en el marco, podríamos tener hasta 100 fotos en él como mínimo.

2. Tenemos siete notas (do, re, mi, fa, sol, la, si) y ocho octavas, lo que en total nos da  $7 \times 8 = 56$  notas que queremos codificar. Para codificar 56 notas de manera unívoca, necesitamos 6 bits (lo que nos da la posibilidad de codificar hasta 64 cosas diferentes, lo que es suficiente para nosotros).

$$6 \text{ bits/nota} \times 1.000 \text{ notas/segundo} \times 20 \text{ segundos} = 120.000 \text{ bits.}$$

Entendemos 1 kilobyte como, aproximadamente, 1.000 bytes. En ese caso, el tamaño de 20 segundos de canción es de alrededor de 15 kbytes.

Lo que nos comenta nuestro amigo sobre compresión es, por lo tanto, cierto. Una manera de hacerlo es tomar la primera nota (6 bits) y, a partir de ahí, en vez de utilizar la codificación total de cada nota (que sería de 6 bits), utilizar una codificación relativa a la nota anterior de menos bits (por ejemplo, 4).

3. Primero debemos calcular cuánto ocupa una foto. Tenemos el número de píxeles que hemos de multiplicar por el tamaño que ocupa cada píxel (a partir de la profundidad de color), así que las cuentas son:

$$10.000.000 \text{ píxeles} \times 4 \text{ bytes (color verdadero)} = 40.000.000 \text{ bytes} = 40 \text{ megabytes (para ello hemos considerado la aproximación de que un millón de bytes es un megabyte).}$$

Si nuestra memoria es de 1GB (aproximadamente 1.000 megabytes), el número de fotos que podemos hacer es 25.

La primera solución es claramente mejor, porque al comprimir se aplican algoritmos que permiten almacenar más información en menos espacio.

## Glosario

**agregador de blogs** *m* Véase planeta de blogs.

**Apache** *m* Es el servidor web más popular, con una cuota de mercado superior al 60% desde hace años. Véase también Apache Software Foundation.

**Apache Software Foundation** *m* Fundación que se encarga de velar por el desarrollo y la promoción del servidor web Apache y de otros proyectos generalmente relacionados con tecnologías web. Más información en su página web.

**aplicación web** *f* Aplicaciones (programas software) a los que el usuario accede mediante un cliente web (navegador).

**Araña** *m* Programa que rastrea y almacena la web siguiendo los hipervínculos en las páginas web para su posterior procesamiento.

**atom** *m* Formato estructurado que se emplea para exportar los contenidos de blogs, entre otros. Véase también RSS.

**bitácora** *f* Véase blog.

**blog** *m* Sitio web en el que se insertan historias o entradas (en inglés *posts*), artículos o textos que pueden incluir imágenes y otros elementos multimedia, en orden cronológicamente inverso.

**buscadores** *m pl* Aplicaciones web que nos permiten encontrar información en Internet de manera rápida. Mediante la introducción de una o varias palabras clave, nos devuelve una página índice que enlaza páginas web con información relacionada.

**cliente** *m y f* Programa software que accede a un servicio ofrecido por un servidor. En el caso de la web, el cliente web es el navegador.

**color verdadero** *m* Dícese de aquella profundidad de color de al menos 24 bits. Con el color verdadero, se pueden representar al menos 16.777.216 colores.

**compresión** *f* Las secuencias de audio y vídeo ocupan mucho espacio, por lo que se utilizan técnicas de compresión que permiten disminuir el tamaño. De este modo se facilita su intercambio (el tiempo de descarga de un vídeo comprimido en formato DivX es menor que sin comprimir, y un vídeo que antes requería mucho espacio ahora cabe en un CD-ROM). DivX y MP3 comprimen con pérdidas, lo que significa que el archivo comprimido no tiene la calidad del original. Sin embargo, las pérdidas son –en muchas ocasiones y dependiendo del factor de pérdidas introducido– tolerables (incluso indetectables) para el ser humano. Existen formatos de compresión sin pérdidas, como el usado en los archivos zip, pero su utilidad es menor para audio y vídeo.

**cuantificación** *f* Proceso por el que se le asigna un valor binario a una muestra (véase muestreo).

**DivX** *f* Tecnología de compresión de vídeo basada en el estándar de compresión MPEG-4.

**fotoblog** *m* Blog de fotos.

**Google Docs** *m* Es una suite ofimática en línea de la empresa Google.

**hipertexto** *m* Documento de texto con enlaces (véase hipervínculos) a otros documentos.

**hipervínculo** *m* Conexión o enlace entre documentos.

**HTML** *m* Acrónimo de *HyperText Markup Language* (lenguaje de marcado de hiperTexto). Es el lenguaje en el que están escritas las páginas web.

**HTTP** *m* Protocolo de comunicaciones utilizado para intercambiar documentos de hipertexto entre dos ordenadores, generalmente un servidor y una máquina cliente. Acrónimo de *Hypertext Transfer Protocol*.

**índice temático** *m* Sitio web en el que las páginas web se encuentran ordenadas de manera jerárquica por temas.

**internet** *f* conjunto global de redes de ordenadores interconectados mediante el protocolo Internet Protocol (IP).

**la nube** *f* Paradigma de computación en el que los servicios se proveen por medio de Internet, sin que el usuario tenga percepción, conocimiento ni control de la infraestructura tecnológica de la que hace uso. Ejemplos de *cloud computing* son Gmail o Google Docs.

**lenguaje de marcado** *m* Lenguaje que permite marcar texto, de manera que se pueda añadir información adicional al contenido. Véase, por ejemplo, HTML.

**modelo cliente-servidor** *m* Arquitectura de computación distribuida, en la que una parte ofrece un servicio (el servidor) que es demandado por la otra (el cliente). La World Wide Web es un ejemplo de arquitectura cliente-servidor.

**motor de búsqueda** *m* Aplicación web que permite buscar por palabras clave en una base de datos de página web previamente almacenada por arañas.

**Mozilla** *m* Proyecto iniciado por la compañía Netscape a finales de la década de los noventa tras liberar su navegador Netscape Navigator. El programa más conocido de Mozilla es el navegador Firefox. Más información en su página web.

**MP3** *m* Formato de compresión para audio.

**muestreo** *m* Toma de muestras de una señal eléctrica con regularidad, para que posteriormente le sea asignada a cada muestra un valor (véase cuantificación).

**Netscape** *m* es la compañía que a mediados de la década de los noventa comercializaba el popular navegador Netscape Navigator. Cuando ya daba por perdida la *guerra de los navegadores* con el Internet Explorer de Microsoft, publicó el código fuente de su navegador bajo una licencia de software libre y fundó el proyecto Mozilla con el objetivo de remontar el vuelo. Fue adquirida posteriormente por AOL. Hoy el proyecto Mozilla es totalmente independiente de Netscape.

**palabras clave** *fpl* Palabras más importantes de un texto.

**página web** *f* Documento cuyo contenido está en formato HTML.

**planeta de blogs** *f* Programa software que toma contenidos de varios blogs (generalmente disponibles como RSS o Atom) y los muestra de manera integrada en una única página web como si se tratara de una nueva bitácora.

**profundidad de color** *f* Número de bits por cada píxel. Cuanto mayor sea la profundidad de color, más calidad tiene una imagen.

**resolución** *f* Número de píxeles de una imagen. Cuanta más resolución, más calidad tiene una imagen.

**RSS** *f* (de *Really Simple Syndication*, sindicación realmente simple). Formato estructurado que se emplea para exportar los contenidos de blogs, entre otros. Véase también Atom.

**servidor** *m* Puede referirse tanto al componente físico (hardware) como al componente lógico (software) que ofrece un servicio, como páginas web. Véase también cliente y Apache.

**Streaming (en español: flujo)** *m* Se trata de un modo de transmisión de datos entre cliente y servidor un poco peculiar (en relación con lo que es común en Internet). Fue ideado principalmente para la transmisión de datos multimedia. El flujo de datos ha de ser constante, para evitar interrupciones incómodas en audio y vídeo, y se ha de mantener la ordenación original de los datos, porque en caso contrario se dificulta la comprensión de audio y vídeo. El streaming suele utilizarse junto con técnicas de compresión.

**suite ofimática en línea** *f* Composición de los programas más comunes de las suite ofimáticas, principalmente un procesador de textos, una hoja de cálculo y un gestor de presentaciones, pero accesibles y operables mediante un navegador web.

**The Cloud** *m* véase la nube.

**videoblog** *m* Blog de vídeos.

**web** *f* véase World Wide Web.

**web 2.0** *f* Páginas y aplicaciones web que facilitan compartir información y la colaboración en Internet. Su diseño está más orientado hacia el usuario que hacia los datos, como sucede en la web "tradicional".



**Wiki** *f* Aplicación web que permite que la propia página web sea editada con el propio navegador web.

**Wikipedia** *f* Enciclopedia en línea totalmente libre que utiliza como infraestructura informática un wiki.

**World Wide Web** *f* Servicio de tipo cliente-servidor basado en el protocolo de comunicación Hypertext Transfer Protocol (HTTP), que sirve para intercambiar información entre ordenadores conectados a una red, generalmente Internet.

## Bibliografía

**Cobo Romaní, C.; Pardo Kuklinski, H.** (2007). *Planeta Web 2.0. Inteligencia colectiva o medios fast food*. ISBN 978-84-934995-8-7.

**Eguíluz Pérez, J.** (2008). *Introducción a XHTML*.

**Herrera, A. M.<sup>a</sup>; Villabona, J.** (2008). *15 años de Internet*. Madrid: Grupo Buho. ISBN: 9788493625986.

**O'Reilly, T.** (2005). *What is Web 2.0?*

**Rojas, O.; Alonso, J.; Antúnez, J. L.; Orihuela, J. L.; Varela, J.** (2005). Madrid: Editorial ESIC. *Blogs. La conversación en Internet que está revolucionando medios, empresas y ciudadanos*. ISBN: 84-7356-427-8.

**Tannenbaum, A. S.** (2003). *Redes De Computadoras* (4.<sup>a</sup> ed.). Madrid: Pearson. ISBN: 9702601622.

# Cultura libre

Gregorio Robles Martínez

PID\_00150269



Los textos e imágenes publicados en esta obra están sujetos –excepto que se indique lo contrario– a una licencia de Reconocimiento-NoComercial-SinObraDerivada (BY-NC-ND) v.3.0 España de Creative Commons. Podéis copiarlos, distribuirlos y transmitirlos públicamente siempre que citéis el autor y la fuente (FUOC. Fundació per la Universitat Oberta de Catalunya), no hagáis de ellos un uso comercial y ni obra derivada. La licencia completa se puede consultar en <http://creativecommons.org/licenses/by-nc-nd/3.0/es/legalcode.es>

# Índice

|                                                               |           |
|---------------------------------------------------------------|-----------|
| <b>Introducción.....</b>                                      | <b>5</b>  |
| <b>Objetivos.....</b>                                         | <b>6</b>  |
| <b>1. Definición de obra libre y sus consecuencias.....</b>   | <b>7</b>  |
| <b>2. Fundamentos de Propiedad Intelectual.....</b>           | <b>10</b> |
| <b>3. Tipos de obras según su licencia.....</b>               | <b>13</b> |
| 3.1. Obras de <i>copyright</i> estricto .....                 | 14        |
| 3.2. Obras bajo el dominio público .....                      | 14        |
| 3.3. Obras libres y <i>copyleft</i> .....                     | 15        |
| <b>4. Historia y definición del software libre.....</b>       | <b>18</b> |
| 4.1. El software libre en los inicios de la informática ..... | 18        |
| 4.2. El proyecto GNU y la definición de software libre .....  | 20        |
| 4.3. La aparición de Linux .....                              | 21        |
| 4.4. Evolución del sistema de desarrollo .....                | 22        |
| 4.5. El nacimiento de Open Source .....                       | 23        |
| <b>5. Licencias de software libre.....</b>                    | <b>24</b> |
| 5.1. Licencias permisivas .....                               | 24        |
| 5.2. Licencias <i>copyleft</i> .....                          | 25        |
| 5.3. Un ejemplo de funcionamiento de las licencias .....      | 26        |
| <b>6. Categorías de licencias.....</b>                        | <b>28</b> |
| <b>7. Licencias para otros tipos de contenidos.....</b>       | <b>31</b> |
| 7.1. Las licencias Creative Commons .....                     | 32        |
| <b>Resumen.....</b>                                           | <b>37</b> |
| <b>Actividades.....</b>                                       | <b>39</b> |
| <b>Ejercicios de autoevaluación.....</b>                      | <b>39</b> |
| <b>Solucionario.....</b>                                      | <b>41</b> |
| <b>Glosario.....</b>                                          | <b>42</b> |
| <b>Bibliografía.....</b>                                      | <b>45</b> |



## Introducción

En este módulo vamos a tratar el tema de la cultura libre. Según la Wikipedia, la cultura libre es "toda creación, y el movimiento que la promueve, que aboga por la elaboración y difusión de cultura de acuerdo a unos principios de libertad". Este movimiento se fundamenta en los cambios tecnológicos que hemos vivido en las últimas décadas, en especial para las **obras culturales que son información**. Cualquier tipo de información (ya sea texto, imágenes, audio o vídeo) puede ser transformada en símbolos binarios. Y contamos con poderosas herramientas para manejar la información en binario: los ordenadores.

En la actualidad, desde el punto de vista económico, crear la primera copia de una obra requiere mucho esfuerzo y dinero, mientras que el coste de copias sucesivas se puede considerar como marginal. Así, cuando un grupo de música graba un nuevo álbum, el proyecto para crear la primera copia requiere una inversión considerable (el equipo de música, los instrumentos, los técnicos, el estudio, etc.); sin embargo, una vez que tenemos la primera copia, obtener más copias del álbum es un proceso muy asequible.

Por otro lado, la difusión de Internet ha propiciado que el intercambio de información sea mucho más rápido y barato. Si hasta hace pocos años, la capacidad de hacer llegar una obra literaria estaba restringida a aquellos que tuvieran una imprenta y un canal de distribución, a día de hoy cualquiera con una conexión a Internet tiene la posibilidad de publicar una obra propia –pongamos por caso una novela– para un público amplio.

Cuestiones como éstas y otras relacionadas con la cultura libre serán tratadas en este módulo. Pero para entender el concepto de cultura libre, su finalidad y sus implicaciones hay que tener en cuenta una serie de aspectos adicionales. Empezaremos definiendo qué entendemos por una obra libre y ahondaremos en las consecuencias que se derivan de tal definición.

También es muy importante conocer el contexto legal en el que se mueven las obras culturales, la propiedad intelectual. Estudiaremos cómo la cultura libre utiliza las reglas de la propiedad intelectual para fines que no fueron los inicialmente previstos. Para ello profundizaremos en los términos de uso de las obras, que vienen indicados en licencias.

A continuación, se presentará cierto contexto histórico, principalmente la historia del software libre, indagando en las diferentes corrientes y licencias que existen en la actualidad y cómo están relacionadas con la cultura libre. Finalmente, conoceremos en detalle algunas licencias que se pueden utilizar para hacer que una obra, que no sea software, pueda ser considerada como libre.

## Objetivos

Al finalizar de leer este módulo, el estudiante deberá ser capaz de:

1. Conocer qué es una obra libre y qué no.
2. Entender los conceptos de propiedad intelectual más importantes que afecten a la cultura libre.
3. Entender conceptos relacionados con la cultura libre, en particular el de *copyleft*.
4. Conocer la historia del software libre, así como los tipos de licencias de software libre que existen.
5. Publicar una obra libre del tipo que sea utilizando una licencia para tal fin, por ejemplo, las licencias de Creative Commons.



## 1. Definición de obra libre y sus consecuencias

Una de las cuestiones esenciales a la hora de hablar de cultura libre es definir las características que ha de tener una obra cultural para que pueda ser considerada como libre. Aunque a día de hoy todavía es una cuestión debatida, una corriente mayoritaria ha partido de la filosofía del software libre –que veremos más adelante– y ha propuesto considerar como libres aquellas obras que cumplen las **cuatro libertades esenciales**.

Las **cuatro libertades esenciales** permiten que quien reciba una obra libre pueda:

- 1) Usarla y disfrutar los beneficios de su uso.
- 2) Estudiarla y aplicar el conocimiento obtenido de ésta.
- 3) Hacer y redistribuir copias, totales o parciales.
- 4) Realizar cambios y mejoras, y redistribuir las copias.

Nótese que, según esta definición, las obras se pueden distribuir con fines comerciales. Éste es uno de los aspectos que pueden resultar más sorprendentes del movimiento de la cultura libre. Por ello, una de las frases más difundidas por los promotores de la cultura libre dice que **libre no es igual a gratis**. En la raíz de todo esto se encuentra el ideólogo Richard Stallman que tiene una concepción de la libertad que se asemeja más a la libertad de expresión<sup>1</sup> que a la gratuidad de las obras (*not free beer*).

<sup>(1)</sup>En inglés, *free speech*.

### **Not free beer**

Una frase famosa de Richard Stallman en sus charlas es: "To understand the concept, you should think of *free* as in *free speech*, not as in *free beer*".

Esta confusión es especialmente importante en el ámbito anglosajón, ya que *free* es una palabra polisémica que admite como acepción tanto libre como gratis. La definición de "libre" que hemos usado en este módulo al hablar de obras libres corresponde a la primera acepción, mientras que la segunda, aun pudiéndose dar el caso de obras libres gratis, no tiene por qué darse siempre.

La **economía** de las obras libres es un tanto peculiar. Debido a su naturaleza digital, nos encontramos en un **mundo de abundancia** y no de carestía, como estamos acostumbrado en la economía tradicional, la de los objetos físicos. Esto es así porque, una vez obtenida una copia, realizar sucesivas copias es inmediato y barato.

En estas situaciones, cuando los costes de reproducción de la obra son mínimos, el precio de hacer una nueva copia tiende a ser el precio del soporte.

Los bajos costes de copias sucesivas no es el único factor que hay que tener en cuenta. En el mundo de la cultura libre, no sólo el autor original puede realizar copias; cualquiera puede hacerlo.

#### Ejemplo de costes de reproducción

Mientras que producir el primer CD del grupo que estaba creando un álbum tiene un precio muy alto, realizar una copia de éste una vez acabado viene a costar lo que cueste un CD virgen.

#### Ejemplo de un libro libre

Se puede dar el caso de que alguien, que llamaremos editor A, tome un libro libre (según la definición arriba expuesta) publicado en Internet, se lo descargue gratuitamente y decida editarlo por su cuenta y riesgo. Esto es posible, porque **la redistribución está permitida**. Además, el editor A puede poner el precio que desee al libro.

Si el precio que el editor A exige por el libro es muy alto, cualquiera podría hacer lo mismo y actuar como editor (convirtiéndose en el editor B), ofertando copias y ofreciéndolas en condiciones más ventajosas (por ejemplo, a un precio más bajo que el editor A). De esta manera, se tiene un mercado en el que la competencia hace que el precio tienda a ajustarse, hasta llegar idealmente al coste de lo que cuesta la reproducción. Y este precio, como se ha dicho antes, viene dado sobre todo por el precio del soporte: en este caso el coste de crear la edición en papel.

Otro de los efectos de una obra libre es que se pueden crear (y distribuir) versiones modificadas de las obras, incluso si no gustan al autor. Una obra modificada tiene carácter de obra nueva y se denomina una **obra derivada**. Esto permite que alguien tome un texto, como estos materiales, los amplíe y los vuelva a publicar. La traducción de obras, como novelas o películas, también supone la creación de una obra derivada.

El aspecto más polémico de esta libertad reside en los casos en los que el autor quiere que no se hagan versiones derivadas de su obra, ya que considera que ésta puede ir en contra de los objetivos que tenía la original. Tal es el caso de un ensayo del propio Richard Stallman, precursor del software libre y que utilizaba la analogía de la libertad de expresión para el movimiento de la cultura libre. Richard Stallman no permite crear versiones derivadas de sus ensayos, ya que considera que se podrían modificar de tal manera que la finalidad del ensayo modificado fuera precisamente la contraria para la que escribió el ensayo.

Una de las nuevas posibilidades que ofrece la cultura libre, especialmente en el mundo digital, es la de **mezclar varias obras en una nueva**. Este mestizaje ha sido muy frecuente en términos de estilos musicales durante toda la historia de la humanidad. Sin embargo, con obras libres se da un paso más, ya que la materia prima no son los estilos musicales, sino que pueden ser las obras en sí.

Finalmente, las obras libres permiten que muchas personas trabajen de manera conjunta, compartiendo sus mejoras. Hay varios escenarios en los que el trabajo cooperativo ha tenido lugar con el objetivo de conseguir una **mayor calidad**. Tal es el caso de Wikipedia o de algunas aplicaciones de software li-

bre que han conseguido organizar literalmente a miles de personas para que el producto final conjunto fuera mucho mejor que la suma de los esfuerzos individuales.

Para ser considerada como libre, una obra debe facilitar el uso de las cuatro libertades esenciales, por lo que, si hiciera falta, los **datos fuente** deben estar disponibles y la obra ha de estar en un **formato libre**. Esto significa que si el modo en el que se crea la obra es diferente a su estado de uso, entonces ambos se han de facilitar.

#### Ved también

Cómo se crea el software se ha explicado en el módulo "Aspectos tecnológicos de los sistemas informáticos".

#### Ejemplo de formato libre

El software se elabora en código fuente y luego se transforma a código binario, que es el que se ejecuta en un ordenador. Pero también podría darse el caso en una obra literaria que se ofrece sólo en papel o en un soporte informático difícil de manipular (por ejemplo, en un formato no editable); en los casos anteriores, aun cuando podríamos hacer uso de la libertad de modificación, ésta no sería completa ya que la dificultad que entraña es demasiado grande.

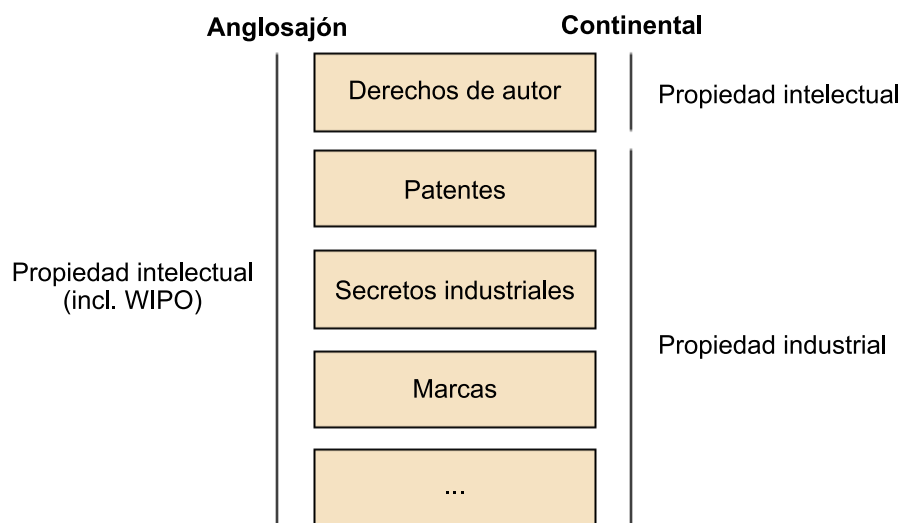
Por último, es indispensable que no se necesiten permisos especiales para crear herramientas que manejen las obras y que no haya limitaciones técnicas para el acceso a la obra. Esto se hace para evitar restricciones que no permiten hacer uso efectivo de las cuatro libertades, como patentes, derechos de privacidad, etc.

## 2. Fundamentos de Propiedad Intelectual

Las obras culturales, como fruto de un trabajo intelectual, se rigen bajo las leyes de propiedad intelectual. Conocer, por lo tanto, en qué consiste la propiedad intelectual, su filosofía, sus fundamentos y su funcionamiento es de capital importancia para entender el movimiento de la cultura libre. En este apartado tenemos ocasión de conocer los conceptos más importantes de la propiedad intelectual.

La propiedad intelectual reconoce en favor de un autor u otros titulares unos **derechos sobre obras del intelecto humano**. Es importante empezar haciendo una distinción, como se puede observar en la siguiente figura, entre el concepto de propiedad intelectual que se maneja en la Europa continental –el de este texto– y su uso en el ámbito anglosajón –incluida la WIPO–.

Diferencias entre la acepción de propiedad intelectual en el ámbito anglosajón y el continental



Para los anglosajones y la WIPO, la propiedad intelectual engloba la legislación sobre el *copyright*, las patentes, las marcas y los secretos comerciales, entre otros.

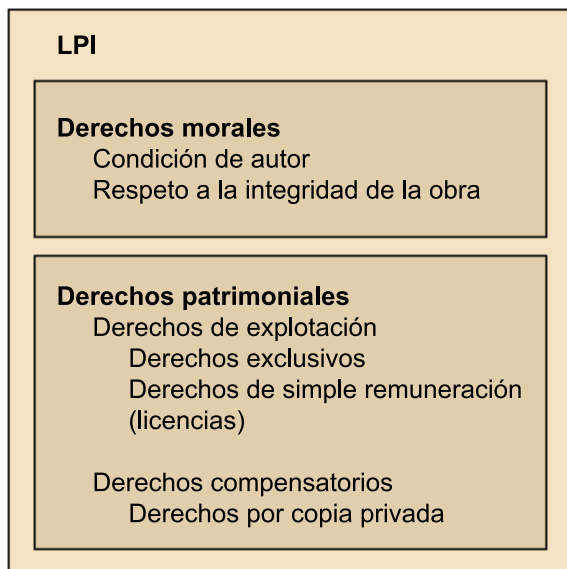
Sin embargo, en el derecho continental europeo, la propiedad intelectual se refiere a los derechos de autor, de los que el *copyright* es solamente una parte. La legislación sobre patentes, marcas y secretos comerciales, entre otros, forma parte de la denominada propiedad industrial. En este texto, salvo que se indique lo contrario, cuando hablemos de propiedad intelectual lo haremos refiriéndonos a la acepción continental.

### La WIPO

La WIPO, la World Intellectual Property Organization (Organización Mundial de la Propiedad Intelectual), es un organismo que forma parte de Naciones Unidas, cuyo objetivo es desarrollar un sistema de propiedad intelectual internacional.

Dentro de la propiedad intelectual, como muestra la siguiente figura de manera gráfica y resumida, podemos diferenciar dos tipos de derechos: los derechos morales y los derechos patrimoniales.

Esquema resumido y abreviado de la Ley de Propiedad Intelectual (LPI)



La legislación española (y continental) de propiedad intelectual diferencia entre derechos morales y derechos patrimoniales.

1) Los **derechos morales** no tienen generalmente limitación temporal y algunos son irrevocables. Dentro de éstos se encuentra el de la autoría; un autor por el mero hecho de haber creado la obra tiene derecho a ser reconocido como tal. Otros derechos incluidos son el respeto a la integridad de la obra, de manera que nadie la pueda modificar, o el de acceder a la única copia que quede de ésta (por ejemplo, una escultura). En la legislación anglosajona de propiedad intelectual, no existen los derechos morales como tales.

2) Los **derechos patrimoniales** son los que garantizan al autor la posibilidad de explotar económicamente su obra. Estos derechos están limitados en el tiempo (actualmente, la mayoría de las legislaciones lo limitan a 90 años después de la muerte del autor) y pueden ser cedidos. Se dividen en derechos de explotación y derechos compensatorios. En el ámbito anglosajón los derechos de explotación se denominan **copyright**, una acepción que es muy común en el resto del mundo y que por ello utilizaremos en este texto para hablar de los derechos de autor.

Como el derecho de explotación se puede ceder, con frecuencia no se halla en manos de los autores originales, sino de terceras personas o de compañías (que serán los **titulares de los derechos**).

#### Ejemplos de titulares de los derechos

Es conocido el caso de los derechos de la mayoría de las canciones de The Beatles, que fueron adquiridos en 1985 por Michael Jackson. En el 2007 se vendieron parcialmente a la empresa discográfica Sony. Los curiosos pueden encontrar más información al respecto en la web Snopes.

En la legislación española, dentro de los derechos patrimoniales, encontramos una serie de derechos denominados **derechos compensatorios**, entre los que se encuentra el popularmente conocido como **canon**. Así, los usuarios podemos realizar copias privadas de una obra a la que hemos accedido de manera legítima. La finalidad de la copia privada es permitir que si uno se compra un CD, pueda por ejemplo hacerse copias en casete para la radio del coche. En contrapartida, existe una remuneración compensatoria por copia privada que las entidades gestoras de los autores, como la Sociedad General de Autores y Editores (SGAE), recaudan a los fabricantes de equipos electrónicos y luego reparten entre sus asociados.

### 3. Tipos de obras según su licencia

En todo este entramado legal, ¿dónde se especifica qué se puede hacer y qué no con una obra? La legislación de propiedad intelectual supone que el autor<sup>2</sup> tiene un control legal casi absoluto de la obra. En líneas generales, puede decidir qué es lo que se puede hacer y qué no con su obra. Sin embargo, hay una serie de actividades que por su repercusión suelen ser las más importantes:

<sup>(2)</sup>Se define *autor* como el autor de la obra o a quien el autor haya cedido los derechos patrimoniales de la obra. En lo sucesivo, cuando hablemos del autor nos referiremos al que ostente los derechos patrimoniales.

- Si se puede **copiar** su obra. Es más, el autor podría permitir sólo un número determinado de copias, si así lo quisiera.
- Si se puede **distribuir** su obra y, si fuere el caso, bajo qué condiciones.
- Si se puede **exponer** (en el caso de un cuadro) o **representar** su obra (en el caso de música) públicamente.
- Si se pueden realizar **trabajos derivados**.
- Si se puede llevar a cabo cualquiera de los puntos anteriores con **ánimo de lucro**.

El control absoluto por parte del autor se ve limitado por algunas consideraciones, por ejemplo, la posibilidad de realizar copias privadas, tal y como hemos visto con anterioridad. Por ello, aunque el autor no nos dé permiso para copiar un CD podríamos hacerlo ateniéndonos a que se trata de una copia para uso privado. Asimismo, hay unos derechos que son necesarios para el uso y disfrute de una obra; de esta manera, cuando compramos una novela, el derecho a leerla viene implícito y el autor no puede reservárselo.

Se puede distinguir entre varios tipos de obras dependiendo de los permisos que dé el autor. Haremos uso de un diagrama de semáforo para mostrar qué se permite en cada caso. En el diagrama de semáforo, el color rojo indica que el autor no cede ese derecho, mientras que sí lo hace si está en verde. En el caso de estar en ámbar, se trata de un aspecto opcional que el autor ha de indicar explícitamente. Según la elección del autor, las obras que nos podemos encontrar son de los siguientes tipos:

- Obras de *copyright* "estricto".
- Obras libres, y dentro de éstas:
  - Obras bajo el dominio público.
  - Obras *copyleft*.
  - Obras no *copyleft*.

Para indicar qué derechos cede o comparte el autor con el usuario, existen las **licencias**. A partir de la lectura de una licencia, sabremos de qué tipo de obra se trata.

Las **licencias** son una especie de contrato con las condiciones de *uso* de la obra. Usando este mecanismo, el autor puede ceder permisos "automáticos" a otros.

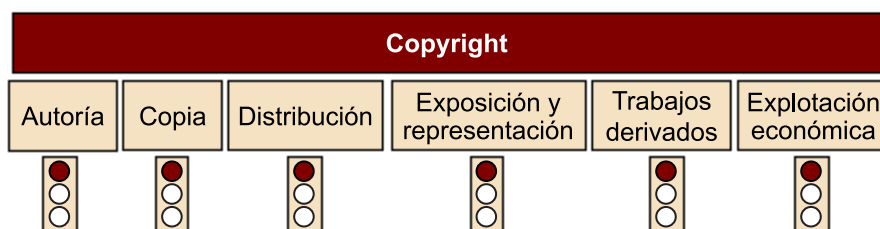
### 3.1. Obras de *copyright* estricto

Por defecto, si el autor no da su permiso no se puede copiar, redistribuir, modificar o ampliar una obra. Así se da paso a un régimen que llamaremos de ***copyright* estricto** en el que todos los derechos están reservados. Se trata del régimen tradicional del *copyright*, que no cede ninguno de estos derechos al usuario.

Por lo tanto, el usuario puede como mucho hacer uso de la obra y, en ocasiones, el uso que puede hacer es limitado.

La siguiente figura muestra el diagrama de semáforo para el *copyright* estricto.

Diagrama de semáforo para el *copyright* tradicional



La razón por la que en este texto hemos denominado a este tipo de obras como obras de *copyright* estricto es que usan el *copyright* siguiendo la filosofía para la que se concibió: el autor tiene unos derechos exclusivos sobre su obra de los que cederá el mínimo número de ellos. Veremos que los otros tipos de obras también se basan en los derechos de autor (y, por lo tanto, en el *copyright*), pero hacen un uso más *laxo* de éstos, ya que los autores ceden algunos de ellos.

### 3.2. Obras bajo el dominio público

Según la Ley de Propiedad Intelectual, el autor puede ceder derechos a otros de manera parcial o total. Hay autores que, llevando al límite las posibilidades de cesión, deciden renunciar a todos los derechos posibles y ofrecer su obra bajo lo que se denomina el **dominio público**. En ese caso, la obra formaría parte del dominio público como aquellas obras para las que los derechos de explotación han caducado.

Se ha de recordar que la cesión de derechos está limitada por la Ley de Propiedad Intelectual. De esta manera, cuando se dedica una obra al dominio público se ceden principalmente los derechos patrimoniales, pero no los derechos morales, entre los que hay algunos irrenunciables. Por ejemplo, la autoría es

#### Copyright

El lema que suele acompañar al signo de *copyright* es el de "todos los derechos reservados".

#### Ejemplo de *copyright* estricto

Algunas editoriales y discográficas se están planteando la posibilidad de que sólo se pueda leer un libro electrónico una vez o escuchar una canción ¡un número limitado de veces!

#### Caducidad de los derechos de explotación

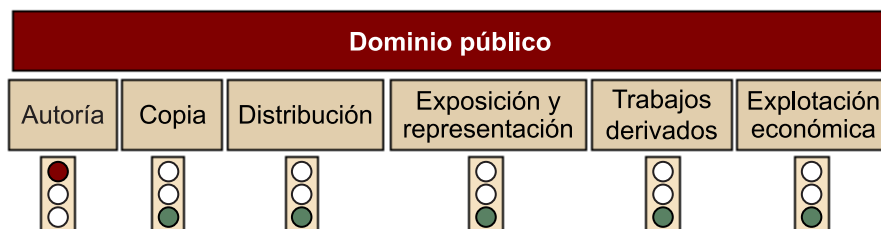
Los derechos de explotación caducan generalmente 90 años después de la muerte del autor.



un derecho irrenunciable aun cuando se dedique una obra al dominio público. De todas formas, existen maneras indirectas de "renunciar" a la autoría *de facto*, como publicar la obra de manera anónima o con un seudónimo.

En la siguiente figura se muestra cómo queda la cesión de derechos bajo el dominio público siguiendo nuestro esquema de colores de semáforo.

Diagrama de semáforo para el dominio público



En el dominio público se representaría todo de color verde, ya que posibilita copiarlo, distribuirlo, exponerlo y representarlo, realizar trabajos basados en la obra e incluso explotarlo económicamente. El caso de la autoría es un poco particular: en el ámbito anglosajón –al no existir los derechos morales– se puede tomar la obra y distribuirla o integrarla en otra sin reconocer al autor original; en el derecho continental, esto no es así.

Tanto en la jurisprudencia anglosajona como en la continental, no hace falta licencia si la obra está en el "dominio público" (ya sea si su autor la colocó ahí o porque expiró el período de derechos de autor). La razón por la que no es necesaria una licencia específica para el dominio público es que los términos en los que se pueden utilizar las obras vienen descritos de manera indirecta en la propia Ley de Propiedad Intelectual. No en vano, el dominio público implica que todos los derechos, salvo los irrenunciables e ilimitados, ya no tienen validez.

Habréis notado que una obra en el dominio público cumple con las condiciones que ha de tener para ser considerada libre. Esto no debería ser ninguna sorpresa; al fin y al cabo, permite hacer con la obra (casi) de todo.

### 3.3. Obras libres y *copyleft*

El dominio público es una de las posibilidades que existen para hacer que una obra se pueda considerar como libre. Sin embargo, hay autores que no quieren ceder tantos derechos como se hace al dedicar una obra al dominio público, sino sólo algunos. ¿Es esto posible y que la obra siga siendo libre?

Las obras de cultura libre son aquellas que permiten disfrutar de las cuatro libertades esenciales. Así, se podrían añadir restricciones a una obra siempre y cuando éstas sean admisibles, o sea, que no limiten esas cuatro libertades

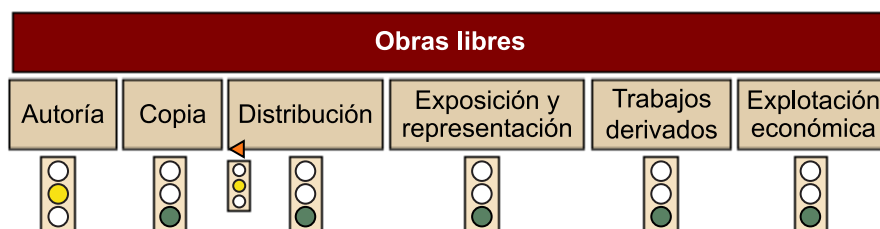
esenciales. Legalmente, las restricciones se indican en licencias. Por ello, se han creado licencias que garantizan las cuatro libertades, pero que tienen restricciones adicionales: son las **licencias libres**.

Así, entre las principales **restricciones admisibles** se encuentran las siguientes:

- 1) La atribución de la autoría.
- 2) La garantía de que las obras derivadas siempre serán libres (conocida como *copyleft*).
- 3) La protección de las propias libertades esenciales.

Siguiendo nuestro esquema de colores de semáforo, la siguiente figura muestra el esquema de colores de una obra libre.

Diagrama de semáforo para las obras libres



Tanto la copia como la distribución, la exposición pública y la representación, al igual que la creación de trabajos derivados y la explotación económica, han de estar garantizadas y, por lo tanto, se muestran en verde. Existen dos aspectos que el autor puede decidir no ceder, como son la autoría (aunque esto sólo es posible en la jurisprudencia anglosajona) y un aspecto adicional que se ha incluido en la caja de "Distribución" y que está relacionado con cómo se distribuyen las versiones derivadas.

Es aquí donde nos enfrentamos con un concepto novedoso: el **copyleft**, también conocido como de colaboración simétrica. El origen de la palabra *copyleft* viene de un juego de palabras en inglés, ya que su finalidad es conseguir lo contrario que el *copyright*. *Copyleft* se basa en la polisemia de "left" en inglés, que significa, por un lado, "izquierda" (en contraposición con derecha o derecho, de "right") y, por otro lado, "dejar" o "permitir" (del verbo *to leave*).

Las obras *copyleft* son obras libres con la peculiar característica de que las obras derivadas han de distribuirse bajo los mismos términos que las originales. Como los términos vienen indicados en una licencia, todas las obras derivadas (y las derivadas de éstas) deberán tener la misma licencia que la obra original, una **licencia copyleft**. Y así, como la obra original es libre, las obras derivadas también serán libres. Por esta razón, *copyleft* también se conoce como de

colaboración simétrica: todo el que quiera, puede participar en igualdad de condiciones, ya que tanto la obra original como sus derivadas tienen la misma licencia (o sea, las mismas condiciones de uso).

Con la estrategia del *copyleft* se consigue "anular" el *copyright* estricto, con lo que se crea un mundo en el que los derechos de autor son compartidos para siempre. Curiosamente esta práctica se sustenta en la legislación de propiedad intelectual y en el *copyright* que pretende dejar sin efecto. Así, se utilizan los derechos exclusivos que las leyes de propiedad intelectual le otorgan al autor para hacer que dejen de ser exclusivos. Se hace uso del *copyright* para dejarlo sin efecto, ya que se permiten usos (copia, distribución, modificación, etc.) que inicialmente no estaban previstos. El lema de esta triquiñuela legal es "all rights reversed" (todos los derechos volteados) en contraposición con el "all rights reserved" del *copyright* tradicional.

La flecha en la parte inferior izquierda en la caja de "Distribución" de la figura anterior se debe a que una obra libre puede tener una licencia con cláusula *copyleft*. El color ámbar de la flecha indica que este requisito es a elección del autor. Pero tanto si se permiten las acciones en las cajas en verde (copia, distribución, exposición y representación, trabajos derivados y explotación económica) como si no, una obra será libre, ya sea con una cláusula *copyleft* o sin ella. El *copyleft* es una característica opcional de una obra libre.

Por consiguiente, las obras *copyleft* son sólo uno de los tipos de obras libres que pueden existir. Y, sin embargo, mucha gente confunde el *copyleft* con la cultura libre y utiliza ambos términos indistintamente. Esto no es cierto, como hemos visto. Recuérdese que la cláusula *copyleft* es solamente una de las restricciones adicionales admisibles. Como veremos más adelante, hay obras libres que no tienen una licencia *copyleft*, y que permiten crear obras derivadas que no sean libres.

## 4. Historia y definición del software libre

Dentro del mundo de la cultura libre, el software libre se puede considerar la cresta de la ola. Esto es así porque se tienen más de 20 años de experiencia en este ámbito, con notables éxitos. De este modo, al menos en ciertos casos, hay soluciones basadas en software libre de alta calidad y que compiten con productos de gigantes de la informática como Microsoft, SUN, HP o IBM, con miles de empleados e ingenieros en nómina.

Alrededor del software libre se han creado enormes comunidades, que suelen comunicarse mediante medios telemáticos: listas de correo electrónico, foros, sitios web que apoyan el desarrollo (forjas), sistemas de notificación de errores, etc. Las ideas y las prácticas del software libre han ido poco a poco influyendo en otros campos intelectuales con sus debidas adaptaciones. Así, se puede entender el software libre como un espacio en el que encontrar experiencias de cultura libre.

### Ejemplo de forja

La forja en web más conocida en el mundo del software libre es SourceForge, aunque hay muchas otras.

En este apartado nos adentraremos en la historia del software libre, y presentaremos el contexto en el que surgió el concepto. A continuación, se introducirá la definición de software libre y el proyecto GNU, uno de los proyectos pioneros en la década de los ochenta. Veremos cómo Linux y las distribuciones entran en escena en la década de los noventa y cómo todo ello desemboca en que algunos promuevan el término Open Source en vez de software libre.

### 4.1. El software libre en los inicios de la informática

En los comienzos de la informática moderna, allá por la década de los cincuenta del siglo pasado, y hasta finales de los años sesenta, **el software era un componente que venía de regalo** con las grandes computadoras que se construían. En aquellos tiempos, sólo las grandes compañías y agencias gubernamentales se podían permitir la adquisición de ordenadores, cuyo precio era de muchos millones de dólares de la época. Los fabricantes de ordenadores basaban su modelo de negocio en la venta del ordenador completo, y veían en el software una manera de darle un valor añadido a su hardware. En los primeros tiempos las propias compañías productoras de ordenadores (hardware y software) promovían comunidades de desarrollo de software para sus plataformas hardware.

El hecho de que el software se desarrollara en aquellos tiempos de esa manera es algo lógico si tenemos en cuenta que la comunidad de científicos y académicos provenía principalmente del mundo de las matemáticas y de la física. En estos campos, lo normal cuando alguien demuestra un algoritmo matemático es publicarlo para que otros matemáticos puedan comprobarlo, e incluso construir sobre éste. Para estas personas, el software no dejaba de

ser una manera de tener algoritmos. Así, se puede considerar que en aquellos tiempos, aun cuando no existiera todavía una noción del software libre, lo era *de facto*: cualquiera podía usarlo, estudiarlo, modificarlo y redistribuir sus modificaciones. Es más, las propias empresas de ordenadores promovían este tipo de desarrollo para mejorar el rendimiento y la funcionalidad de los ordenadores que vendían, lo que permitía ampliar su negocio. Un ejemplo fue la comunidad SHARE auspiciada por IBM, siendo el nombre muy indicativo de las intenciones de IBM al respecto.

El gran cambio se produjo en la década de los setenta, cuando se empezó a vislumbrar que podría existir una industria del software por sí misma, independiente de la del hardware. Fue entonces cuando la concepción del software por parte de sus creadores cambió radicalmente: **el software pasó a ser un producto**. Así, nacieron compañías especializadas en la creación de software que estaban muy interesadas en proteger su inversión. Una de las maneras de hacerlo fue **distribuyendo al cliente sólo el código binario** del software, tratando el código fuente en el que trabajaban los informáticos como secreto industrial. De esta manera, sólo los empleados de la empresa podían realizar modificaciones y mejoras a un programa.

Otra manera complementaria de asegurarse una ventaja competitiva era tener un **monopolio legal** sobre su programa. Hubo entonces un amplio debate sobre si el software debía entrar dentro de la propiedad industrial (como patente) o de la propiedad intelectual (indicando los términos de uso en licencias). Finalmente, se decidió que el software se circunscribía bajo el régimen de propiedad intelectual. Así, como hemos visto con anterioridad, el autor tiene unos derechos exclusivos sobre su creación y podía compartir algunos derechos con sus usuarios.

El modelo de negocio de la industria del software tendió hacia un escenario en el que los creadores cedían el mínimo número de derechos, generalmente sólo el de uso del software –a veces incluso, de manera limitada–, reservándose la mayoría como parte de su activo de negocio. Había nacido el **software privativo**, también conocido como **software propietario**. En estos materiales, nos atenderemos a la denominación de software privativo, porque creemos que es más exacta, ya que se trata de un software que priva a los usuarios de algunas libertades. La denominación software propietario puede llevar a confusión, porque el software libre también tiene "propietarios"; recordemos que aunque éstos hayan cedido algunos de sus derechos, siguen siendo los legítimos "propietarios" del software como autores que son.

#### Ved también

En el módulo "Aspectos tecnológicos de los sistemas informáticos" hemos explicado los algoritmos, el código fuente y código binario, y la dificultad –sino imposibilidad– que tenemos los humanos para trabajar directamente con código binario.

## 4.2. El proyecto GNU y la definición de software libre

A mediados de los años ochenta, Richard Stallman, un investigador del laboratorio de Inteligencia Artificial de una prestigiosa universidad estadounidense (el MIT, Massachusetts Institute of Technology), decidió lanzar el **proyecto GNU** y una fundación, la **Free Software Foundation** (FSF), que lo promoviera. GNU tenía como objetivo crear un sistema operativo y herramientas de software totalmente libres, para que los usuarios pudieran usar, estudiar y modificar el propio software.

### GNU

GNU es el acrónimo recursivo de "GNU is Not UNIX", siendo UNIX el sistema operativo que se quería reimplementar.

Stallman, quien intelectualmente había crecido en un entorno en el que compartir software era natural, veía con incredulidad e incompreensión cómo la situación había cambiado en apenas unos años. Así, el intercambio de software, incluso en ámbitos científicos, dejó de ser premiado para pasar a estar muchas veces prohibido y perseguido judicialmente. El propio Stallman cuenta como anécdota que la decisión de iniciar el proyecto GNU se debió a que la impresora que tenían instalada en su departamento se atascaba con frecuencia. La compañía que la fabricaba se negó a dejarle el código fuente del software para que hiciera un simple cambio: que la impresora avisara a las personas que habían enviado un trabajo de impresión cada vez que eso ocurriera.

Stallman definió las condiciones que había de cumplir un software para ser considerado como software libre y lo formuló en el manifiesto del software libre.

Según el **manifiesto del software libre**, para que un software sea libre, el usuario ha de tener:

- 1) La libertad de **ejecutar** el programa, para cualquier propósito (libertad 0).
- 2) La libertad de **estudiar** cómo trabaja el programa, y **adaptarlo** a sus necesidades (libertad 1). El acceso al código fuente es una condición necesaria.
- 3) La libertad de **redistribuir copias** para que pueda ayudar al prójimo (libertad 2).
- 4) La libertad de **mejorar** el programa y **publicar sus mejoras**, y versiones modificadas en general, para que se beneficie toda la comunidad (libertad 3). El acceso al código fuente es una condición necesaria.

Se puede observar que la definición de obra libre tiene muchos parecidos con la que se acaba de presentar; no en vano, está basada en ella. Además, es importante darse cuenta de que la disponibilidad del código fuente no es por sí misma una de las libertades, pero no se puede hacer uso de algunas de ellas si no es posible acceder a éste.

Stallman decidió dejar su trabajo en el MIT para dedicarse al proyecto GNU, desarrollando programas libres desde cero. Durante los primeros años vendía copias de esos programas, que enviaba bajo petición por correo postal, de manera que fue uno de los primeros en ganar dinero con software libre. Como bien argumentaba, el software que vendía era libre porque aseguraba una serie de libertades al que lo recibía, pero no tenía por qué ser necesariamente gratis.

### 4.3. La aparición de Linux

A principios de los años noventa, un joven estudiante finlandés llamado Linus Torvalds empezó un proyecto propio para realizar el núcleo de un sistema operativo, la parte más básica del sistema operativo. Torvalds anunció una primera versión de **Linux**, que es como llamó a su programa, en un foro telemático en el que había interesados en la investigación y docencia de sistemas operativos. Publicó el código fuente e invitó a la comunidad a que le enviaran mejoras. Poco a poco, este pequeño proyecto personal fue transformándose en un proyecto con una gran comunidad de colaboradores que se comunicaban principalmente vía correo electrónico y mediante el intercambio de código fuente.

Linux se convirtió pronto en un éxito. Además, junto con las herramientas del proyecto GNU y otras existentes, hizo posible tener un entorno software completamente libre. Pero todas estas herramientas se encontraban desperdigadas por Internet y, en general, era tedioso encontrarlas, configurarlas e instalarlas en el ordenador personal. Para facilitar esta tarea, nacieron las **distribuciones de GNU/Linux**.

Las distribuciones de GNU/Linux tienen como objetivo ofrecer a los usuarios una manera fácil y homogénea de instalar (y actualizar) el software en sus ordenadores personales.

#### Ejemplos de iniciativas de distribuciones libres

Existen muchas iniciativas en el campo de las distribuciones de software libre: Debian –mantenida por cerca de un millar de voluntarios– o las distribuciones creadas por gobiernos, como en el caso de GNU/Linux en Extremadura. Una distribución que se ha hecho muy popular en tiempos recientes ha sido Ubuntu, cuyo principal propósito es ser fácil de instalar y usar.

Las distribuciones se pueden, en general, descargar de manera gratuita de la Red, pero hay algunas que se pueden adquirir también vía Internet o incluso en centros comerciales. Tal es el caso de la distribución de Red Hat, orientada al

#### GNU/Linux

Aunque popularmente se habla mucho de Linux, sería más preciso hablar de GNU/Linux, ya que Linux es sólo una parte del sistema operativo, habiendo sido gran parte del resto desarrollado por el proyecto GNU.

entorno empresarial. Bien es cierto que al comprar la distribución, por lo que se está pagando en realidad es por los manuales impresos que vienen en la caja, además de por la posibilidad de soporte telefónico ante dudas y preguntas.

Linux fue la pieza del puzzle que faltaba para tener un entorno completo libre. Ahora ya era posible usar un ordenador que únicamente tuviera instalado software libre.

#### 4.4. Evolución del sistema de desarrollo

Además de ser la pieza del puzzle que faltaba para tener un entorno completamente libre, Linux fue revolucionario en otro sentido: el organizativo. Algunos autores utilizan la siguiente analogía para explicarlo: mientras los antiguos modelos de desarrollo se asemejan a cómo se construían las catedrales medievales, el desarrollo de Linux se parecía más a cómo funciona un bazar.

El desarrollo de software tradicional, el de **catedral**, se rige por los siguientes aspectos:

- 1) Jerarquías sólidas: la estructura del equipo de desarrollo está bien definido. Cada uno tiene sus áreas y funciones. No está bien visto salirse de esta estructura.
- 2) Procesos bien definidos: el proceso de desarrollo está claramente especificado, desde que se capturan las necesidades del cliente hasta que se le presenta el programa terminado.
- 3) Largos plazos de desarrollo entre versiones: el proceso lleva su tiempo. Desde que el cliente indica sus necesidades hasta que recibe el software pueden pasar meses, incluso años.

Torvalds introdujo un estilo mucho más flexible, denominado **bazar**. Esto es así porque el desarrollo de Linux tiende a ser como un mercadillo en el que se intercambian de manera dinámica ideas y soluciones:

- Lo más transparente posible: todo se muestra y se discute en la lista de correo electrónico y cualquiera puede proponer modificaciones.
- Los plazos entre versiones son muy cortos: para que un número mayor de personas pueda probar las nuevas funcionalidades y encontrar errores más rápidamente.
- Se anima a que todo el mundo participe, formando una comunidad.

Las ventajas del modelo de bazar sobre el de catedral radican en que se fomenta la realimentación y la prueba por parte de los usuarios. En entornos de catedral, desde que se capturan los requisitos hasta que el usuario ve el programa

#### Ved también

En el módulo "Aspectos tecnológicos de los sistemas informáticos" vimos un modelo basado en procesos bien definidos: el modelo de desarrollo en cascada.



final, puede pasar mucho tiempo. En un entorno de bazar, al ser transparente y participativo, el usuario puede en todo momento implicarse, incluso dirigir el desarrollo. Evidentemente, el uso de herramientas de Internet facilita esta labor.

El modelo de desarrollo en bazar es una forma de crear software que el movimiento del software libre ha adoptado como propio. A veces esto ha causado equívocos, ya que mucha gente piensa que para que algo sea software libre se debe hacer siguiendo el modelo de bazar. Pero esto no es cierto; para que un software sea libre sólo ha de cumplir con las cuatro libertades.

#### 4.5. El nacimiento de Open Source

A finales de los años noventa, había una gran cantidad de soluciones basadas en software libre y la industria empezó a mirarlo con interés. Sin embargo, según algunas figuras importantes de la comunidad, el hecho de llamarse *free software* no resultaba conveniente, ya que daba a entender que era software gratis.

Por eso propusieron un nuevo término, desde su punto de vista más "comercial" o cercano a la industria: el **Open Source Software**, que podríamos traducir como software de fuentes abiertas. Asimismo, publicaron una definición con diez puntos que un software ha de cumplir para ser considerado "Open Source" y crearon una asociación, la **Open Source Initiative (OSI)**.

La promoción del término *Open Source* ha llevado a mucha confusión, porque puede entenderse que un programa que es software libre no es *Open Source* y viceversa. Esto no es cierto. Salvo en contadas ocasiones, en las que la FSF y la OSI no se ponen de acuerdo en un programa concreto por una cláusula menor en su licencia, **software libre y Open Source son sinónimos**. Es posible que filosóficamente podamos encontrar diferencias entre los que promueven un término u otro. La FSF tiene una componente principalmente ética, mientras que la visión de la OSI es más empresarial. Nótese que estas diferencias no quedan plasmadas en los programas libres, ya que estos cumplen tanto la definición de software libre como la de *Open Source*.

## 5. Licencias de software libre

En más de veinte años de historia del software libre han surgido multitud de licencias, hasta el punto en que la proliferación de licencias se ha convertido en un problema. Por ello, dedicaremos los siguientes subapartados a conocer en qué consisten estas licencias. Asimismo, se mostrarán con un ejemplo las implicaciones más importantes que aquéllas tienen. Para finalizar el tema, se hará una categorización de las licencias y los conceptos introducidos en éste y en apartados anteriores.

### 5.1. Licencias permisivas

Las licencias permisivas son aquéllas en las que el autor retiene el *copyright* para excluir garantía y para exigir la atribución de autoría, tanto en su obra como en obras derivadas. El autor permite la redistribución y la modificación de la manera que se considere oportuna, siempre y cuando se cumplan los requisitos anteriores.

#### Garantía del software

Es común, también en el software privativo, que el software venga sin ningún tipo de garantía (hasta el mínimo exigido por ley).

Por ser tan breves y tener tan pocas condiciones, las licencias permisivas también reciben el nombre de **licencias minimalistas**. Las licencias más conocidas en este ámbito parten de universidades americanas como Berkeley (existen las licencias BSD, de *Berkeley Software Distribution*) o MIT (del Massachusetts Institute of Technology). Esto es debido a que estas universidades publicaban con estas condiciones el software fruto de proyectos de investigación financiados por el Gobierno federal estadounidense. La razón por la que no ponían mayores impedimentos ni retenían un mayor número de derechos es que consideraban que el Gobierno ya había sufragado los gastos y que cualquier ciudadano podía beneficiarse de su trabajo al ser contribuyente.

### Ejemplo de licencia permisiva

Copyright (C) Universitat Oberta de Catalunya. All rights reserved.

Se permite la redistribución y el uso en formato fuente y binario, con modificaciones o sin ellas, siempre y cuando se cumplan las condiciones siguientes:

- Las redistribuciones de código fuente tienen que incluir el aviso sobre el *copyright* anterior, esta lista de condiciones y la renuncia de responsabilidad que se muestra más adelante.
- Las redistribuciones en formato binario tienen que incluir en la documentación u otros materiales que se proporcionen con la redistribución el aviso sobre el *copyright* anterior, esta lista de condiciones y la renuncia de responsabilidad que se muestra más adelante.
- No puede usarse ni el nombre de la universidad ni los nombres de los colaboradores para promocionar o promover productos derivados de este programa sin su consentimiento previo por escrito.

Con este tipo de licencias pueden crearse incluso obras derivadas privativas, ya que se puede redistribuir de cualquier manera (ya sea en fuente o en binario). Si un programa se publica con una licencia permisiva, éste será libre, ya que cumple con las cuatro libertades del software libre. Pero las obras derivadas no tienen por qué ser software libre; el que ha realizado las modificaciones puede redistribuir esta nueva obra bajo los términos que crea conveniente, siempre que cumpla con los requisitos mencionados arriba: que se mantenga a los autores originales en el *copyright*, que se muestre la exclusión de garantía y que no se utilice el nombre de los autores originales para promocionarlo sin permiso.

## 5.2. Licencias *copyleft*

En las licencias **copyleft** (también conocidas como **recíprocas** o **robustas**), el autor retiene el *copyright*, pero permite la redistribución y la modificación con la condición de que la redistribución sea bajo los mismos términos.

### Ved también

Hemos hablado del término *copyleft* en el subapartado 3.3.

En otras palabras, la redistribución se ha de hacer usando la misma licencia. Se trata, por lo tanto, de una licencia más restrictiva que las licencias permisivas, pero que asegura que las versiones derivadas de un software sean siempre libres. Por esta razón, muchos las consideran como una manera de defender activamente las libertades. Existen varias maneras de incluir una cláusula *copyleft* en una licencia, aunque todas comparten el mismo esquema: la distribución de cualquier versión del programa ha de realizarse con esta misma licencia.

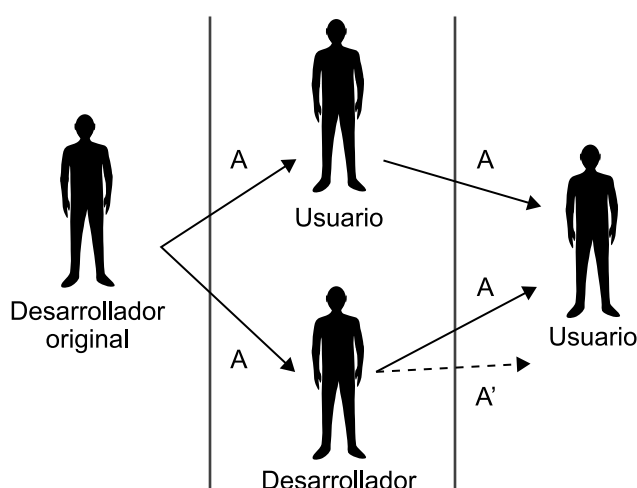
El razonamiento detrás del copyleft es que la Free Software Foundation consideró insuficiente otorgar las cuatro libertades únicamente al receptor del software libre. Pero con el *copyleft* se aseguran de que cualquier usuario que reciba una copia o una versión derivada del programa tenga las mismas libertades.

De entre las licencias *copyleft*, la más conocida es la **GNU GPL** (GNU General Public License, la Licencia Pública General de GNU). Esta licencia fue ideada por Richard Stallman con ayuda de varios abogados y estaba pensada para software del proyecto GNU. Se calcula que alrededor del 70% del software libre tiene una licencia GNU GPL, ya que se ha hecho muy popular y no sólo la utilizan los programas software del proyecto GNU.

### 5.3. Un ejemplo de funcionamiento de las licencias

Uno de los aspectos más complejos de las licencias de software libre viene dado por los actores que intervienen en el proceso de creación y (re)distribución, así como la posibilidad de crear obras derivadas y distribuirlas. La siguiente figura pretende aclarar este punto con un simple diagrama.

Diagrama con los diferentes actores, las distribuciones, la redistribución de la obra modificada, etc.



Tengamos por caso un desarrollador que crea una aplicación, que llamaremos la aplicación A, y que decide hacerlo con una licencia libre.

El desarrollador **no está obligado a publicar el software**, ya que la definición de software libre no le obliga.

#### Ejemplo del compilador ADA

Hay ejemplos en los que al creador no le interesa publicar su software en Internet. Tal es el caso de la empresa AdaCore, que basa su modelo de negocio en publicar una versión vieja del software que crea. AdaCore se encarga de desarrollar un compilador llamado GNAT para el lenguaje de programación ADA, muy popular en la industria aeroespacial. Ofrece la última versión de su compilador sólo a un número limitado de clientes con los que suele cerrar contratos comerciales, en general grandes empresas o instituciones del sector aeronáutico. Estas empresas podrían pedir el código fuente y redistribuir la aplicación y su código fuente, pero no suelen estar interesadas en ello. Al cabo de un tiempo, AdaCore publica el compilador en la Red, de manera que cualquiera puede descargárselo,

#### Ved también

Hemos visto lo que es un compilador en el módulo "Aspectos tecnológicos de los sistemas informáticos" cuando hablamos de cómo se crea software.

distribuirlo, modificarlo y redistribuir las modificaciones. Con esta estrategia, AdaCore combina la posibilidad de ofrecer un servicio especial por el que cobrar con los beneficios de tener una comunidad alrededor de su compilador.

Ya sea con código fuente o no, según las libertades del software libre, los usuarios pueden (pero no tienen por qué) redistribuir la aplicación *A* a otros usuarios. Por otro lado, en caso de difusión, el usuario que recibe el software en la redistribución recibirá la aplicación *A* con los mismos derechos.

En el momento de la distribución de *A'* es cuando adquiere importancia en qué tipo de licencia se ha publicado el software:

- En el caso de que *A* hubiera sido publicada originalmente con una licencia *copyleft* como la popular GNU GPL, la aplicación *A'* debe distribuirse bajo las mismas condiciones (o sea, bajo la GNU GPL). El usuario receptor de *A'* tendrá aseguradas todas las libertades del software libre.
- Pero si el software *A* hubiera sido publicado con una licencia permisiva, el desarrollador del software derivado *A'* podría distribuirlo con la licencia que quisiera. Los únicos requisitos que ha de cumplir son los indicados en las licencias permisivas, principalmente que se cite a los autores originales y la exclusión de garantía.

En definitiva, podemos concluir que la definición de software libre está pensada para maximizar las libertades del usuario receptor de la aplicación original *A*. Las libertades del receptor de la copia de *A'* dependerán de la licencia con la que se publicó la obra original *A*.

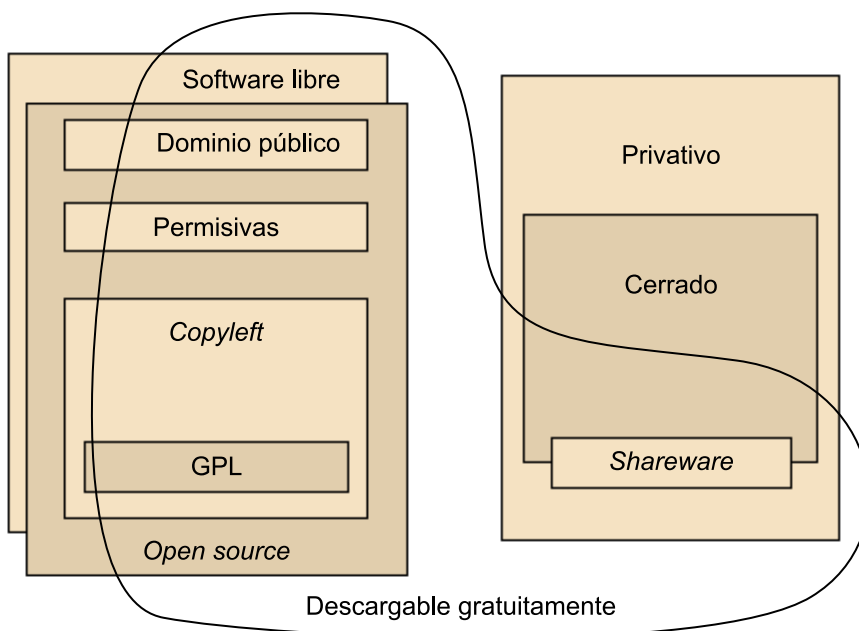
Hay mucha gente que argumenta que las licencias permisivas permiten *privatizar* el conocimiento, ya que existe la posibilidad de que se puedan crear obras derivadas con una licencia privativa. Como se puede ver en la figura, esto no es cierto del todo, ya que aunque *A'* tuviera una licencia privativa, el software *A* seguirá siendo libre.

## 6. Categorías de licencias

Existen multitud de licencias de software y muchos otros conceptos ligados a las licencias del software que hemos visto hasta ahora, o que utilizamos comúnmente, y que pueden resultar difíciles de encuadrar. La figura siguiente intenta clarificar con un diagrama de Venn la problemática de cómo relacionar algunas licencias y conceptos de software. Se han agrupado las aplicaciones según una serie de conceptos que darán los conjuntos; cada aplicación se puede considerar como un elemento y puede pertenecer a uno o varios conjuntos. Así, pueden existir:

- Conjuntos disjuntos, por ejemplo, una aplicación no puede ser software libre y software privativo a la vez.
- Conjuntos que están incluidos en otros, es decir, subconjuntos, como el caso de software publicado bajo la GPL, que es un subconjunto del software *copyleft*.
- Conjuntos parcialmente disjuntos (hay software libre que se puede descargar gratis, pero no todo).

Diagrama de Venn con algunos conceptos relacionados con las licencias y el software



Adaptado de uno publicado en la página web de la Free Software Foundation.

A continuación, se ofrecen algunas aclaraciones a la figura, sobre todo para aquellas relaciones que son más importantes:

- **El software libre y el software privativo son conjuntos disjuntos.** Esto debería ser elemental, ya que el software libre otorga las libertades presentadas en la definición de software libre y el privativo no lo hace.
- **El software *Open Source* y el software privativo son conjuntos disjuntos.** Al igual que en el caso anterior, para ser *Open Source*, el software ha de cumplir una serie de requisitos idénticos a los de software libre y las aplicaciones privativas no cumplen uno o más de esos requisitos.
- **Software libre**, tal y como lo entiende la Free Software Foundation, **y *Open Source***, liderado por la Open Source Initiative, **son filosóficamente idénticos.** Sin embargo, en la gráfica se muestran ligeramente disjuntos, porque hay unas pocas excepciones en las que algunas licencias son consideradas como software libre por la FSF, pero no *Open Source* por la OSI y viceversa. Pero se trata de un problema de interpretación de la licencia y de las distintas definiciones por parte de dos organizaciones independientes, no de diferencias ideológicas. Como se muestra en el diagrama de Venn, podríamos tener aplicaciones que fueran *Open Source* pero no software libre, y aplicaciones que sean software libre pero no *Open Source*. Son pocas, menos de una docena, pero existen.
- Todo el software en el dominio público, el distribuido con una licencia permisiva y el que tiene una licencia *copyleft* (incluido el software bajo la GPL), es tanto software libre como *Open Source*. Todos ellos otorgan las cuatro libertades del software libre y cumplen con la definición de *Open Source*.
- **El software en el dominio público, el software con licencia permisiva y el software con licencia *copyleft* constituyen conjuntos disjuntos.** Todos ellos son software libre, pero difieren, entre otras cosas, en las condiciones de redistribución (el software *copyleft* obliga a hacerlo con las mismas condiciones, para el de dominio público, pero no para el permisivo).
- La característica de **descargable gratuitamente se puede dar tanto en software libre como en software privativo.** El precio es una característica independiente de la mayoría de los conceptos que estamos manejando: podrá existir software libre que se pueda obtener gratuitamente y software libre por el que haya que pagar, al igual que pasa con el privativo. Recordemos que en la definición de software libre no se encuentra ninguna indicación con respecto al precio ni a la gratuidad del software. El concepto *shareware*, generalmente atribuido a software disponible en la Red que se puede utilizar con ciertas limitaciones (en algunos casos por un tiempo limitado, en otros con funcionalidad limitada, etc.), suele estar ligado a la descarga gratuita y, por lo tanto, se considera como subconjunto del anterior.

- Finalmente, en la figura se distingue entre software privativo y **software cerrado**. En esta ocasión, es complicado ser muy preciso, ya que no existe una definición universal de software cerrado. Si suponemos que el software cerrado es aquél que sólo se distribuye en binario, existe también software privativo que no es cerrado. Esto se da en casos en los que se proporciona el código fuente, pero no se cumple con una o más de las libertades que hace que un software sea considerado como libre. Un ejemplo de este tipo de software es aquél que permite el acceso al código, incluso su modificación, pero no la copia ni la redistribución de las modificaciones.



## 7. Licencias para otros tipos de contenidos

Las licencias de software están pensadas para el software e incluyen, en general, cláusulas que tienen sentido sólo para ese tipo de obras. Pero, como ya sabemos, la filosofía del software libre se ha expandido más allá del software. Es por ello que se han empezado a proponer licencias para otros tipos de contenidos. En esta apartado conoceremos las más importantes.

Uno de los primeros esfuerzos fue la **GFDL (GNU Free Documentation License, o Licencia de Documentación Libre de GNU)**, ideada por la Free Software Foundation para textos, en particular para la documentación anexa a un programa de software libre. Su planteamiento se asemeja al de la GNU GPL, pero considera algunos aspectos particulares de los textos. Al igual que en el software se distingue entre código fuente y código binario, en la GFDL se distingue entre copias "transparentes" y copias "opacas" de un documento:

- Las copias "transparentes" (algo así como las fuentes en el software) serían el fichero de texto editable.
- Por su parte, las copias "opacas" (el binario en el caso del software) se corresponden con el fichero no editable, por ejemplo un PDF.

Al igual que ocurriría con el software, la GFDL obliga a hacer llegar la copia "transparente" si así lo desea un usuario.

Sin embargo, se incluyeron una serie de restricciones adicionales que han hecho que la GFDL sea motivo de mucha polémica. Una de ellas es que se pueden modificar los agradecimientos, dedicatorias y el listado de versiones de un documento, pero sólo para añadir nuevas líneas. Otra, probablemente la más discutida, es la posibilidad de incluir **secciones invariantes**, que no permiten modificación. Esta consideración estaba pensada en principio para partes del texto no técnicas. Pero la inclusión de una sección invariante hace que la obra no pueda considerarse como libre siguiendo la definición de obra libre. Por último, una desventaja práctica del uso de la GFDL es que obliga a adjuntar el texto de la licencia completo al final del documento. Debido a que la licencia es voluminosa (unas diez páginas DIN A4) para textos pequeños y medianos resulta un lastre, sin tener en cuenta de que se trata de un dispendio inútil de papel.

## 7.1. Las licencias Creative Commons

**Creative Commons** es una familia de licencias propuestas por una organización sin ánimo de lucro del mismo nombre cuya finalidad es cambiar el marco legislativo con vistas a fomentar la creatividad.

La organización Creative Commons está íntimamente vinculada a la universidad estadounidense de Stanford, en la que es profesor su fundador, Lawrence Lessig.

**Creative Commons ofrece una gama de licencias** entre las que el autor puede elegir según su conveniencia. En contraposición con la postura del *copyright* estricto, donde los autores no ceden ningún o casi ningún derecho, la filosofía de las licencias Creative Commons se basa en que el autor se reserva algunos derechos. De esta manera se ha de entender el lema de Creative Commons, que es "**some rights reserved**" (algunos derechos reservados), en contraposición con el del *copyright* que es "**all rights reserved**" (todos los derechos reservados).

Las licencias Creative Commons más populares ofrecen la posibilidad de copiar, distribuir y comunicar públicamente la obra. Aparte de las licencias Creative Commons que se van a presentar en detalle, existen otro tipo de licencias Creative Commons de carácter específico:

- Por ejemplo, existe una licencia pensada para países en desarrollo. Las obras que se publiquen bajo esta licencia pueden ser copiadas, modificadas y distribuidas en países en desarrollo, pero tiene restricciones similares a las del *copyright* "estricto" en países industrializados. Esta licencia se retiró, ya que era utilizada por muy poca gente.
- Otra licencia Creative Commons es la de *copyright* de los fundadores ("Founders' Copyright" en su denominación en inglés). Esta licencia está basada en la legislación de *copyright* de los padres fundadores, aquéllos que promulgaron la constitución estadounidense en 1787. En ella, el *copyright* tenía una duración de 14 años, prorrogables al finalizar con otros 14 años, para pasar después al dominio público. Esta licencia se puede considerar como una "licencia protesta" cuya finalidad es recordar cómo se ha ido extendiendo el tiempo de validez del *copyright*, hasta llegar en la actualidad a los 90 años después de la muerte del autor.

Las licencias Creative Commons más utilizadas ofrecen un amplio abanico de posibilidades. La variedad de licencias se debe a que hay una serie de condiciones entre las que el autor puede elegir. Para facilitar la tarea a los autores,

la organización Creative Commons ha creado un formulario web sencillo que permite elegir fácilmente la licencia adecuada. Las características (y el logotipo) entre las que un autor puede elegir son las siguientes:

- **Reconocimiento (Attribution):** el reconocimiento de autoría es obligatorio. Durante los primeros años de Creative Commons, esta característica era opcional. Sin embargo, en la actualidad las licencias Creative Commons no dan opción a que un autor permita la copia o redistribución de su obra sin que se le reconozca la autoría. Hubo dos razones para que este cambio se llevara a cabo: por un lado, como hemos visto ya con anterioridad, en la legislación continental el reconocimiento es un derecho irrenunciable e ilimitado en el tiempo que forma parte de los derechos morales; por otro lado, más del 99% de los autores que rellenaban el formulario de Creative Commons para elegir licencia no cedía este derecho.



- **No Comercial:** la obra no puede copiarse, distribuirse o comunicarse públicamente con fines comerciales. Así, en el caso de que un autor publicara una novela con una licencia con cláusula no comercial, no estaría permitido vender copias de aquélla. Esta cláusula es muy querida por los que mantienen que la cultura ha de ser gratis, pero también es muy polémica. Esto es debido a que la definición de *fin comercial* es muy amplia y dada a múltiples interpretaciones. No es fácil dilucidar si un grupo de música que tocara una canción con una licencia con cláusula no comercial en un concierto en el que se pague entrada está cumpliendo o no con la licencia, aun cuando el grupo no cobrara por su actuación. O si se pueden ofrecer copias de un texto en un curso por el que se ha pedido matrícula. La casuística es enorme y los términos no son nada claros.



- **No Obras Derivadas:** se prohíbe la realización de trabajos derivados. De esta manera, una obra que contenga esta cláusula podrá ser copiada y distribuida, pero no está permitido alterarla.



- **Compartir Igual (ShareAlike):** los trabajos derivados sólo pueden publicarse y distribuirse bajo las mismas condiciones (en definitiva, bajo la misma licencia). Esta cláusula es la análoga en espíritu al concepto de *copyleft* que ya conocemos.



Las características descritas anteriormente **se pueden combinar**, siempre y cuando tengan sentido. Esto significa que se podría publicar una obra pidiendo reconocimiento, negando la posibilidad de hacer obras derivadas y prohibiendo fines comerciales. Esta licencia tiene como nombre *Creative Commons Reconocimiento-NoDerivadas-NoComercial*.

Se ha de notar que no todas las combinaciones son posibles, ya que la característica **Compartir Igual** (*ShareAlike*) es incompatible con la **No Derivadas**, porque para obligar a que una obra derivada sea publicada bajo las mismas condiciones que la original se ha de tener permiso para hacer obras derivadas. Las combinaciones posibles se presentan en la siguiente tabla, en la que también se muestra el **icono unificado** de la licencia que se puede encontrar con frecuencia en Internet.

Tipos de licencias Creative Commons

| Licencia                                  | Icono    | Icono unificado |
|-------------------------------------------|----------|-----------------|
| Reconocimiento                            |          |                 |
| Reconocimiento-NoDerivados                | <br>     |                 |
| Reconocimiento-NoComercial                | <br>     |                 |
| Reconocimiento-CompartirIgual             | <br>     |                 |
| Reconocimiento-NoComercial-CompartirIgual | <br><br> |                 |
| Reconocimiento-NoComercial-NoDerivados    | <br><br> |                 |

Ahora que conocemos las licencias Creative Commons, ¿cómo se usan? El mecanismo es muy sencillo. Lo primero que hay que hacer es elegir las condiciones con las que queremos compartir nuestra obra. Como se ha comentado con anterioridad, la organización Creative Commons ha creado un formulario

web que con unas pocas preguntas ayuda en la elección. Una vez realizada la elección, Creative Commons nos proporciona la licencia de tres maneras diferentes:

### 1) Un **texto para humanos** (disponible en varios idiomas).

Texto para humanos de la licencia Creative Commons Reconocimiento 3.0 España



The image shows the Creative Commons 'Reconocimiento 3.0 España' license interface. It features a green header with the CC logo and the text 'Reconocimiento 3.0 España'. Below this, a section titled 'Usted es libre de:' (You are free to:) lists two permissions: 'copiar, distribuir y comunicar públicamente la obra' (copy, distribute and communicate publicly the work) and 'hacer obras derivadas' (make derivative works), each with a corresponding icon. To the right is a circular seal that says 'Free Cultural Works APPROVED FOR'. A section titled 'Bajo las condiciones siguientes:' (Under the following conditions:) follows, with a person icon and the text 'Reconocimiento. Debe reconocer los créditos de la obra de la manera especificada por el autor o el licenciador (pero no de una manera que sugiera que tiene su apoyo o apoyan el uso que hace de su obra).' (Attribution. You must acknowledge the work in the manner specified by the author or licensor (but not in a way that suggests they endorse or support your use of the work)). Below this are three bullet points: 'Al reutilizar o distribuir la obra, tiene que dejar bien claro los términos de la licencia de esta obra.' (When reusing or distributing the work, you must clearly state the terms of this license.), 'Alguna de estas condiciones puede no aplicarse si se obtiene el permiso del titular de los derechos de autor' (One of these conditions may not apply if you obtain permission from the rights holder), and 'Nada en esta licencia menoscaba o restringe los derechos morales del autor.' (Nothing in this license diminishes or restricts the moral rights of the author.). At the bottom, a green box contains the text 'Los derechos derivados de usos legítimos u otras limitaciones reconocidas por ley no se ven afectados por lo anterior. Esto es un resumen legible por humanos del texto legal (la licencia completa) disponible en los idiomas siguientes: Catalan Castellano Euskera' (Rights derived from legitimate uses or other limitations recognized by law are not affected by the foregoing. This is a human-readable summary of the legal text (the full license) available in the following languages: Catalan Castellano Euskera). The word 'Advertencia' (Warning) is in the top right corner of this box.

El original está disponible en <http://creativecommons.org/licenses/by/3.0/es/>

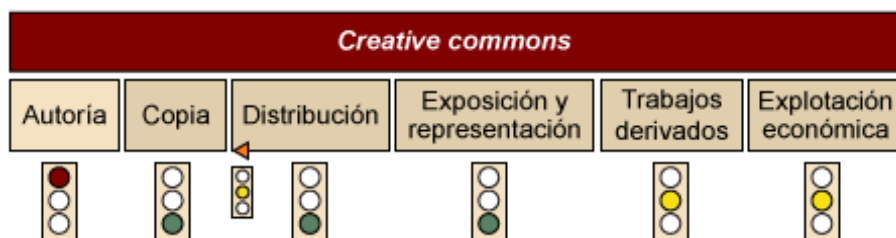
2) Un **texto para abogados**: se trata del texto jurídico completo y, como el lector se puede imaginar, de varias páginas y con numerosas cláusulas.

3) Un **texto para máquinas**: se trata de un modo de marcar la obra para que, por ejemplo, los buscadores puedan identificarla automáticamente como una obra bajo las condiciones de una licencia Creative Commons. Motores de búsqueda como Google o Yahoo! permiten, en su búsqueda avanzada, buscar por contenidos con licencias específicas gracias a estos metadatos que se incluyen en la obra.

Asimismo, se puede registrar la obra de manera voluntaria para que aparezca citada en el sitio de Creative Commons.

Para terminar de presentar las licencias Creative Commons, sólo nos queda debatir si todas ellas cumplen las condiciones especificadas en la definición de obra libre. Utilizando nuestro esquema de semáforo, obtendríamos la siguiente figura.

Diagrama de semáforo para las licencias Creative Commons



Como el autor puede decidir si permite crear obras derivadas y/o la explotación económica de la obra, éstas aparecen en ámbar. Al igual que con las licencias de software, tenemos la posibilidad de indicar que las obras derivadas se publiquen bajo las mismas condiciones. Por ello la flecha en el vértice inferior izquierdo de la cajita de *distribución* se ha coloreado también ámbar.

Un análisis de las diferentes características nos hará ver rápidamente que aquellas licencias Creative Commons que no permiten crear obras derivadas ni hacer uso comercial no pueden ser consideradas como libres. Por eliminación, las licencias Creative Commons que pueden ser consideradas libres según la definición de obras libres son la de Reconocimiento y la de Reconocimiento-CompartirIgual. Nótese que si hacemos la analogía con las licencias de software, la licencia Reconocimiento sería como una licencia permisiva, mientras que la licencia Reconocimiento-CompartirIgual se podría considerar equivalente a una licencia *copyleft*.

En este punto, es importante darse cuenta que aun siendo la característica CompartirIgual análoga en espíritu al concepto *copyleft*, no toda licencia Creative Commons que sea CompartirIgual puede ser considerada una obra libre. Esto ha dado lugar a muchas confusiones, ya que como vimos con anterioridad las obras *copyleft* son un subconjunto de las obras libres. Sin embargo, la utilización de Compartir Igual en combinación con No Comercial o No Derivadas no puede ser considerado como libre.

No quedaría completo este subapartado sobre licencias de obras libres, si no comentamos brevemente el sistema **ColorIuris**.

ColorIuris es un sistema internacional de gestión y cesión de derechos de autor que se garantiza mediante un contrato y no una licencia. Las diferentes modalidades de contratos ColorIuris tienen un esquema idéntico en esencia al de Creative Commons –aunque en este caso se identifican mediante un código de colores–, pero se plasman en contratos. La razón es que, según sus impulsores, las licencias no tienen la solidez de los contratos y podrían dar problemas si se llegara a juicio.

## Resumen

En este módulo se ha introducido el concepto de cultura libre, presentando las características que una obra ha de tener para ser considerada como tal. Dado que la libertad de las obras se plasma en las condiciones bajo las que se comparten aquéllas, se ha presentado la legislación que rige en estos casos, la propiedad intelectual. Es aquí donde hemos visto cómo se articulan jurídicamente los derechos de autor y el protagonismo que adquieren las licencias dentro de este marco.

Como ejemplo más avanzado de la cultura libre, gran parte de este módulo se ha dedicado a introducir el concepto de software libre, empezando por su historia y las motivaciones de sus primeros promotores. Asimismo, se han mostrado los tipos de licencias de software libre que existen y éstos se han puesto en contexto con otros conceptos de software que se manejan habitualmente. Al final, se han presentado licencias para otro tipos de contenidos que no sean libre, poniendo especial énfasis en la familia de licencias Creative Commons.





## Actividades

1. Investigad más a fondo la biografía de las siguientes personas, ligadas al software libre, y cread una ficha para cada uno en la que se incluya la siguiente información: nombre completo, nacionalidad, enlace a la página web personal, fotografía, proyectos en los que ha participado y sus principales contribuciones al movimiento del software libre.

- Richard Stallman
- Linus Torvalds
- Bruce Perens
- Eric S. Raymond
- John 'maddog' Hall
- Miguel de Icaza
- Matthias Ettrich
- Brian Behlendorf
- Andrew Tridgell
- Mark Shuttleworth
- Theo de Raadt

2. Realizad un diagrama de Venn como el del apartado 6 relacionando los siguientes conceptos: software libre, *Open Source*, dominio público, GNU GPL, GPLv2-compatible, Apache, *copyleft*, licencias permisivas, QPL, GNU LGPL, 3-clause BSD y 4-clause BSD.

3. Comentad brevemente por qué las siguientes afirmaciones (conocidas como los mitos del software libre) son falsas:

- a) Una de las particularidades del software libre es que es gratis.
- b) Lo bueno del software libre es que toda modificación revierte en la comunidad, ya que se han de publicar las mejoras que se haga a un programa.
- c) Una aplicación ha de ser software libre o software *Open Source*.
- d) Las licencias permisivas permiten que otros se apropien de software libre.
- e) El software *copyleft* no puede mezclarse con software que tenga otra licencia diferente al software original.
- f) Si un programa es software libre, puedo pedirle al autor original que me haga llegar las fuentes.
- g) Todo software que ofrezca el código fuente es software *Open Source*.
- h) El *copyleft* no respeta los derechos de autor.

4. Estudiad la licencia Christian Software Public License creada para la distribución Jesús.

- a) ¿Es la licencia Christian Software Public License una licencia de software libre? ¿Por qué?
- b) ¿En qué licencia se basa? ¿Qué ha sido modificado?
- c) ¿Es una licencia compatible con la GPLv2? ¿Por qué?

5. Estudiad las modificaciones introducidas en la tercera versión de la licencia GNU GPL.

6. A diferencia de las licencias de software, que generalmente sólo tienen una versión oficial en inglés, las licencias Creative Commons se traducen y adaptan a las legislaciones nacionales. Explicad las ventajas y desventajas de ambas posturas.

7. Las licencias Creative Commons ofrecen la posibilidad de elegir una licencia que no se podrían considerar como libres. Comentad las situaciones y razones en las que esas licencias pudieran ser más convenientes para los autores.

8. Realizad un diagrama de equivalencia entre las licencias Creative Commons y el sistema de gestión de derechos de autor Colorluris.

## Ejercicios de autoevaluación

1. Jacinto quiere conversar con su novia Amanda, que está de Erasmus, mediante un servicio de mensajería instantánea llamado Jabber. Amanda ve que hay un programa de software libre que se llama Pidgin que se puede utilizar para tal fin.

Teniendo esto en cuenta, contestad a las siguientes preguntas:

a) Amanda tiene instalado en la oficina el sistema operativo Windows. ¿Funcionará el Pidgin en Windows? La pregunta surge porque el software libre sólo funciona en Linux. ¿Es verdadero o falso? ¿Por qué?

b) Finalmente, Amanda consigue instalar otro programa de mensajería instantánea Jabber en su ordenador de la oficina. Sin embargo, cuando intenta conectarse, no puede. En realidad, tampoco ha podido usar nunca el Messenger en la oficina, pero sí navegar por la web y bajarse el correo electrónico. ¿A qué se puede deber esto?

2. Para un trabajo nuestro hemos citado un artículo que tiene una licencia de tipo Creative Commons ShareAlike-NonCommercial. Hemos decidido incluir varios párrafos enteros de ese artículo en nuestro trabajo y los hemos modificado levemente aportando nuevas ideas. Nuestro editor tiene miedo de que nos denuncien si lo publicamos con Copyright © y nos reservemos todos los derechos sobre el artículo. ¿Qué le diríais?

## Solucionario

### Ejercicios de autoevaluación

1.

a) Es falso. El software libre es software distribuido bajo ciertas condiciones (legales, no técnicas) que permiten al usuario tener la libertad de uso, copia, modificación y redistribución. Por lo tanto, puede haber software libre para Windows y para Linux.

b) Esto es debido a la existencia de un cortafuegos (*firewall*) que no permite (filtra) el tráfico de mensajería instantánea.

2. Al tener licencia (CC) de tipo ShareAlike-NonCommercial, tenemos la libertad de modificar el trabajo original para crear un trabajo derivado, siempre y cuando tenga la misma licencia. Por lo tanto, no podremos publicar el trabajo derivado con © y reservarnos todos los derechos. Pero sí podremos publicarlo con licencia (CC) ShareAlike-NonCommercial.

## Glosario

**BSD** *m* Acrónimo de Berkeley Software Distribution (Distribución de Software de Berkeley). Da nombre tanto a sistemas como a un tipo de licencias. Los sistemas BSD son sistemas libres basados en Unix, pero con un núcleo y un conjunto de herramientas ligeramente diferentes a las que encontramos en GNU/Linux. Existen varios tipos de licencias de BSD: FreeBSD, OpenBSD y NetBSD, cada uno con sus peculiaridades. Las licencias BSD son también conocidas como licencias minimalistas.

**código abierto** *m* Véase *Open Source*.

**copyleft** *m* Tipo de licencia que obliga a los que redistribuyen el software a hacerlo bajo las mismas condiciones con las que lo recibieron. De este modo, se transfiere a quien recibe cualquier trabajo derivado las mismas libertades de redistribución y modificación que le dieron al original. El lema oficial del *copyleft* es *all rights reversed* (todos los derechos revertidos). La licencia más conocida es la GNU GPL, aunque existe alguna más. Más información en la página web de GNU.

**Copyright** *m* Derechos relacionados con la explotación de la obra o prestación protegida. Son parte de los derechos patrimoniales en el ámbito continental europeo. El lema del *copyright* es *all rights reserved* (todos los derechos reservados).

**Creative Commons** *fpl* Familia de licencias propuestas por una organización sin ánimo de lucro del mismo nombre cuya finalidad es cambiar el marco legislativo con vistas a fomentar la creatividad.

**Debian** *m pl* Sistema operativo libre gestionado y promocionado íntegramente por cerca de mil voluntarios. En la actualidad, Debian utiliza el kernel de Linux para llevar a cabo su distribución (aunque se espera que existan distribuciones Debian con otros kernels, por ejemplo con HURD, en el futuro). Actualmente, está disponible para varias arquitecturas, entre ellas la i386. Más información en su página web.

**derechos compensatorios** *m pl* Derechos que compensan a los autores por la copia o reproducción de obras en el ámbito privado. Son parte de los derechos patrimoniales.

**derechos de explotación** *m pl* Véase *Copyright*.

**derechos morales** *m pl* Derechos del autor de una obra, mayoritariamente sin limitación temporal e irrevocables. Incluyen, entre otros, la condición de autor o el respeto a la integridad de la obra. Este tipo de derechos sólo existe en la acepción de propiedad intelectual continental.

**derechos patrimoniales** *m pl* Garantizan al autor la posibilidad de explotar su obra. Son derechos limitados en el tiempo y pueden ser cedidos. Están compuestos por los derechos de explotación y los derechos compensatorios.

**distribución** *f* La labor de las distribuciones es la integración de software independiente para su correcto funcionamiento en conjunto. Las distribuciones de GNU/Linux, como Debian, Red Hat, Slackware o SuSE, toman el código de los autores originales y lo empaquetan de manera que al usuario final le sea fácil instalar, actualizar, borrar y usar el software en su ordenador.

**dominio público** *m* Situación en la que se encuentra una obra para la que los derechos patrimoniales han caducado o se han cedido completamente.

**Eric Raymond** *m* Célebre hacker norteamericano conocido no sólo por el desarrollo de programas de software libre, sino también por ser autor de varios artículos que plasman la filosofía más pragmática del software libre, en especial "La Catedral y el Bazar". Fue uno de los propulsores de la Open Source Initiative y del término *Open Source*.

**Free Software Foundation (Fundación del Software Libre)** *f* Entidad sin ánimo de lucro fundada a mediados de los años ochenta por Richard Stallman para promocionar el uso del software libre. Su proyecto más conocido es el proyecto GNU, que no sólo ha aportado gran cantidad de software, sino que también se ha preocupado de difundir la filosofía del software libre. Más información en su página web.

**FSF** *f* Véase Free Software Foundation.

**(Proyecto) GNU** *m* Acrónimo recursivo de GNU Not Unix. Proyecto lanzado por la Free Software Foundation con el objetivo de conseguir un sistema operativo similar a Unix, pero

totalmente libre. Entre sus grandes logros está la articulación de la licencia GNU GPL. Más información en su página web.

**GNU/Linux** *m* Unión del kernel Linux y las herramientas proporcionadas por el proyecto GNU. Se trata de una solución de compromiso adoptada por la comunidad de software libre debido a que el fulgurante éxito de Linux ha propiciado que todo el sistema se llame como una de sus partes: el kernel.

**GNU GPL** *m* Acrónimo de GNU General Public License (Licencia Pública General de GNU). Se trata de la licencia copyleft más popular creada por la Free Software Foundation dentro del proyecto GNU. Existe una traducción de la Licencia Pública General de GNU al castellano.

**GPL** *m* Véase GNU GPL.

**hacker** *m y f* Programador habilidoso, experto en sistemas informáticos, gurú. Nótese la diferencia con cracker.

**licencia** *f* Especie de contrato con las condiciones de uso de la obra. Usando este mecanismo, el autor puede ceder permisos "automáticos" a otros.

**licencia minimalista** *f* Tipo de licencia de software libre. Sin embargo, al contrario que las licencias *copyleft* (también conocidas como robustas), el programa puede ser redistribuido bajo las condiciones que se quiera, siempre que se mantenga la nota de autoría. Esto significa que alguien puede redistribuir un programa licenciado con una licencia minimalista como software propietario, si así lo desea. A las licencias minimalistas también se las conoce como licencias BSD, ya que son estos sistemas los que las han hecho tan populares.

**licencia permisiva** *f* Véase licencia minimalista.

**Linus Torvalds** *m* autor principal del kernel Linux. Linus comenzó a trabajar como entretenimiento en el desarrollo de un kernel de tipo Unix cuando era estudiante de informática en una universidad finlandesa a principios de la década de los noventa. Con la popularización de Linux se trasladó al famoso Silicon Valley californiano, en el que ha estado trabajando primero para Transmeta, una compañía dedicada a la elaboración de chips, y posteriormente en Open Source Development Labs, dedicado íntegramente al desarrollo de Linux.

**Linux** *m* Kernel de sistema operativo. Su autor principal es Linus Torvalds, aunque en su elaboración han ayudado miles de desarrolladores. Más información en <http://www.kernel.org>

**obra derivada** *f* Obra que contiene elementos sustanciales de o se basa en gran parte en otra obra que ha sido publicada con anterioridad.

**OpenOffice.org** *m* Suite ofimática libre desarrollada principalmente por SUN. Consta de un procesador de textos (Writer), una hoja de cálculo (Calc), un programa para presentaciones (Impress) y una aplicación para imágenes (Draw). Se puede encontrar más información sobre OpenOffice.org en su página web.

**Open Source (en español: código abierto)** *m* Denominación alternativa del software libre enfocada más hacia los aspectos pragmáticos (modelo de desarrollo más dinámico, productivo, de mejor calidad, etc.). Uno de los creadores de este término y de la Open Source Initiative que lo avala fue Eric Raymond. Véase <http://www.opensource.org>.

**Open Source Initiative** *f* Organización sin ánimo de lucro dedicada a gestionar y promocionar el término *Open Source*. Su página web es <http://www.opensource.org>.

**OSI** *m* véase Open Source Initiative.

**propiedad intelectual** *f* Reconocimiento en favor de un autor u otros titulares de unos derechos sobre obras del intelecto humano. En el ámbito continental europeo, la propiedad intelectual se refiere exclusivamente a los derechos de autor, mientras que en el anglosajón también incluye las patentes, las marcas y los secretos industriales.

**Red Hat** *f* Distribución de GNU/Linux. Se trata de una distribución comercial –gestionada por la compañía Red Hat Linux– que utiliza paquetes en formato RPM.

**Red Hat Linux** *f* Empresa que realiza y comercializa Red Hat. Fue una de las primeras empresas de software libre en aparecer en medios de comunicación no dedicados específicamente a la tecnología, en especial tras su impresionante salida a bolsa en el índice NASDAQ a finales de la década de los noventa. Aunque sus acciones a día de hoy valen una centésima parte del valor de pico que alcanzó antes del desastre de las *puntocom*, desde hace

un par de años tiene un balance contable positivo. La dirección de su página web es: <http://www.redhat.com>.

**Richard Stallman** *m* Gurú, hacker y filósofo del movimiento del software libre. A mediados de los años ochenta fundó la Free Software Foundation y el proyecto GNU. Ha participado en la creación de multitud de programas de software libre (el editor de textos Emacs, el depurador GDB, etc.). En los últimos años, su actividad principal se limita a viajar por el mundo y dar conferencias sobre software libre y en contra de la patentabilidad del software.

**(licencia) robusta** *f* Véase copyleft.

**RPM** *m* Acrónimo de Red Hat Package Manager (Gestor de Paquetes de Red Hat). Sistema de paquetes creado y utilizado en Red Hat y distribuciones derivadas. Una aplicación software suele empaquetarse en uno o varios paquetes para facilitar su instalación y configuración.

**software libre** *m* Tipo de software con condiciones de uso y distribución que cumplen con las propiedades para ser considerado libre. En el artículo "Definición de Software Libre" incluido en esta colección se pueden encontrar las cuatro libertades que ha de tener un software para ser considerado libre.

**software propietario (o software privativo)** *m* Software con condiciones de uso y distribución que no cumplen con las condiciones para ser software libre. Véase software libre.

## Bibliografía

**Escolar, N.** (2001). *Por favor, ¡pirateen mis canciones!*

FreedomDefined.org. Definición de Obras Culturales Libres v1.0.

**Lessig, L.** (2004). *Free Culture: How big media uses technology and the law to lock down culture and control creativity*. Nueva York: Penguin Press.

**Perens, B.** (1999). "The Open Source Definition". En: Chris DiBona; Sam Ockman; Mark Stone (eds.). *Open Sources: Voices from the Open Source Revolution*. Cambridge, Massachusetts: O'Reilly and Associates.

**Raymond, E. S.** (1998). "The Cathedral and The Bazar". *First Monday* 3(3).

**Simon, H. A.** (1971). *Designing Organizations for an Information-Rich World*, pp. 37-72.

**Stallman, R.** (1999). "The GNU Operating System and the Free Software Movement". En: Chris DiBona; Sam Ockman; Mark Stone (eds.). *Open Sources: Voices from the Open Source Revolution*. Cambridge, Massachusetts: O'Reilly and Associates.

**Stallman, R. M.; Lessig, L.; Gay, J.** (2002). *Free Software, Free Society: Selected Essays of Richard M. Stallman*. Free Software Foundation.

