

# Software libre

Malcolm Bain  
Manuel Gallego Rodríguez  
Manuel Martínez Ribas  
Judit Rius Sanjuán

XP06/M2114/02159



# Aspectos legales y de explotación del software libre

## Parte I

### David Megías Jiménez

Coordinador

Ingeniero en Informática por la UAB.  
Magíster en Técnicas Avanzadas de Automatización de Procesos por la UAB.  
Doctor en Informática por la UAB.  
Profesor de los Estudios de Informática y Multimedia de la UOC.

### Jordi Mas

Coordinador

Ingeniero de software en la empresa de código abierto Ximian, donde trabaja en la implementación del proyecto libre Mono. Como voluntario, colabora en el desarrollo del procesador de textos Abiword y en la ingeniería de las versiones en catalán del proyecto Mozilla y Gnome. Es también coordinador general de Softcatalà. Como consultor ha trabajado para empresas como Menta, Telépolis, Vodafone, Lotus, eresMas, Amena y Terra España.

### Malcolm Bain

Autor

Abogado inglés. Máster en Relaciones Internacionales y Derecho Europeo de la UB. Especializado en el derecho de las tecnologías de la información y la comunicación. Participa en varios proyectos de investigación relacionados con ellas, bajo el auspicio de la Unión Europea.

### Manuel Gallego Rodríguez

Autor

Licenciado en Derecho por la Universitat Pompeu Fabra y Máster en Derecho de la Empresa. En la actualidad, trabaja como abogado en la firma MALET & GONZÁLEZ DE CARVAJAL de Barcelona, especializado en el área mercantil: Operaciones Societarias, Contratación mercantil y Nuevas Tecnologías.

### Manuel Martínez Ribas

Autor

Abogado en BAKER & MCKENZIE. Coordinador y responsable de proyectos europeos. Conferenciante habitual en Barcelona, Madrid (ESADE, IESE, Abad Oliva, Fomento, Recoletos, Cambra Barcelona, Institut Català de Technologies, Internet Global Conference, Univ. Pompeu Fabra, Univ. Politècnica Barcelona, Univ. Navarra), Bruselas, París (Univ. Paris), Fontainebleau (INSEAD), Darmstadt, Munich, Estocolmo, Amsterdam, Milán, Roma, Londres, Zurich. Autor de varias publicaciones de comercio electrónico.

### Judit Rius Sanjuán

Autora

Licenciada de Derecho y Master en Estudios Internacionales por la Universitat Pompeu Fabra. Beca de La Caixa para estudiar un Master en derecho, ciencia y tecnología en la Universidad de Stanford. Abogada en ejercicio, especializada en la asesoría a empresas del sector informático y farmacéutico, ha colaborado en proyectos de investigación subvencionados por la Unión Europea y es coautora de un estudio sobre Derecho y Comercio Electrónico en España.

Segunda edición: febrero 2007

© Fundació per a la Universitat Oberta de Catalunya

Av. Tibidabo, 39-43, 08035 Barcelona

Material realizado por Eureka Media, SL

© Autores: Malcom Bain, Manuel Gallego Rodríguez, Manuel Martínez Ribas y Judit Rius Sanjuán

Depósito legal: B-13.124-2007

Se garantiza permiso para copiar, distribuir y modificar este documento según los términos de la *GNU Free Documentation License*, Version 1.2 o cualquiera posterior publicada por la Free Software Foundation, sin secciones invariantes ni textos de cubierta delantera o trasera. Se dispone de una copia de la licencia en el apartado "GNU Free Documentation License" de este documento.

## Índice

<b>Agradecimientos</b>	9
<b>Presentación</b>	11
El programa del curso	17
<b>1. Introducción a los aspectos legales del software libre</b>	23
1.1. La libertad del software	24
1.2. Software abierto y software libre	25
1.2.1. El software libre	26
1.2.2. El software de código abierto	28
1.2.3. Objetivos diferentes	30
1.2.4. Las licencias de software libre	33
1.3. Consecuencias legales del modelo de software libre	36
1.3.1. La teoría de la génesis social del conocimiento	37
1.3.2. El argumento tecnológico	38
1.3.3. El coste	38
1.3.4. Una mayor difusión	39
1.4. Principales actores e intereses	39
1.4.1. Generadores y distribuidores de software libre	40
1.4.2. Usuarios de software libre	42
1.5. Terminología	44
<b>2. Propiedad intelectual: derechos de autor y copyright</b>	47
2.1. El concepto tradicional de los derechos de autor	48
2.1.1. Definición y objeto	48
2.1.2. Origen de los derechos de autor	49
2.1.3. ¿Derechos de autor o copyright?	50
2.2. Derechos de autor y software	51
2.2.1. Evolución histórica de la protección	51
2.2.2. Marco normativo	55
2.2.3. Sumario de conocimientos	62
2.3. Objeto y requisitos de protección	63

2.3.1. Objeto de protección .....	63
2.3.2. Requisitos para la protección .....	65
2.4. Autores y titulares de derechos .....	68
2.4.1. El autor .....	68
2.4.2. Titular original y titular derivado .....	75
2.4.3. Identificar al titular .....	76
2.5. Contenido de los derechos de autor .....	77
2.5.1. Derechos de carácter personal o derechos morales .....	78
2.5.2. Derechos de carácter económico o patrimonial .....	80
2.6. Límites a los derechos de autor .....	95
2.6.1. Límites temporales: la duración .....	95
2.6.2. Límites sustantivos: las excepciones .....	96
2.6.3. Sumario de conocimientos .....	105
2.7. Medidas de protección frente a la infracción de derechos .....	106
2.7.1. Medidas de protección .....	106
2.7.2. Medidas de reacción frente la infracción de derechos .....	116
2.7.3. Otras medidas de reacción .....	118
2.8. El sistema de derechos de autor anglosajón: el <i>copyright</i> .....	119
2.8.1. Marco legal .....	120
2.8.2. Características diferenciadoras principales ....	121
2.8.3. Doctrina del “uso justo” .....	123
2.9. Derechos de autor y software libre .....	124
2.9.1. <i>Copyleft</i> : el concepto .....	125
2.9.2. ¿Por qué no dominio público? Asegurar la libertad del usuario .....	127
2.9.3. Funcionamiento del <i>copyleft</i> .....	128
2.9.4. Software libre sin <i>copyleft</i> : licencias permisivas .....	129
2.9.5. <i>Copyleft</i> en obras que no son software .....	130
2.10. Derechos de autor y derecho <i>sui generis</i> de las bases de datos .....	131
2.10.1. Bases de datos que son obras originales ....	132
2.10.2. Bases de datos que no son obra original ....	132
2.11. Conclusión .....	134
<b>3. Propiedad industrial. Patentes de software y software libre. Marcas y secretos industriales .....</b>	<b>137</b>
3.1. Definición y objeto de las patentes .....	139
3.1.1. Naturaleza jurídica .....	141
3.1.2. Procedimiento .....	142

3.1.3. Definición de la patente .....	142
3.1.4. Clases de patente .....	144
3.1.5. Fundamentos de la patente .....	146
3.1.6. Requisitos de patentabilidad .....	149
3.1.7. Excepciones de patentabilidad .....	151
3.1.8. Procedimiento para la obtención de una patente .....	152
3.1.9. Homologación internacional de trámites en la solicitud de patentes .....	156
3.1.10. El documento de patente y las "reivindicaciones" .....	159
3.2. Titulares, duración y contenido de las patentes .....	164
3.2.1. Duración de la patente .....	165
3.2.2. Contenido de la patente .....	166
3.2.3. Transmisión y licencias de patentes .....	168
3.2.4. Obligación de explotar la patente: las licencias forzosas u obligatorias .....	171
3.2.5. Las patentes de combinación, selección y aplicación .....	173
3.2.6. Diferencias entre los derechos de autor o <i>copyright</i> y las patentes .....	175
3.3. Origen y marco normativo de la protección del software por las patentes .....	177
3.3.1. Razones para excluir la patentabilidad del software .....	180
3.3.2. La exclusión de la patentabilidad del software en sí mismo: normativa y jurisprudencia .....	183
3.3.3. La protección del software por derechos de autor o por patente .....	185
3.3.4. La patentabilidad del software .....	189
3.3.5. La Propuesta de Directiva del Parlamento Europeo y del Consejo .....	202
3.3.6. Ventajas y desventajas para proteger el software por las patentes .....	207
3.4. Patentes de software y software libre .....	210
3.4.1. Las patentes de software y el secreto del código fuente .....	211
3.4.2. Elementos patentables del software .....	215
3.4.3. Las patentes de software y el software libre: contradicciones y peligros .....	216
3.5. Marcas .....	229
3.5.1. Concepto, función y clases de marca .....	229
3.5.2. Duración y objeto .....	232
3.5.3. Adquisición de las marcas .....	234
3.5.4. Cesión y licencias de las marcas .....	238

3.6. Secretos industriales .....	239
3.7. Conclusiones .....	242
<b>4. Introducción al contrato de licencia de software .....</b>	<b>243</b>
4.1. Consideraciones generales .....	245
4.1.1. Licencias de software propietario y licencias de software libre .....	248
4.1.2. Soporte físico y formas de las licencias de software .....	249
4.1.3. Software estándar .....	254
4.1.4. Contratación “en masa” y condiciones generales .....	257
4.2. Función jurídica y económica .....	259
4.2.1. Consideraciones generales: la singularidad del software .....	261
4.2.2. Función jurídica y económica en las licencias de software propietario .....	262
4.2.3. Función jurídica y económica en las licencias de software libre .....	265
4.3. Naturaleza jurídica .....	266
4.4. Marco normativo .....	272
4.5. Licencias de software especiales .....	277
4.5.1. Licencia “llave en mano” .....	277
4.5.2. Licencia de uso especial para <i>upgrade</i> .....	279
4.5.3. Licencia de uso especial para <i>outsourcing</i> .....	279
4.6. Contratos accesorios a la licencia de software .....	280
4.6.1. Contrato de mantenimiento .....	281
4.6.2. Contrato de consultoría y formación .....	284
4.6.3. <i>Escrow</i> .....	285
4.7. Conclusiones .....	286
<b>5. Elementos de la licencia de software .....</b>	<b>289</b>
5.1. Elementos subjetivos o personales. Partes en las licencias de software .....	291
5.1.1. El proveedor-licenciante del software .....	291
5.1.2. El usuario-licenciatario. Especial referencia al usuario consumidor .....	306
5.1.3. Posición de los sujetos intervinientes en las licencias de software libre .....	314
5.2. Aceptación de las licencias de software. Licencias <i>shrink-wrap</i> y <i>click-wrap</i> .....	320
5.2.1. Licencias <i>shrink-wrap</i> y <i>click-wrap</i> ; discusión sobre su validez .....	322

5.2.2. Aceptación de las licencias de software libre: la GNU-GPL .....	328
5.3. Elementos objetivos en las licencias de software .....	331
5.3.1. Plazo .....	332
5.3.2. Precio .....	336
5.4. Conclusiones .....	339





## Agradecimientos

Los autores agradecen a la Fundación para la Universitat Oberta de Catalunya (<http://www.uoc.edu>) la financiación de la primera edición de esta obra, enmarcada en el Máster Internacional en Software Libre ofrecido por la citada institución.



## Presentación

### Ofensivas en el mundo de la tecnología

En enero del 2003 comienzan los rumores y en marzo se inicia el primer ataque: no se trata de la segunda guerra del Golfo, sino de la ofensiva de SCO contra IBM primero, y luego contra la comunidad de desarrolladores de software libre en relación con código incluido en el *kernel* del sistema operativo GNU/Linux 2.4, la última versión liberada por esas fechas.

Con esta ofensiva salen a la luz tensiones que ya existían entre el mundo del software propietario y el del software libre. Ya en el año 2000, Microsoft había hecho varias declaraciones críticas contra Linux, un sistema operativo que comenzaba a adquirir parte del mercado hasta entonces reservado para Microsoft (Windows) y varias otras empresas que distribuyen en forma propietaria diferentes versiones de Unix (IBM-AIX, Oracle-Solaris).

Pero actualmente estas tensiones no son sólo cuestiones técnicas de estabilidad, seguridad o precio, sino que surgen de temas más fundamentales sobre quién es el titular legal del código incluido en el programa, quién puede distribuirlo, y cómo y cuánto se habrá de pagar por un sistema informático hasta la fecha considerado gratuito y de disponibilidad ilimitada.

También en el año 2003, la Comisión Europea redacta y el Parlamento Europeo debate una propuesta de directiva para permitir –o no– que se otorguen patentes sobre los programas de ordenador innovadores, y determinar los requisitos formales para obtener esta forma de protección. Después de la movilización de una gran parte de la comunidad de desarrolladores de software (no solamente los involucrados en el software libre, sino también empresas de desarrollo de software comercial y otros actores del sector), en septiembre del 2003 el Parlamento Europeo aprueba una versión de la directiva que pone serias trabas a la patentabilidad de software por sí sólo.

Por encima del código, hace falta demostrar un “efecto tecnológico adicional” para obtener la protección absoluta de una patente.

Estas dos “historias de horror”, por llamarlas de alguna manera, indican que los aspectos legales del software en general y del software libre en particular se encuentran en el corazón de los debates actuales en el mundo de las nuevas tecnologías. Es más, estas discusiones son parte de un debate más amplio sobre la libertad, la cultura y la explotación de las obras inmateriales –entendido ese concepto como cualquier programa, película, música, texto o imagen en formato digital– en la nueva sociedad de la información. Este debate incorpora varias controversias similares, por ejemplo, la polémica de la carga y descarga de música o películas desde las redes P2P o la lucha para obtener un mayor control por parte de los titulares de los derechos sobre las obras (reunidos bajo la RIAA en Estados Unidos), de sus obras difundidas por las redes y el “ámbito” de ese control: espacial –en qué países–, temporal –por cuánto tiempo–, y funcional –qué se puede hacer con ellas.

El argumento se articula de esta manera: como las nuevas tecnologías permiten la copia y la difusión masiva y a bajo coste de obras inmateriales protegidas por ley (aquí nos hallamos ya en el territorio del derecho de la propiedad intelectual e industrial, hay que establecer nuevos mecanismos de control –tecnológicos y legales– para proteger a los autores y titulares de los derechos sobre estas obras.

### Movimientos de oposición

Frente a esta tendencia –marcada sobre todo por las grandes empresas de entretenimiento y las discográficas– vemos surgir un movimiento de protesta que reivindica la libertad de la cultura y de los elementos constitutivos: música, textos escritos, obras visuales y, sobre todo, el conocimiento en general. Estos movimientos de protesta no son originados por revolucionarios de pelo largo; muy al contrario, un equipo de profesores de derecho de las universidades de Harvard y Stanford han formado un centro de recogida y difusión de obras del dominio público (el Creative Commons), en el cual han establecido un nuevo sistema de expresión de *copyright* que permite a los autores variar cuidadosamente el grado de control sobre la copia, modificación y difusión de sus obras.

Otro componente de esta protesta es el movimiento de software libre, liderado en su día, y para algunos aún hoy, por la Fundación para el software libre, la Free Software Foundation (FSF). El software es parte intrínseca de la cultura, no solamente como conocimiento tecnológico, sino también como medio de acceso por excelencia a la cultura y al conocimiento. El software subyace a la red de redes, Internet, que interconecta los actores comerciales, los ciudadanos y las instituciones públicas. Pensemos, además, que el software se ejecuta en los ordenadores y los hace funcionar, que son elementos esenciales de la creación de conocimiento y las relaciones sociales, comerciales, científicas y educativas de hoy.

### El movimiento del software libre

Frente a la tendencia de apropiarse del conocimiento científico y tecnológico de las grandes empresas de informática –lideradas por Microsoft, pero también por otras empresas como Oracle, Apple, Sun y demás– se erige el movimiento de software libre (libre y abierto, por ahora no hacemos diferencia), un movimiento que quiere mantener y proteger la libertad en este sector fundamental para la sociedad moderna (o quizás sea mejor hablar del movimiento de la “creatividad libre”, por la aparición de iniciativas similares para todo tipo de obras). Se argumenta que la tendencia “propietaria” corre el riesgo de dejarnos en una sociedad del permiso –del “permiso por favor”– que hay que pedir a los titulares del conocimiento, un permiso que se otorgará bajo condiciones restrictivas y, a veces, draconianas. Otra vez estamos en el mundo de los aspectos legales, ya que estas restricciones y prohibiciones se expresan en las licencias de uso –para instalar un software, para ver una película, para escuchar una canción.

Justamente lo que quizás diferencia el sector del software de otros elementos de esta protesta general es que el movimiento de software libre se basa estricta y directamente en las leyes actuales para marcar su protesta, y se diferencia del software propietario por el uso de dichas leyes. El software libre se distingue del software propietario no por su calidad o su tecnología, aunque se haya argumentado que tiene ventajas en estas características, sino por la licencia de uso –por usar el término más común–, que no contiene

las medidas de protección y de control que se incluyen tradicionalmente en las licencias de software propietario.

Por lo tanto, el software libre está íntimamente vinculado con los aspectos legales relevantes para el software. Es importante destacar, por ejemplo, que:

- El software se protege por los derechos de autor, y los titulares y los usuarios tienen que acatar el régimen legal de la propiedad intelectual para, por un lado, permitir su uso, y por el otro, poder usar, distribuir o modificar cualquier obra (programa) protegida.
- Por el momento, en la Unión Europea, el software en sí mismo no es patentable, por las restricciones de sus licencias y su misma naturaleza. Sin embargo, como comentaremos, los intentos de instaurar las patentes de software son un peligro para el movimiento del software libre.
- El software se distribuye generalmente bajo licencia, y las licencias tienen varias formas y clausulados para determinar los usos, derechos, restricciones y prohibiciones.
- La mayoría de los usuarios más avanzados –y algunos menos– de aplicaciones informáticas requieren adaptar el software estándar a sus procesos. Por el contrario, el software libre lo permite bajo ciertas condiciones, sin pedir autorización suplementaria.

Todas estas cuestiones, y más, serán analizadas y esperamos que resueltas en este curso, con la estructura y la organización que explicaremos al final de esta presentación.

### Ejemplos actuales

Si observamos los diversos asuntos relacionados con el software libre que se han comentado en la prensa o que se comentarán en este mismo curso, veremos que en todos subyace una cuestión legal fundamental –y también unas dudas importantes, porque no todo está claro, y existen muchas cuestiones en discusión.

Por ejemplo, una noticia de actualidad en España el año pasado fue que la Junta de Extremadura ha lanzado LinEX, un programa amplio para promover la creación de una distribución basada en Debian GNU/Linux con el objetivo de instalarlo en todos los centros de educación pública de la región. Detrás de esta decisión hay por supuesto razones tecnológicas y económicas. Sin embargo, las consecuencias legales tienen igual importancia. El uso de software libre otorga una libertad de difusión del software a todos los ciudadanos y residentes sin necesidad de conseguir otra licencia. Asimismo, la Junta puede modificar fácilmente dichos sistemas para adaptarlos a las necesidades locales sin pedir permiso a los titulares originales –una estrategia de independencia legal considerable. Los centros educativos se benefician del programa por no ser dependiente de uno u otro proveedor (de programas o servicios de mantenimiento y actualización). Y pueden abrir el código fuente de las aplicaciones instaladas para fines educativos (clases de informática, etc.). Pero ¿bajo qué condiciones (derechos y limitaciones de derechos) operan la Junta y los centros en cuestión?

La existencia de paquetes de ofimática (el OpenOffice.org) como alternativa a los paquetes comerciales (Office de Microsoft, principalmente) también tiene sus aspectos legales. Dada la dominación de los paquetes comerciales en el mundo empresarial y doméstico, si un usuario de OpenOffice.org quiere compartir archivos de texto, de presentaciones o de cálculo con otras personas usando programas propietarios, es esencial mantener la interoperabilidad entre formatos de datos. Sin embargo, estos formatos son elementos de programa privados (propietarios) y protegidos por leyes de derecho de autor. En la ausencia de un estándar público para dicho formato de datos (como por ejemplo el SMTP para los correos electrónicos), ¿tiene un desarrollador interesado en adaptar OpenOffice.org el derecho de descompilar o estudiar el código original de las aplicaciones propietarias para poder exportar o importar archivos de origen propietario a la aplicación libre? ¿La obtención de una patente sobre un formato de exportación de archivo en XML (quizás la patente que consigue Microsoft en Estados Unidos en febrero del 2004 sobre los scripts en XML para Office 2003) implicará la necesidad de obtener una licencia para poder realizar esta interoperabilidad?

En el año 1998, Netscape abrió el código del Navigator para crear una versión libre que hoy se llama Mozilla. Esta decisión radical no

sólo provocó la renuncia del director de tecnología (que no veía las razones estratégicas y comerciales de esta decisión), sino que además provocó un debate ardiente sobre los términos y disposiciones de la nueva licencia o, mejor dicho, las licencias: Netscape tuvo que redactar una licencia para el Navigator inicial (la *Netscape Public Licence*, NPL) y una licencia (más abierta) para cualquier modificación futura (la *Mozilla Public Licence*). Esta historia tiene lecciones para cualquier desarrollador o empresa que deseara “liberar” un programa. ¿Cómo y por qué el titular de los derechos en un programa fue persuadido (por la comunidad de desarrollo libre) para “abrir” la licencia propuesta? ¿Qué obligaciones legales internas o ante terceros tenía que respetar Netscape para que se viera obligado a mantener durante cierto periodo de tiempo (4 años) la NPL para el programa original?

Jon Johanson, el creador del programa DeCSS (un descifrador de código encriptado de formato DVD) ha sufrido varios juicios en contra por publicar su código (y fuente libre) en Internet. ¿Qué relevancia tiene lo que hizo para provocar tantas reacciones de las empresas de multimedia? ¿A qué riesgos se enfrenta un programador que “cuelgue” en Internet un programa que pueda –o no– tener aplicaciones secundarias (o primarias) que puedan resultar en una demanda contra el autor o el difundidor? Podría permitir la elusión de medidas de protección tecnológicas (hoy protegidas por ley) de obras digitales –por ejemplo, por permitir acceso a los flujos de datos dentro del sistema operativo entre un lector de disco y el monitor (un riesgo potencial de cualquier SysOp libre).

En definitiva, la creación de software libre, su modificación y difusión, la reingeniería de las aplicaciones propietarias, su interoperabilidad, la patentabilidad del software, etc. son temas actuales y constantemente presentes en cualquier actividad que gire alrededor del tema central de este curso sobre aspectos legales y de explotación del software libre, y que levantan cuestiones legales importantes. No creemos necesario, en estas circunstancias, dar más argumentos. Sólo queremos indicar una vez más que los aspectos legales del software libre son numerosos y de extrema relevancia, pues condicionan todos los aspectos de un proceso de creación, distribución y uso de software libre. En conclusión, para cualquier persona interesada en crear, desarrollar, distribuir o usar software libre, será fundamental



tener un buen conocimiento de los aspectos legales del mismo, y ése es precisamente el propósito de este curso.

## El programa del curso

El curso se estructura en diez unidades y pretende cubrir los más importantes aspectos legales involucrados en la creación, distribución, modificación y uso del software libre.

A continuación, vamos a hacer una breve referencia a cada una de estas unidades para conocer la estructura y los conceptos que se van a introducir.

### La “filosofía” del software libre

La unidad 1 es una breve introducción a la “filosofía” del software libre. Esta unidad va a ser una introducción al núcleo del curso. Nuestro único objetivo será situar al lector en el contenido del curso.

### La propiedad intelectual e industrial

Las unidades 2 y 3 tratan del marco teórico legal de la propiedad intelectual e industrial. La propiedad industrial e intelectual es la denominación por la que se conoce a diferentes figuras jurídicas que intentan resolver el problema de cómo proteger los bienes inmateriales (como el software) donde el concepto de posesión de los bienes materiales es inexistente. La protección que otorgan pasa por el reconocimiento de unos derechos exclusivos al autor o inventor de un bien intangible que permiten al titular de los mismos limitar o delimitar el acceso o el disfrute de los demás sobre el objeto de su propiedad.



Para tener una idea más completa de lo que implica este concepto, se recomienda leer el artículo 2.VIII) del Convenio de 14 de julio de 1967 que estableció la Organización Mundial de la Propiedad Intelectual (OMPI o WIPO en inglés) y recorrer la página web [www.OMPI.org](http://www.OMPI.org).

**Texto complementario**

Confusión terminológica: en los sistemas anglosajones, principalmente Estados Unidos y el Reino Unido, la expresión “propiedad intelectual” engloba patentes y marcas junto a *copyright*/derechos de autor.

La mayoría de países cuentan con legislación nacional sobre propiedad intelectual e industrial, lo que propicia que tradicionalmente varíe de un país a otro, y sobre todo de una filosofía jurídica a otra. Por ejemplo, como estudiaremos en la Unidad 2, en la propiedad intelectual existen dos sistemas, los derechos de autor y el sistema del *copyright*. No obstante, en los últimos años ha existido una tendencia mundial a la armonización de las leyes nacionales a través de tratados internacionales que nos facilitará mucho la comprensión de los diferentes sistemas a los que hagamos referencia.

La propiedad intelectual e industrial se ordena tradicionalmente en dos grandes apartados:

- **Propiedad intelectual**, que se utiliza con un carácter sinónimo a derechos de autor y *copyright* (sobre obras literarias, artísticas y científicas); y
- **Propiedad industrial**: patentes, marcas...

Cada sistema de protección tiene sus propios requisitos y ámbitos; no obstante, es posible una superposición de protecciones que se rige generalmente por la regla básica de la independencia de las protecciones. Por ejemplo, como veremos, en algunos sistemas jurídicos es posible que un programa de ordenador sea protegido como obra literaria (derechos de autor) y como integrante de una patente.

A continuación vamos a realizar una breve introducción a los diferentes sistemas de protección existentes.

- 1) **Derecho de autor**: la expresión de un autor de obras científicas, artísticas o literarias. El derecho de autor es un derecho exclusivo que se otorga a las obras científicas, artísticas o literarias originales contenidas en cualquier medio tangible de expresión. Los derechos otorgados pueden ser divididos en el derecho “positivo” de reproducir, copiar y distribuir la copia en exclusiva y el derecho “negativo” de impedir a terceros que exploten la obra.

La protección de derechos de autor no se concede a ninguna idea, procedimiento, proceso, sistema, método de operación, concepto, principio o descubrimiento, independientemente de la forma como

esté descrito, explicado o materializado. Más bien, los derechos de autor imparten una protección que se limita a la expresión particular de un autor sobre una idea, proceso, concepto o algo similar, en un medio tangible.

En la unidad 2 vamos a estudiar con profundidad este mecanismo de protección especialmente relevante para el software, como veremos, pues es su principal y hasta ahora único recurso de protección en Europa, a pesar de las dificultades existentes en equiparar el régimen legal de derechos de autor, propio de obras literarias, a un programa de ordenador.

- 2) Las **patentes**: un contrato de la sociedad sobre invenciones técnicas. Se puede decir que una patente es un contrato entre la sociedad en conjunto y un inventor individual. Bajo los términos de este contrato social, se confieren al inventor derechos exclusivos: un derecho “positivo” de explotar en exclusiva su invención; y un derecho “negativo” de impedir, por un período de tiempo fijo, que otros fabriquen, usen y vendan la invención patentada, a cambio de que el inventor divulgue al público los detalles de la misma. De este modo, los sistemas de patentes alientan la divulgación de información al público y premian al inventor por sus esfuerzos, aunque también impiden a la sociedad acceder a la información y desarrollarla durante un periodo de tiempo determinado.

El sistema de protección por patentes y sus implicaciones para el software será estudiado con detenimiento en la unidad 3, aunque hay que tener en cuenta que es un sistema discutido. En Estados Unidos existen patentes de software, mientras que en Europa no existe la posibilidad legal de patentar software, pero en la práctica se otorgan y existen poderosos *lobbies* de presión que intentan modificar la legislación sobre el tema.

Existen otros sistemas de protección dentro de la propiedad intelectual e industrial, que aunque no vamos a estudiar en detalle en este curso, consideramos interesante que conozcáis:

- 3) **Marcas**: identificación del origen. La protección jurídica de las marcas es relevante en la comunidad del software libre, donde denominaciones como la Free Software Foundation o incluso Linux,

llevan aparejada una importante carga ideológica. Además, numerosas compañías, como Red Hat, basan su modelo de negocio en la marca como garantía de calidad, de ser pioneros en la venta de paquetes de software libre y en aportar soluciones de mantenimiento y consultoría especializada.

- 4) **Secretos industriales:** la ventaja competitiva. Aunque en el software libre los secretos industriales pierden gran parte de su relevancia (pues por su propia naturaleza, el software se distribuye con el código accesible) eso no implica que no se planteen conflictos en la comunidad de software libre relacionados con la figura de los secretos industriales. Un ejemplo es la reciente demanda SCO-IBM donde SCO basa parte de su acusación en que IBM ha vulnerado sus secretos industriales al distribuir software de su propiedad de manera abierta. De hecho, éste es un tema que debe tenerse presente; por su naturaleza abierta, cualquier distribución de software libre puede violar un secreto industrial (por supuesto, si no viene de una fuente libre).

### Las licencias de uso del software

Una vez el software se encuentra protegido por los derechos de propiedad intelectual, las licencias de uso de software constituyen el instrumento legal habitual a partir del cual quien ha desarrollado un software permite que otros lo utilicen, distribuyendo el mismo.

La licencia de software es un contrato por el que el titular de los derechos sobre el software permite su utilización a otra persona (el usuario), y determina las condiciones bajo las cuales dicho usuario puede utilizar el programa informático, así como las prohibiciones y los límites que debe respetar en tal uso. Cuando se instala un programa informático, o a veces incluso por el simple hecho de abrir el sobre que lo contiene, se dice que se aceptan las condiciones de su licencia de software, lo cual veremos que no siempre es una afirmación válida.

Las licencias de uso se han revelado, para los titulares de software propietario, como un buen instrumento para distribuir software, asegurar la máxima protección a sus derechos exclusivos y permitir obtener el máximo rendimiento económico –en su caso–, en un tra-

dicional modelo de comercialización del software basado en la venta de copias.

Sin embargo, las licencias de uso también han resultado ser apropiadas para la distribución de software libre. En este caso, y con un contenido distinto a las licencias de software propietario, permiten asegurar a los usuarios las libertades de uso, modificación y redistribución.

En las unidades 4, 5 y 6 vamos a estudiar con detenimiento las licencias de software, sus elementos y sus características principales.

En las unidades 7 y 8 vamos a estudiar las licencias de software libre propiamente dichas. Como lo demuestra el debate en el movimiento de software libre y el de software abierto, la licencia GPL no es la única licencia de software libre. Hay varios tipos de licencias, que varían principalmente según el grado de libertad otorgada relativa a la copia, la modificación y la distribución ulterior del software en cuestión. La unidad 7 establece criterios de comparación de las licencias en función de los derechos otorgados y retenidos, y estudia varias licencias paradigmáticas de cada tipo. Asimismo, comenta algunos movimientos casi (o pseudo) libres como el de *Microsoft Shared Source* y comenta algunos de los problemas legales fundamentales que pueden surgir en el momento de selección, aplicación e interpretación de una licencia.

En la unidad 8, como aspecto particular de la aplicación de software libre, comentaremos aspectos legales referentes al control de los productos de cifrado (criptología) y la protección de datos personales.

### Casos prácticos

Finalmente, en las unidades 9 y 10 propondremos una revisión de todos los conceptos desarrollados durante el curso y se introducirán nuevos conceptos para intentar que el lector alcance una comprensión global de la materia. El método utilizado es el de la lectura y el análisis de casos prácticos, reales o no, para aplicar los conceptos aprendidos en las unidades anteriores y conseguir una interrelación

de los mismos. En la introducción a dichas unidades explicamos el método de estudio propio para cada caso.

Los casos se centran en diferentes supuestos de explotación del software libre, desde los puntos de vista del creador, del distribuidor y del usuario de software: cuáles son los interrogantes y las dudas legales principales a las cuales se pueden enfrentar dichas personas.

## 1. Introducción a los aspectos legales del software libre

En esta unidad vamos a introducirnos en el núcleo de estudio de todo este curso, el software libre. Primero vamos a comprender qué significa jurídicamente el término *libre*, tantas veces repetido; a continuación nos adentraremos en el movimiento del software libre y del software abierto, y acabaremos con una iniciación al núcleo de la asignatura: las licencias de uso de software libre.

Al ser ésta una unidad introductoria, nuestro único objetivo es situar al lector en el contenido y la terminología del curso que empezamos, por lo que en la mayoría de temas vamos a remitir a las posteriores unidades para su desarrollo.

Los objetivos de esta unidad son:

1. Introducir al lector en los aspectos legales y de explotación del software libre, ilustrarle sobre la importancia de los aspectos legales para el desarrollo, la explotación y uso de software libre y situar los aspectos legales dentro del contexto tecnológico, comercial y político.
2. Presentar las iniciativas de software libre y software abierto, su origen histórico y los principales argumentos de desencuentro. Asimismo, clasificar las licencias de software libre que se van a estudiar, básicamente para introducir al lector en la terminología y los conceptos más relevantes.
3. Efectuar un repaso no exhaustivo de los principales argumentos que utiliza la comunidad del software libre para defender su utilización y sus implicaciones legales.
4. Presentar a los diferentes actores y los proyectos que se mencionarán en este curso para que el lector conozca los intereses en juego en la comunidad de software libre.
5. Unificar la terminología y los conceptos más importantes de este curso.

## 1.1. La libertad del software

La palabra *libre* (*free*) en inglés tiene dos sentidos: ‘libertad’ y ‘gratuidad’. Por ello, es necesario aclarar desde este momento que el uso de la palabra *free* relativo al software no implica que el titular o proveedor del software otorgue o distribuya el software de manera gratuita (aunque lo puede hacer).



El término *free* se refiere a que el software se distribuye bajo una licencia que permite a los usuarios aprovecharlo libremente.

Por lo que respecta a la contraprestación económica para la distribución de software libre, veremos que la mayoría de las licencias permiten al distribuidor imponer el precio que quiera.

### Ejemplo

Por ejemplo, las licencias de tipo BSD autorizan a privatizar el código y, por lo tanto, venderlo como un producto comercial. La *General Public License* (GPL) permite explícitamente cobrar por la distribución (cláusula 1). El precio está limitado únicamente por las reglas del mercado: como el usuario podrá publicar posteriormente el código fuente en Internet o realizar cualquier otra forma de distribución gratuita, cualquier tercero podrá obtener una copia sin pagar.

Aunque, como veremos en el apartado siguiente, el movimiento del software libre tiene diferentes ramificaciones, lo que unifica todos los modelos de desarrollo y tipos de licencias son las libertades que los desarrolladores de software libre desean que adquieran los usuarios.

Estas libertades usualmente se clasifican en cuatro:

- **Libertad 0:** la libertad de ejecutar y **usar** el software para cualquier propósito.



- **Libertad 1:** la libertad de **estudiar** el programa y **adaptarlo** a sus necesidades.
- **Libertad 2:** la libertad de **distribuir copias**.
- **Libertad 3:** la libertad de **modificar** el programa y **liberar** las modificaciones al público.

Para el ejercicio de estas libertades, en especial la 1 y la 3, el usuario necesita disponer del código fuente del programa. Por lo tanto, será imprescindible que las licencias de software libre contengan, efectivamente, el compromiso del titular-autor de proporcionar el código fuente a los usuarios; o, al menos, ponerlo a su disposición.



En definitiva, podemos definir las **licencias de software libre** como aquellas que, mediante la puesta a disposición del código fuente del programa de ordenador, permiten y aseguran a los usuarios el ejercicio de las libertades de ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software.

Esta definición nos proporciona una primera herramienta de análisis unificador de las licencias de software libre. No obstante, en 1998 hubo una cierta escisión conceptual en el movimiento de software libre que en realidad fue la concretización de una división que se venía desarrollando desde principios de los años noventa. Esta división se formalizó con la creación de la iniciativa de código fuente abierto (*Open Source Initiative*, OSI) que estableció la definición de software de código fuente abierto (*Open Source Definition*, OSD). A continuación, vamos a estudiar esta evolución histórico-filosófica dentro del movimiento del software libre.

## 1.2. Software abierto y software libre

Estudiaremos las dos categorías principales de software libre: el software libre en sentido estricto, y el software de código abierto. Segui-

remos una línea cronológica para explicar los dos movimientos, aunque, como veremos, muchos consideran el primero –el software libre de la Free Software Foundation– incluido dentro del concepto de software abierto de la OSI.

### 1.2.1. El software libre

Aunque encontramos un antecedente en la Universidad de Berkeley y en la licencia BSD que estudiaremos más adelante, para muchos los verdaderos iniciadores del movimiento del software libre (*free software*) son Richard Stallman y la **Free software Foundation (FSF)** a principios de 1984.

#### Nota

Ver:

- [www.gnu.org](http://www.gnu.org)
- [www.fsf.org](http://www.fsf.org)



Richard Stallman, en aquella época empleado en el AI Lab. del MIT, abandonó su trabajo para comenzar el proyecto GNU (acrónimo de *GNU No es Unix*) y fundó la Free software Foundation para conseguir fondos para el desarrollo y la protección del software libre. Richard Stallman sentó los fundamentos éticos del software libre con documentos como *The GNU Manifesto* y *Why Software Should Not Have Owners*.

Desde el principio del proyecto GNU, Richard Stallman estaba preocupado por las libertades que tendrían los usuarios del software creado. Estaba interesado en que no sólo los que recibieran los programas directamente del proyecto GNU, sino cualquiera que lo recibiera después de cualquier número de redistribuciones y (quizás) modificaciones, pudiera seguir disfrutando de los mismos derechos (modificación, redistribución, etc.).



El objetivo de la R. Stallman y la FSF es garantizar la libertad de uso del software (en el sentido de gozar de las cuatro libertades mencionadas) y mantener esa libertad en relación con obras derivadas de software originalmente libre.

En definitiva, para la FSF no es suficiente con cumplir las libertades básicas, lo que es importante desde la perspectiva de la FSF es que una aplicación es realmente libre si las distribuciones ulteriores –del mismo software o de derivados– permanecen libres.

Para conseguir este objetivo, R. Stallman escribió la **General Public License (GPL)** como base para asegurar la libertad de los usuarios del software libre. La GPL incluye términos de redistribución que no permiten a los redistribuidores añadir a su licencia cualquier restricción adicional (a las de la licencia original), ni al software original, ni a obras derivadas. Esta condición particular, que se conoce como *copyleft*, establece la imposibilidad legal de capturar el software libre, modificarlo y privatizarlo. Por lo tanto, el *pool*, o la cantidad de software con *copyleft* disponible, no puede más que aumentar a medida que los desarrolladores crean nuevas aplicaciones sobre la base del software con *copyleft*.

**Nota**

En la unidad 2 estudiaremos el concepto de *copyleft* con más detenimiento y en la unidad 7 la licencia GPL, no sólo porque es la licencia más usada en el mundo del software libre con un 75% del código libre publicado, ni porque es el precursor de muchas licencias libres actuales, sino también porque la filosofía de libertad de la FSF ha sido la base y uno de los elementos más destacados del movimiento libre.

La FSF defiende que el único software realmente libre es el que se distribuye mediante una licencia tipo GPL (con *copyleft*) que garantiza la libertad de los usuarios futuros, al obligar a los usuarios a continuar con la cadena de distribución abierta, negando la característica de verdadero software libre al software de código abierto. No obstante, no todo el software libre se distribuye con *copyleft* y, de hecho, el concepto de *copyleft* es lo que distingue el software libre de la FSF del resto de software libre distribuido bajo otras licencias conocido como software abierto.

A continuación, vamos a estudiar el software abierto que, para algunos, es sinónimo de libre.

### 1.2.2. El software de código abierto

Para algunos, una modalidad del software libre, para otros, el término general que engloba a todo el software libre, y para otros, una peligrosa desviación de los conceptos originales del software libre para conseguir su mayor comercialización, el software de código abierto (*open source software*) nace como una escisión del software libre de la FSF en 1998.

El proyecto “open source” surgió de una reunión estratégica mantenida el día 3 de febrero de 1998 en Palo Alto, California para reaccionar frente al plan de Netscape de liberar el código fuente de su navegador ‘Netscape Navigator’. Entre los presentes estaban: Eric Raymond, Bruce Perens (líder del grupo Debian), John “Maddog” Hall (de la organización Linux International) y Sam Ockman (grupo de usuarios de Linux de Silicon Valley).

Estas personas comprendieron que, con esta iniciativa que promulgaba una empresa comercial, se podía abandonar la actitud conflictiva que se había asociado con el software libre en el pasado e intentar ofrecer una perspectiva sobre el software libre más pragmática y orientada al mundo empresarial. Hasta entonces, la GPL “dominaba” la filosofía del movimiento libre.

Sin embargo, varias licencias libres muy comunes como la BSD o la Apache, que no tienen las obligaciones de *copyleft* robusto respaldadas por la FSF, permitían mayores posibilidades de distribución y de adaptación comercial. Se consideró oportuno, por lo tanto, establecer una iniciativa para amparar estas licencias abiertas pero sin *copyleft*. En realidad, muchas de sus ideas provienen de un escrito publicado por Eric Raymond en el año 1997, titulado *La catedral y el Bazar*.

Para reforzar esta iniciativa, sus defensores también establecieron el uso de la expresión *open source* (‘fuente’ o ‘código abierto’, o su acrónimo OSD) para, por un lado, diferenciarse del término *free software* (generalmente relacionado con el software de *copyleft* de la FSF) y por otra parte evitar el uso de la palabra *free*, para no tener connotaciones de gratuidad.

#### Nota

Ver *La catedral y el bazar* en:

<http://www.opensource.org/halloween/>

**Nota**

Es interesante notar que actualmente estos objetivos no se cumplen del todo, porque primero se sigue usando mayoritariamente la expresión “free software” (y en castellano, software “libre” en vez de abierto) para calificar mucho software que no es estrictamente *copyleft*, sino solamente abierto; y segundo, por que hay todavía mucha confusión relativa a la libertad y la gratuidad del software libre.



La OSD trata de reconciliar las libertades básicas del software libre con las necesidades comerciales de las empresas implicadas en la creación, distribución y uso de software libre.

De esta manera, el software libre mantiene las libertades fundamentales (copia, modificación, distribución, acceso a código fuente), pero no el concepto de *copyleft*, pues considera que la distribución posterior de obras modificadas debe permanecer flexible y no exigirse, como hace la FSF, la garantía de que la distribución posterior será libre.

Es importante tener en cuenta que la OSD no es una licencia, ni un modelo de licencia, sino **diez directrices** para la clasificación de licencias relativas a aplicaciones y productos de software en sus diversas formas (componentes, programas, distribuciones completas).



La OSD surge de las directrices Debian de software libre (*Debian Free Software Guidelines*, DFSG), adaptadas en 1998 básicamente por la eliminación de las referencias a Debian. En efecto, la definición de software abierto en las DFSG era suficientemente amplia como para incluir las licencias de tipo BSD, la GPL y su hermana LGPL, y otras como la del MIT/X y la Apache. Por lo tanto, sus requerimientos fueron adoptados por la OSI como las pautas generales que toda licencia abierta debía cumplir.

**Nota**

Para conocer con más detalle la OSD, podéis ver:

<http://www.opensource.org/docs/definition.php>

Además, la OSI ha elaborado una marca de certificación, la *OSI Certified*, que es una manera clara de indicar que una licencia cumple con la OSD. La marca sirve también para diferenciar el término general *open source*, que no tiene un uso suficientemente definido para garantizar esta conformidad.



En definitiva, la OSD fue diseñada para establecer una declaración abierta y comprensiva de los principios del movimiento de software abierto y una manera de clasificar y “certificar” la multitud de licencias libres que existen.

Se argumentó que al establecer estándares de esta manera, la definición permitiría a desarrolladores, usuarios, organizaciones comerciales y la Administración pública entender mejor el movimiento de software libre, respetar más sus principios, y, por qué no, encontrar nuevos modelos de negocio que aseguraran su futuro.

### 1.2.3. Objetivos diferentes

Una vez analizado el software libre y el software abierto, consideramos que no se puede argumentar que son dos movimientos opuestos entre sí. Lo que queda claro es que ambos persiguen objetivos diferentes (aunque no contrapuestos).

Por un lado está la Free Software Foundation y su defensa de la libertad de uso y distribución a cualquier precio para el beneficio de la comunidad, aun con el riesgo de perder el apoyo y las contribuciones de desarrolladores que quieren mantener un control económico sobre sus obras derivadas. Por el otro lado, tenemos la iniciativa código abierto y su doble intención de que se cree software libre certificado por la OSI y que además grandes compañías se sumen al proyecto.

De hecho, como veremos al estudiar la GPL, la FSF cumple con la OSD, por lo que la GPL se considera una licencia abierta también. Sin embargo, contiene otros requerimientos que la hacen muy particular, en concreto el más importante es la obligación de *copyleft*, que exige la persistencia de la libertad de uso y explotación sobre los programas derivados (la herencia de la libertad).

En realidad, salvo la cláusula de *copyleft* que instrumentaliza las diferencias conceptuales entre software *copyleft* y software abierto, las discrepancias no son legales, sino de postura. Según los defensores del software libre de la FSF, utilizar licencias de código abierto contribuye a que muchas compañías basen su negocio en software no-libre que trabaja con software libre. Realmente, son compañías cuyos productos añaden lo que ellas denominan “un valor agregado” al software libre y lo convierten en software no-libre al distribuirlo sin las licencias “adecuadas”. Los defensores del sistema GNU consideran que el software libre no distribuido con la licencia GPL no es verdaderamente libre, pues aunque el desarrollador lo distribuye de manera totalmente libre, no garantiza que los usuarios futuros gocen de esa libertad y permite que en el futuro sea distribuido como software propietario.

Por el contrario, los defensores de las licencias de código abierto argumentan que distribuir el software libre así garantiza el acceso a un mayor número de usuarios, la mejor comercialización del software y el deseo de no imponer ninguna restricción a la libertad de los redistribuidores y dejar que sean ellos los que decidan cómo distribuirán el software resultante. Una simplificación del pensamiento sería la frase: “Yo no pido nada de aquellos que usan mi código, salvo el reconocimiento”. La OSI se diferenció del movimiento de software con *copyleft* porque consideraba que la postura radical de la FSF asustaba a los empresarios. Bajo la perspectiva de los fundadores de la OSI, la incorporación de empresas grandes como IBM o Sun favorecerá y aún fortalecerá el movimiento de software libre frente al software propietario y acelerará su difusión y adopción.

Reconociendo este aspecto, la FSF ha ido cambiando gradualmente su postura tan agresiva contra la OSI.



Es interesante notar la evolución de la opinión de la FSF sobre el movimiento OSD a través de estos comentarios hechos por R. Stallman:

“La distinción es que la filosofía de código abierto se basa en hacer software confiable y poderoso. Se enfatizan los valores prácticos. No están equivocados, pero

#### Nota

Para ampliar este tema, recomendamos ver la postura de los defensores del software libre en [www.fsf.org](http://www.fsf.org) y la postura de los defensores de un software de código abierto en: [www.opensource.org](http://www.opensource.org).

eso no es todo. Yo creo que la libertad es más importante que los atributos de confiabilidad de un software. Si tengo que elegir entre un programa muy poderoso y mi libertad, me quedo con mi libertad.” (R. Stallman, *Linux Today*, 17/8/99)

“Algunos de los que favorecieron este término tenían como objetivo evitar la confusión de *free* con *gratis* –una meta válida. Otros, sin embargo, apuntaban a apartar el espíritu de principio que ha motivado el movimiento por el software libre y el proyecto GNU, y resultar así atractivos a los ejecutivos y usuarios comerciales, muchos de los cuales sostienen una ideología que pone las ganancias por encima de la libertad, de la comunidad y de los principios. Por lo tanto, la retórica de *open source* se centra en el potencial de realización de potente software de alta calidad, pero esquiva las ideas de libertad, comunidad y principio.”

“*Software libre* y *open source* describen la misma categoría de software, más o menos, pero dicen diferentes cosas acerca del software, y acerca de los valores. El proyecto GNU continúa utilizando el término *free software* [software libre] para expresar la idea de que la libertad, no solamente la tecnología, es lo importante.” (R. Stallman, 2000)

“Nuestra relación con el *open source* es la siguiente: estamos en desacuerdo en los principios básicos, pero coincidimos bastante en las recomendaciones prácticas. Entonces podemos hacer trabajos en conjunto en muchos proyectos. No los vemos como un enemigo, ya que nuestro enemigo es el software propietario. Reconocemos que han contribuido a nuestra comunidad.” (R. Stallman, 2000-2002)

**Nota**

Todos estos temas se desarrollarán con más profundidad en la unidad 7. No obstante la diferenciación indi-



cada, para simplificar conceptos vamos a utilizar el término *software libre*, a menos que indiquemos expresamente que nos referimos a una iniciativa concreta.

#### 1.2.4. Las licencias de software libre

La legislación sobre derechos de autor concede en exclusiva al titular del software el derecho a realizar o autorizar a terceros la reproducción (instalar y ejecutar), copia, modificación y distribución del software. Por tanto, para que un usuario pueda beneficiarse de las libertades del software, necesita el permiso explícito (la licencia) del titular de estos derechos: de lo contrario, se entendería que utiliza el software de forma ilegítima.



Una **licencia de uso de software** es un instrumento legal que autoriza a los usuarios del software a realizar ciertos actos que la ley normalmente reserva de manera exclusiva al titular de los derechos de autor o de patente. Asimismo, permite al autor reservarse los derechos que no se ceden e imponer y otorgar al usuario otras obligaciones y derechos no necesariamente vinculados con el derecho de autor (confidencialidad, etc.). Establece, por lo tanto, lo que el usuario puede y no puede hacer con el software.

Las licencias de uso constituyen también el instrumento legal habitual para distribuir el software libre. La diferencia entre las licencias de software libre y las de software propietario reside en los derechos y obligaciones especificados en la licencia. Aquéllos otorgados por las licencias de software libre ("licencias libres") suelen ser directamente opuestos a los otorgados y reservados por una licencia de software propietaria ("licencia propietaria"), sobre todo en cuanto al uso, la distribución y la modificación del software.

Mediante las licencias de software libre, el titular del software no pretende proteger sus derechos exclusivos de explotación. Al contrario, mediante la licencia de software libre el titular permite expresamente

#### Nota

Estudiaremos la legislación sobre derechos de autor en la unidad 2.

a los usuarios usar, modificar, así como redistribuir el software, con o sin modificaciones. En este sentido, para que una licencia de software sea “libre”, debe garantizar, al menos, las cuatro libertades básicas estudiadas al inicio de esta unidad.

Por otra parte, las condiciones y restricciones que se imponen a los usuarios en las licencias de software libre son mucho menores que las recogidas en las licencias de software propietario. Además, estas “limitaciones” a las libertades no pretenden proteger derechos exclusivos del proveedor, sino fundamentalmente pretenden:

- Proteger la reputación del autor del software.
- Asegurar las libertades de uso, modificación y distribución a los usuarios en todo momento; en particular, evitar e impedir cualquier intento de “apropiación” del software libre.

Una vez diferenciadas las licencias de software propietario de las licencias de software libre, es importante destacar que **no todas las licencias de software libre son iguales**. El abanico de posibilidades va desde unas obligaciones mínimas (en las licencias de tipo BSD) que obligan únicamente a mantener el aviso de autoría y la negación de garantías y de responsabilidad (*disclaimer*), hasta el “máximo” (en cierto sentido) de la cláusula *copyleft* de la GPL, que obliga al usuario a distribuir cualquier modificación y obra derivada bajo la misma licencia GPL. Además, la gama va más allá de lo realmente libre hasta licencias que no son “libres” en el sentido estricto de asegurar las cuatro libertades, pero que intentan ajustarse al modelo de desarrollo libre, como la *Sun Community License*.

#### Reflexión

A causa de la obligación de distribuir de manera libre cualquier obra modificada o derivada, se ha dicho que “la GPL es menos libre” que otras licencias abiertas. La FSF rechaza esta calificación, pues argumenta que, por el contrario, la licencia GPL es más libre porque garantiza más libertad para el usuario final. Hay que matizar:

- La BSD, por ejemplo, otorga más libertad a los desarrolladores porque éstos pueden incorporar y dis-

tribuir implementaciones de “código BSD” bajo licencias ambas libres y propietarias.

- La GPL transmite más libertad a los usuarios finales, porque éstos siempre recibirán aplicaciones con código fuente abierto y una licencia libre.

A continuación vamos a enumerar brevemente los principales tipos de licencias libres, aunque no las vamos a estudiar con detenimiento, pues éste es el propósito de la unidad 7. A los fines de estudio, hemos clasificado las licencias abiertas y libres en cuatro categorías.

Estas cuatro categorías son las licencias libres con *copyleft* robusto, las licencias libres sin *copyleft* robusto, las licencias libres sin *copyleft* y las licencias “seudo-libres”.

#### a) Las licencias libres con *copyleft* robusto

La licencia creada por la Free Software Foundation, la *General Public License* (GPL), es el principal exponente de este tipo de licencias que con la utilización de la figura del *copyleft* pretenden asegurar las cuatro libertades fundamentales del software libre, y también que las modificaciones de éste y el nuevo software derivado sean distribuidos en las mismas condiciones.

#### b) Las licencias libres sin *copyleft* robusto

La licencia *Berkeley Software Distribution* (BSD) es quizás la versión más simple de todas las licencias libres y es también la primera licencia libre creada. Surge de las distribuciones de versiones de Unix de la Universidad de California, Berkeley, en los años 1970 y 1980. La filosofía que late tras esta licencia se encuentra en que el código es fruto de las investigaciones y los trabajos universitarios financiados por el gobierno de los Estados Unidos (y los impuestos del pueblo americano), por lo tanto, debe ser de acceso libre, protegiendo lo que llamaríamos aquí los “derechos morales” de los autores por la simple obligación de mantener los avisos de autoría (*copyright notice*).

La BSD ha sido modelo de muchas licencias parecidas, entre las cuales citamos la licencia Apache, las licencias de la familia X (X, XFree86, XOpen, X11) y las licencias de la familia BSD (OpenBSD, Free BSD, NetBSD).

Este tipo de licencia es compatible con la GPL, por no tener restricciones mayores que las permitidas por la FSF, aunque no contiene obligaciones de *copyleft* y permite la privatización de obras derivadas.

#### c) Las licencias libres sin *copyleft*

Estas licencias son incompatibles con la GPL, en el sentido de que no se puede integrar material de estos programas en un programa o su obra derivada bajo la GPL porque las licencias sobre estos materiales incluyen obligaciones que son más restrictivas que la GPL. En muchos casos, derivan de la obligación de publicidad que estaba incorporada en la primera versión de la BSD, pero también pueden surgir de obligaciones sobre patentes, nombramiento, indemnizaciones u otros temas.

Los principales ejemplos son la *Netscape Public Licence* (NPL) y la *Mozilla Public Licence* (MPL). La licencia MPL se usa para varios programas de mozilla.org entre las cuales se encuentran el navegador Mozilla (mozilla.org).

#### d) Las licencias “seudo-libres”

Aunque en este curso nos centraremos en las licencias de software libre, en la unidad 7 también se hará un breve análisis de las licencias creadas por empresas que intentan beneficiarse del modelo de desarrollo libre. Los ejemplos más importantes son la *Sun Community Source License* (SCSL) y la *Microsoft Shared Source Initiative* (MSSI).

### 1.3. Consecuencias legales del modelo de software libre

El movimiento del software libre utiliza varios argumentos para defender su utilización, pero algunas veces olvida que todos estos ar-

gumentos tienen importantes consecuencias legales. Para conocer estas consecuencias y a la vez entender un poco mejor la iniciativa del software libre, consideramos adecuado comentar brevemente estos argumentos.

### 1.3.1. La teoría de la génesis social del conocimiento

El primer argumento utilizado por los defensores del software libre es filosófico-ideológico. El principio básico en la mayoría de las argumentaciones de los defensores del movimiento del software libre es que el conocimiento como tal no pertenece a nadie, pues cualquier conocimiento proviene de unos conocimientos anteriores y es una copia en mayor o menor medida de otras ideas.

En otras palabras, nadie tiene ideas que no hayan estado directa o indirectamente influenciadas por las relaciones sociales que mantiene en las comunidades de las que forma parte, y si la génesis es social, el uso debe permanecer social a su vez. La función principal de la generación de conocimiento es mejorar la sociedad y, por tanto, debe llegar al máximo número de personas.

Una vez explicado lo anterior y si consideramos el software como conocimiento, continuar con el hilo de la argumentación de organizaciones como la FSF parece sencillo. No obstante, al estar dentro de un curso de aspectos legales debemos plantearnos qué consecuencias tiene esta filosofía para un análisis jurídico del software libre.



La consecuencia más directa de esta filosofía de la génesis social del conocimiento es que la limitación de la copia, el uso o la reproducción del software no tiene sentido, pues dificulta la generación de nuevo conocimiento y no permite que cumpla su principal objetivo: revertir en la comunidad.

Este argumento lleva a la crisis del sistema tradicional de derechos de autor y modifica las estructuras típicas de las licencias de uso.

Esta filosofía del software libre puede parecer idealista en el contexto actual, dominado por los sistemas de derechos de autor y patentes. Sin embargo, para sus defensores, estar o no de acuerdo con ella no supone un impedimento para percibir claramente sus beneficios prácticos, tanto tecnológicos como comerciales o estratégicos. A continuación vamos a comentar estos beneficios.

### 1.3.2. El argumento tecnológico

Uno de los argumentos básicos del movimiento del software libre es la defensa de la capacidad que debe tener cualquier usuario de conocer, modificar y adaptar el programa a sus necesidades en cada momento.



Este argumento implica que el software libre se distribuye de forma abierta (o sea, con el código fuente accesible) y que las licencias prevén expresamente la capacidad de modificar el software sin necesidad de pedir autorización al titular original.

Aunque, como veremos, el software propietario permite a veces su modificación por el usuario legítimo (por el juego de las excepciones de los derechos de autor y la permisividad de algunas licencias propietarias), lo cierto es que dicha capacidad de adaptación es muchas veces limitada, la solicitud de autorización al titular es a menudo obligatoria y el acceso al código fuente es casi siempre imposible.

### 1.3.3. El coste

Hemos visto que *software libre* no significa ‘software gratuito’, aunque a menudo el argumento de su bajo coste es utilizado por muchos de sus usuarios como determinante en la elección. Al ser consustancial al software libre, su distribución con el código fuente y la autorización de copias y distribuciones posteriores ilimitadas, el concepto de “pago por licencia” del software tradicional pierde su sentido.

La consecuencia jurídica directamente derivada es que las licencias de software libre no establecen normalmente pago de *royalties* por los derechos de uso sobre el software, y el modelo de negocio de los desarrolladores de software libre se basa en la prestación de servicios de mantenimiento, personalización del software o consultoría.

#### 1.3.4. Una mayor difusión

Existe una cuarta razón, que podríamos considerar la razón “pragmática”. Los defensores del software libre sostienen que distribuir una obra mediante una licencia libre (en el sentido de asegurar al usuario las cuatro libertades básicas) es la mejor manera de obtener beneficios de la publicación de contenidos cuando no se es un autor consagrado que pueda obtener beneficios de un *copyright* cerrado.

El argumento es que los únicos que realmente ganan con el sistema de derechos de autor o *copyright* son los autores famosos y los intermediarios, o sea, las editoriales. El resto de autores viven del prestigio que les dan sus obras, que les permiten prestar “servicios adicionales”, como proporcionar mantenimiento o consultaría o dar conferencias, cursos, charlas, escribir en periódicos, etc. Siguiendo este argumento, para la mayoría de autores lo realmente importante para aumentar su rendimiento económico es que su obra llegue al máximo número de personas. Así, su conclusión es que, para un autor desconocido, el sistema de derechos de autor supone una barrera para llegar al público y obtener beneficios de sus creaciones.



Un artículo muy clarificador sobre el tema escrito por el colectivo Wu Ming, puede leerse aquí:

[http://www.wumingfoundation.com/italiano/Giap/giap2\\_IV.html#copyright1](http://www.wumingfoundation.com/italiano/Giap/giap2_IV.html#copyright1).

### 1.4. Principales actores e intereses

Una vez entendido el concepto de software libre a efectos jurídicos y las importantísimas consecuencias legales de optar por su utiliza-

**Nota**

En la unidad 2 estudiaremos por qué la solución de los conflictos de autoría no es obvia.

**Nota**

En la unidad 2 estudiaremos las importantísimas consecuencias legales de este modelo de desarrollo de software en la teoría de derechos de autor.

ción, en este apartado vamos a introducir a los principales actores e intereses en juego. Creemos importante mostrar al lector la relevancia que el software libre tiene actualmente y la multitud de proyectos e iniciativas privadas y públicas que se trabajan para su consolidación como alternativa al software propietario.

### 1.4.1. Generadores y distribuidores de software libre

Como hemos visto, una diferencia fundamental entre el software libre y el propietario es el modelo de desarrollo. Para las empresas de software propietario, el modelo generador de software más habitual se basa en el desarrollo interno por unos programadores asalariados. La existencia de conflictos de autoría son obvios, pero su solución no lo es.

En cambio, en el modelo del código abierto o libre, el software normalmente se genera de forma cooperativa por programadores, muchas veces voluntarios, que trabajan de forma coordinada desde Internet. Además, muchas veces los usuarios, al poder conocer el código fuente y poder modificarlo, en ocasiones pasan a ser autores de una versión mejorada o adaptada a sus necesidades. Este modelo de desarrollo de software tiene importantísimas consecuencias legales, especialmente en la teoría de derechos de autor.

Numerosos proyectos en el mundo del software generan actualmente software libre. Vamos a intentar clasificarlos de manera, por supuesto, no exhaustiva.

#### a) Proyectos vinculados a empresas

Como hemos visto, el software libre no debe confundirse con software gratuito, pues el modelo de software libre no impone ninguna restricción sobre la cantidad que podemos cobrar por un programa libre. Sin embargo, el hecho de disponer del código fuente y poderlo distribuir hace que la venta de software libre siempre sea a un precio bajo, que normalmente no suele ser más que el coste del CD-ROM, el manual y un pequeño margen para la distribución.

De esta manera, el software libre lleva aparejado un nuevo concepto donde el código pasa a ser un bien común y además necesario para



construir negocio a su alrededor. Normalmente, las fuentes de ingresos se encuentran en el valor añadido que se dan al producto, especialmente el soporte técnico, la formación y la personalización del software.

Quizás la empresa más representativa dentro del movimiento del software libre es Red Hat. Fundada en 1994 por Bob Young y Marc Swing, basa su modelo de negocio en la prestación de servicios y la consultoría profesional y sobre todo en el prestigio de su marca. Actualmente, su modelo de negocio parece sólido, se encuentra presente en todo el mundo con más de 500 empleados y multitud de delegaciones.

Otros ejemplos de proyectos generadores de software libre vinculados a compañías privadas son la empresa Sun Microsystems que mantiene OpenOffice.org; la fundación Mozilla, hasta hace poco directamente dependiente de America Online, que mantiene el proyecto Mozilla.org; o Ximian que mantiene el sistema de correo electrónico Evolution. También hay proyectos, como Apache, que aunque están representados por fundaciones, reciben ayuda de empresas como IBM.

#### **b) Proyectos desarrollados en universidades**

Los inicios del movimiento del software libre se encuentran en proyectos de desarrollo de software en universidades norteamericanas, que a su vez reciben financiación de empresas privadas o de gobiernos. Uno de los ejemplos más representativos es la familia de sistemas operativos BSD, en concreto NetBSD y FreeBSD desarrollados en la Universidad de Berkeley, en California.

#### **c) Proyectos liderados por grupos de voluntarios**

Como hemos comentado, Richard Stallman y La Free Software Foundation (FSF) fueron los iniciadores de todo el movimiento de software libre. Su modelo de desarrollo de software libre se basa en las contribuciones de cientos de voluntarios de todo el mundo.

Aunque actualmente la FSF continúa siendo una organización de enorme influencia en el sector agrupada en torno al proyecto GNU,

hay otros ejemplos de proyectos de desarrollo de software libre basado en el trabajo de voluntarios. Algunos ejemplos son la OSI, Debian o Abiword.

Según argumentan la mayoría de estas organizaciones, la motivación principal de sus colaboradores es pensar en el software libre como un bien común, lo que nos lleva a relacionar este tema con la teoría del conocimiento comentada en el apartado 3 y sus consecuencias para la autoría y los derechos morales.

### 1.4.2. Usuarios de software libre

Hemos estudiado los actores que intervienen en la génesis o creación del software libre; ahora vamos a estudiar quiénes son sus principales usuarios.

Tradicionalmente, el principal usuario de software libre era la propia comunidad de desarrolladores y distribuidores; no obstante, en los últimos años el software libre se ha empezado a dar a conocer a la sociedad en general y su uso se ha multiplicado. Numerosas iniciativas intentan implementar software libre, pero quizás los ejemplos más exitosos se hallan en la Administración pública y en centros de enseñanza.

Las motivaciones de estas iniciativas para elegir utilizar software libre son variadas y sus consecuencias legales muy importantes, como hemos visto: desde el ahorro de costes que supone no tener que pagar el elevado precio de las licencias y poder realizar tantas copias como se desea, pasando por la posibilidad de evitar la dependencia de grandes multinacionales en la búsqueda de servicios de mantenimiento, hasta el aseguramiento de la libertad de modificación, etc.

Aparte de las administraciones públicas y los centros de enseñanza, el software libre también encuentra defensores de su uso en las organizaciones de cooperación al desarrollo que luchan por reducir lo que se ha denominado “división digital”, o sea, las diferencias de acceso a las nuevas tecnologías entre las regiones más y menos desarrolladas del planeta. En la tercera edición del *Informe de comercio electrónico y desarrollo de la Conferencia de las Naciones Unidas para el Comercio y el Desarrollo* (UNCTAD) se destacó la importancia

#### Nota

Ver:

<http://www.unctad.org/Templates/webflyer.asp?docid=4255&intItemID=1634&lang=1>

que tiene el acceso a software libre para el desarrollo de las nuevas tecnologías e Internet en los países en desarrollo.

El software propietario ha demostrado ser poco sensible con las economías menos favorecidas, para las cuales el precio de licencia se hace inalcanzable; con las lenguas minoritarias, a las que han arrinconado bajo el argumento de que no “son rentables”; o con los propios usuarios, que muchas veces han sufrido precios desorbitados o han tenido que pagar por parches o versiones posteriores que solucionan problemas del mismo software que habían comprado.

#### Ejemplo

Un ejemplo es el caso de Brasil. El Gobierno brasileño anunció a comienzos/mediados del 2003 que utilizaría software libre en prácticamente todas las áreas de su Gobierno, convirtiéndose así en el primer ejecutivo a nivel mundial en hacer un despliegue masivo de este sistema en la Administración pública.

La industria informática en Brasil mueve más de 3.000 millones de dólares al año, de los cuales prácticamente 1.000 se van a Estados Unidos en concepto de licencias de software. El Gobierno de Lula está apostando por la creación de un tejido local de empresas que den soluciones y servicios en software libre, fomentando así el desarrollo local y desvinculando la inversión pública de las grandes multinacionales.

Una parte importante de la responsabilidad del avance del software libre en Brasil recae en la iniciativa *software libre* la cual impulsa desde hace varios años el uso del software libre en la Administración pública y ha venido organizando el Forum Internacional de Software Libre.



Para saber más sobre proyectos e iniciativas de software libre en Brasil, podéis ver:

<http://www.softwarelivre.org/>

**Nota**

En la guía de aprendizaje disponéis de un glosario completo con la terminología más utilizada en este curso, os remitimos a él para los demás conceptos.

## 1.5. Terminología

Cualquier aproximación al tema del software libre implica encontrarse con multitud de definiciones y términos diferentes. Para unificar criterios, en este apartado vamos a enumerar los principales conceptos del curso y la definición que los autores vamos a utilizar cuando nos refiramos a ellos.

Por **software libre** entenderemos al software distribuido bajo un determinado tipo de licencia de uso que permite el acceso al código objeto y el código fuente del mismo, y por lo tanto su modificación y adaptación a las necesidades del usuario, y también la libre distribución y utilización de la aplicación. No debe asimilarse con software gratuito, la licencia con la que se distribuye y el uso de los derechos de autor que sus creadores realizan son su característica y diferencia principal, no su coste.

Por **derechos de autor** al sistema de protección de las creaciones humanas originales, literarias, artísticas o científicas, expresadas por cualquier medio o soporte, tangible o intangible, actualmente conocido o que se invente en el futuro, entre las que se incluye el software.

Por **licencia de uso** al instrumento legal (contrato) que se emplea por el titular de un software para otorgar permisos de uso del software a terceros, a cambio o no de una remuneración económica.

En cuanto a la nomenclatura relativa a las licencias de software libre en sentido amplio, usaremos la terminología siguiente:

- **Software libre y licencia libre:** seguiremos la práctica de la FSF, que usa el término de software libre para cualquier licencia que respete las cuatro libertades.
- **Software abierto y licencia abierta:** software que cumple las directrices de la definición de software de código abierto (OSD).
- **Software copyleft y “licencia con copyleft”:** aplicaciones y licencias que se distribuyen con una cláusula de copyleft robusto como la GPL.

- **Software y licencia propietaria:** aplicaciones distribuidas bajo licencias que no son libres.



## 2. Propiedad intelectual: derechos de autor y copyright

Los objetivos de esta unidad son:

1. Introducir la figura de los derechos de autor como sistema de protección de las creaciones intelectuales originales.
2. La concreta aplicación de los derechos de autor al software y los diferentes marcos normativos de aplicación.
3. Comprender los fundamentos de la figura de los derechos de autor: el objeto y los requisitos de protección.
4. Conocer las posibilidades de autoría que reconoce la normativa actual.
5. Estudiar los derechos que el sistema de derechos de autor otorga.
6. Analizar los límites a los anteriores derechos.
7. Comentar las medidas de protección frente a las potenciales infracciones de derechos.
8. Comparar los mecanismos de protección otorgados por el sistema continental y anglosajón (*copyright*).
9. Relacionar derechos de autor y software libre, con introducción del concepto *copyleft*.
10. Entender cómo se protegen unas obras muy especiales, las bases de datos.
11. Cerrar la unidad con una reflexión sobre la naturaleza cambiante de la materia y un enlace al próximo tema, las patentes.

## 2.1. El concepto tradicional de los derechos de autor

Para empezar, vamos a explicar el concepto tradicional de derechos de autor y su justificación histórica. Asimismo, vamos a diferenciar los dos grandes sistemas de protección, los derechos de autor del sistema continental y el *copyright* del sistema anglosajón.

### 2.1.1. Definición y objeto

El derecho de autor es un término jurídico que describe los derechos concedidos a los creadores por sus obras literarias, artísticas y científicas.

Una obra puede definirse como la expresión formal de una idea o sentimiento que se quiere comunicar al público. El tipo de obras que abarca el derecho de autor incluye: obras literarias, como novelas, poemas, obras de teatro, documentos de referencia, periódicos y **programas informáticos**; obras artísticas; obras científicas; bases de datos; películas, composiciones musicales y coreografías; obras arquitectónicas; publicidad, mapas y dibujos técnicos.

La protección tiene incidencia sobre la **forma**, el **continente**, la **expresión** de la idea creativa, pero no sobre el contenido. Ni el material de inspiración (hechos, fechas...) ni las ideas son protegidas por los derechos de autor. Así, el concepto de obra incluye los programas informáticos, en cuanto a su código, pero no el algoritmo interactivo que representa.



En la teoría general de los derechos de autor también existen ciertas prestaciones que son auxiliares de la creación sobre las que se reconocen derechos afines o conexos a los autores. Algunos derechos conexos protegen actividades creativas (artistas, intérpretes y ejecutantes...) y otros, actividades industriales (productores, entidades de radiodifusión...). Por ejemplo, la *Flauta Mágica* de Mozart es una obra de dominio público, pues su autor murió en 1791 y se puede representar y ejecutar libremente. No obstante, si se desea copiar una representación de la obra en un disco o colocarlo en un sitio web, se deberá solicitar permiso, pues hay unos artistas y productores que quizás tienen derechos conexos vigentes.



En los derechos de autor aplicados al software, los derechos conexos son inexistentes.

### 2.1.2. Origen de los derechos de autor

Para entender el concepto y el alcance de los derechos de autor, es necesario conocer su origen e historia y, por ello, a continuación vamos a realizar una breve reseña sobre el tema.

Los derechos de autor encuentran su origen en el momento histórico en que las obras se empezaron a explotar mediante la reproducción en soportes físicos (los copistas medievales) y, posteriormente, a distribuirse con la aparición de la imprenta. Los editores, como colectivo, se beneficiaron directamente del invento de la imprenta, pues las obras se transformaron en objetos comerciales con posibilidad de proporcionarles beneficios económicos. Pero, al mismo tiempo, el uso de la imprenta dio lugar a la aparición de la primera “piratería” intelectual y los editores presionaron a los legisladores para obtener protección y asegurar sus rendimientos económicos. De esta manera, el primer objetivo de los derechos de autor fue el de **regular y proteger a los editores contra el plagio**.

A partir de ese momento, la historia del derecho de autor se caracteriza por la concesión de privilegios a determinados impresores, aunque también se utilizó por parte de los gobernantes de la época como sistema para controlar y censurar las obras publicadas.



Parece que la primera ley sobre derechos de autor fue el Estatuto de la Reina Ana de 1710 en Inglaterra, que concedía a los autores derechos exclusivos de imprimir y distribuir sus obras durante un periodo limitado.

La evolución de los derechos de autor a nivel nacional, y posteriormente internacional, tiene relación directa con su extensión a distintos tipos de obras y la adaptación de sus normas a las características propias de cada tipo de obra a la que iba extendiendo la protección.

Como veremos a lo largo de este curso, la adaptación de los derechos de autor a los programas de ordenador y a las nuevas modalidades tecnológicas de difusión y plasmación de las obras han generado una revolución sin precedentes en las tradicionales estructuras jurídicas de los derechos de autor.

### 2.1.3. ¿Derechos de autor o copyright?

Los términos *derechos de autor* y *copyright* se utilizan frecuentemente como traducciones de un mismo concepto, pero es necesario tener en cuenta que bajo esta doble denominación se encuentran dos concepciones diferentes del derecho de autor que conviven (y a veces compiten) en el mundo actual:

- El término *author's right* –‘derecho de autor’ en español– está basado en la idea, nacida en la Europa continental, de que una obra está estrechamente relacionada con su creador y donde el acento se pone en el autor. La base de los derechos de autor es, en contraposición al *copyright*, el reconocimiento de unos derechos morales irrenunciables e intransmisibles a todo autor.
- El término *copyright* –‘derecho de copia’ en español– proviene de la tradición anglosajona y se basa en la idea de que lo más importante no es el autor, sino la obra y los derechos de propiedad que los autores tienen sobre sus creaciones, con los que se puede comerciar sobre la base de los principios económicos.

Como simplificación, podríamos decir que el sistema de derechos de autor es más personalista y protege la creación como extensión de la persona, en cambio, el sistema *copyright* es más colectivista y protege la creación para estimular a los creadores como beneficio de interés general.

Actualmente, se da una significativa coincidencia de regulación entre ambas filosofías jurídicas, gracias en buena parte a la internacionalización de la propiedad intelectual y a la función armonizadora de los tratados internacionales sobre la materia. No obstante, aún existen diferencias relevantes, como por ejemplo los derechos morales del autor.

Este curso va a centrarse en el sistema de derechos de autor, aunque a menudo vamos a realizar menciones al sistema de *copyright* para que podamos comprender las diferencias más relevantes. Además, en el apartado 8 de esta unidad vamos a realizar un resumen de las principales características del sistema del *copyright*.

## 2.2. Derechos de autor y software

A continuación, vamos a estudiar la evolución que ha llevado a la protección del software mediante la figura de los derechos de autor y el marco normativo de los derechos de autor aplicados al software.

### 2.2.1. Evolución histórica de la protección

Inicialmente, los programas de ordenador no se comercializaban de manera independiente pues se distribuían con el hardware y su protección se confundía con la del conjunto del bien comercializado, el ordenador (fenómeno conocido como *bundling*). Además, no existían tecnologías que hicieran posible la realización de copias de los mismos o su utilización fuera de su ordenador de manera generalizada, por lo que no existía una preocupación por su protección.

La necesidad de protección empezó a plantearse a finales de los años sesenta cuando a causa de la legislación antimonopolio norteamericana, IBM se vio obligada a separar sus negocios de hardware y software. Consecuentemente, los programas de ordenador empezaron a estandarizarse y comercializarse en soportes movibles, y la copia se generalizó.



En ese momento, la necesidad de protección jurídica autónoma para los programas de ordenador pareció obvia por cuatro motivos principalmente:

- a) Proteger las inversiones realizadas en la creación de programas de ordenador.

- b) Incitar a la divulgación de los programas de ordenador a más amplios sectores de la humanidad.
- c) Proteger la actividad humana creadora por razones de justicia y de política económica.
- d) Proteger esta tecnología de extrema vulnerabilidad a la copia.

El sentimiento de necesidad de protección de los programas de ordenador generó un debate sobre qué figura legal era la más adecuada para protegerlos. En ese primer momento, se entendió que podía ser objeto de protección por la figura jurídica de las **patentes** (el programa de ordenador como invención) y así lo interpretó la jurisprudencia de Estados Unidos en varias ocasiones. Sin embargo, a finales de los sesenta y principios de los setenta, los distintos legisladores nacionales y los tratados internacionales empezaron a rechazar la protección del software como patente. Realizaremos un análisis más detallado de las principales razones de este rechazo en la unidad 3.

En consecuencia, los países más industrializados (con gran presión de las grandes empresas informáticas) buscaron vías alternativas de protección del software:

- a) Por un lado, se planteó la posibilidad de crear una **protección específica o *sui generis*** (el programa como una creación de tipo nuevo).

Se pretendía establecer un mecanismo de protección con una duración de 4 a 9 años y un registro obligatorio, aunque esta iniciativa nunca llegó a prosperar.

Como seguidores de esta corriente, destacan los trabajos iniciados a principio de los años setenta por la Organización Mundial de la Propiedad Intelectual (conocida por su acrónimo OMPI) que desembocan en las Disposiciones Tipo sobre Protección de Software de 1978 y el Proyecto de Tratado Internacional sobre la Protección de Programas de Ordenador de 1983, así como otros

proyectos legislativos nacionales como el del Ministerio de Industria y Tecnología de Japón (MITI) de 1982 o el del Gobierno brasileño de 1984.



Para saber más, consúltese:

E. Galbi (1970). "Proposal for the Protection of Computer Programs". *Bulletin of the copyright Office* (vol. 28, pág.17 y sig.).

- b) Por otro lado, se planteó la posibilidad de proteger el software mediante la figura de los **derechos de autor** y *copyright* (el programa como obra literaria), y ésta fue la opción que triunfó.



Como resultado de estos debates, se llegó a un principio generalmente aceptado, según el cual los programas informáticos deberían estar protegidos por el derecho de autor, mientras que los aparatos que usaran programas informáticos o inventos relacionados con este tipo de programas deberían estar protegidos por patentes.

La solución es deudora en buena parte de las **ventajas que ofrece la protección** de una obra por el sistema de derechos de autor:

- a) Simplicidad: la obra no necesita inscripción en registros, ni cumplimiento de formalidades, ni está sujeta a examen previo para ser protegida.
- b) Automatismo: el derecho del autor nace por el mero hecho de la creación original.
- c) Economía: la protección es económica.
- d) No se requiere novedad, sino, como veremos, originalidad.
- e) La protección se extiende a la documentación accesoria.
- f) Extensión internacional: la protección se otorga, en virtud de tratados internacionales, para casi todo el mundo.

- g) Armonización: más o menos completa, de los derechos de autor a nivel internacional.



Parece que Filipinas fue el estado donde se adoptó en 1972 la primera legislación nacional que declaró aplicable el derecho de autor a los programas de ordenador. Sin embargo, fue Estados Unidos el principal exponente de esta segunda opción, al reformar su Copyright Act el 1976 para incluir en la misma los programas de ordenador. Le siguieron, en 1985, los acuerdos adoptados a nivel internacional en el seno de la OMPI, así como también Alemania, Japón Gran Bretaña, Canadá y Francia.

España también se incorporó a este segundo movimiento legislativo en aras a asegurar la protección del software por la figura jurídica de los derechos de autor con la reforma de la Ley de Propiedad Intelectual de 1987 (con la inclusión de los artículos 95 a 100 relativos a la regulación específica de los programas de ordenador por el derecho de autor), si bien antes de dicha ley gran parte de la doctrina ya consideraba que se podía entender que los principios de la Ley española de Propiedad Intelectual de 1879 podían haber sido utilizados para interpretar que los programas de ordenador podían ser objeto de protección bajo el derecho de autor.

No obstante, la armonización de la protección a nivel internacional, la equiparación de los programas de ordenador a una obra literaria y su protección de los derechos de autor son temas complicados, pues se basan en la aplicación de unos conceptos que fueron pensados para unas obras muy diferentes y que, por tanto, son objeto de muchas excepciones. Además, las legislaciones de los países continentales han incorporado la solución de protección del software mediante derechos de autor concebida para el sistema anglosajón del *copyright*, provocando dificultades con las características tradicionales del sistema continental de derechos de autor, en especial el reconocimiento de derechos morales.

**Nota**

Aunque este debate parecía cerrado, en la unidad 3 comprobaremos que ha vuelto a la actualidad, con la aparición de diversos conflictos jurisdiccionales sobre las patentes de software y de una propuesta de Directiva Europea sobre patentes de software.

### 2.2.2. Marco normativo

A partir de la decisión enunciada en el apartado anterior de proteger el software como una obra literaria, el software fue integrado en el marco normativo de los derechos de autor. Por lo tanto, se hace necesario conocer este marco normativo.

En este apartado vamos a realizar un recorrido por los principales ordenamientos jurídicos que configuran la protección de los derechos de autor, empezando desde una perspectiva internacional hasta llegar a la regulación europea y española.

#### Derechos de autor en un ámbito internacional

Como hemos comentado, una de las principales ventajas de los derechos de autor como medida de protección del software es el automatismo que ofrece protección desde el momento de la creación original. Debido a la naturaleza intangible y de fácil difusión del software, sobre todo a través de Internet, es necesaria la protección a nivel internacional para una armonización del automatismo y del contenido de los derechos.

A continuación, vamos a realizar un breve recorrido por los principales tratados e instituciones internacionales sobre la materia.

El más importante generador de normativa de derechos de autor a nivel internacional es la Organización Mundial de la Propiedad Intelectual (conocida por sus acrónimos, OMPI o WIPO, en inglés). La **OMPI** es un organismo internacional cuyo propósito es colaborar para que los derechos de los creadores y los dueños de propiedades intelectuales estén protegidos en todo el mundo.

En 1974, la OMPI se convirtió en una agencia especializada del sistema de organizaciones de las Naciones Unidas y se le encomendó administrar los asuntos de propiedad intelectual. Actualmente, la OMPI administra 21 tratados y desarrolla un rico y variado programa de trabajo.

Los tratados más importantes en la materia generados en el seno de esta organización son:

- **Convención de París** para la Protección de la Propiedad Industrial de 1883.
- **Convenio de la Unión de Berna** para la Protección de Obras Literarias y Artísticas de 1886.
- **Convenio Universal de Ginebra** sobre los Derechos de autor de 1952.

La última revisión de las Convenciones de Berna y de Ginebra es de 1971, por lo que el software no está incluido en la categoría de obras protegidas por estos dos convenios. No obstante, son dos convenios muy importantes (sobre todo el de Berna) pues han marcado los principios básicos de todas las legislaciones sobre derechos de autor posteriores.



La Convención de Berna se apoya en tres principios básicos:

- a) Principio de trato nacional: las obras originadas en alguno de los estados contratantes podrán recibir en cada uno de los demás estados contratantes la misma protección que éstos otorgan a las obras de sus propios ciudadanos. Extiende, por lo tanto, no sólo los derechos materiales, sino también el automatismo de la protección a nivel internacional.
- b) La protección será automática y no debe estar condicionada al cumplimiento de formalidad alguna.



- c) La convención abarca también los derechos morales, es decir como veremos en los siguientes apartados, el derecho de reclamar la autoría de la obra y el derecho de oponerse a cualquier modificación de la misma. En este punto, es importante tener en consideración que el sistema anglosajón del *copyright* no reconoce estos derechos morales y por lo tanto la aplicación de este convenio no fue completa.

La **Organización Mundial del Comercio** (conocida por sus acrónimos OMC o WTO en inglés) es una organización internacional que se ocupa de las normas que rigen el comercio entre los países. La OMC se interesa a mediados de los años noventa por los derechos de autor debido al crecimiento del comercio internacional de servicios y de obras susceptibles de protección por los derechos de autor.

El **Acuerdo sobre los Aspectos de la Propiedad Intelectual Relacionados con el Comercio** de 1994 (ADPIC o TRIPS en inglés) es el acuerdo más importante sobre la materia de derechos de autor adoptado en el seno de esta organización. El ADPIC fue adoptado como parte del acuerdo de comercio multilateral de la Ronda Uruguay (1986-1994) y es el sucesor del Acuerdo General sobre Aranceles Aduaneros y Comercio (*General Agreement on Tariffs and Trade*, GATT).

Lo más relevante del **Acuerdo ADPIC** sobre derechos de autor es que:

- Obliga a los signatarios a acatar las disposiciones de la Convención de Berna, salvo los requisitos de la misma sobre derechos morales.
- Protege los programas de ordenador como obras literarias y las bases de datos como compilaciones.
- Para los estados parte del ADPIC, las disposiciones de la Convención de Berna son aplicables a los programas de ordenador, sea o no sea signatario el estado de la Convención de Berna.

- Impone a los estados parte la obligación de conceder a los dueños de programas de ordenador el derecho de autorizar o prohibir el alquiler de sus productos.



El texto completo del Acuerdo ADPIC y una explicación de sus disposiciones en términos sencillos se encuentran en la dirección <http://www.wto.org>. El texto completo de todos los tratados OMPI y una lista de sus signatarios figuran en la página web de la OMPI, en <http://www.wipo.int>.

La creación legislativa a nivel internacional no ha finalizado, y así, al final de 1996, se celebraron dos nuevos tratados en el marco de la OMPI para adaptar los derechos de autor a los nuevos acontecimientos del mercado y a la evolución de las tecnologías, principalmente a Internet. Por ello, con frecuencia reciben el nombre de “Tratados Internet”: el Tratado de la OMPI sobre Derechos de Autor y el Tratado de la OMPI sobre Interpretación o Ejecución y Fonogramas.

El **Tratado de la OMPI sobre Derechos de Autor (TODA o WCT, en inglés)** entró en vigor el 6 de marzo del 2002 y se ocupa de la protección de los autores de obras literarias y artísticas, incluyendo programas informáticos y bases de datos originales. Como estudiaremos posteriormente, la Unión Europea y Estados Unidos han sido pioneros en la aplicación de las disposiciones de este tratado al adoptar la Directiva de Derechos de Autor y la *Digital Millenium Copyright Act (DMCA)*.

Lo más relevante del **Tratado TODA sobre Derechos de Autor** es que:

- a) Universaliza la Convención de Berna y redefine sus conceptos, garantizando a los titulares de derechos de autor que seguirán siendo protegidos de forma adecuada y eficaz cuando sus obras se divulguen a través de nuevas tecnologías y sistemas de comunicación, tales como Internet, creando nuevos derechos aplicables al entorno de la red.

- b) Introduce la protección legal de las medidas tecnológicas de protección de los derechos de autor que estudiaremos más adelante.

## Derechos de autor en el derecho europeo

Tradicionalmente, en Europa han sido los estados los que han regulado la materia de la propiedad intelectual, por lo que existe una enorme casuística en la regulación del tema. No obstante, en este momento el centro productor de la legislación sobre propiedad intelectual en Europa ya no está formado tanto por los estados como por la Comisión Europea.

La función de la Comisión Europea es compleja, ya que, aparte de armonizar las legislaciones estatales, debe a su vez atender los compromisos internacionales adquiridos principalmente, como hemos visto, con la OMPI y la OMC. El tema es complejo y no existe una armonización completa en temas como, por ejemplo, los derechos morales, al convivir en Europa el sistema del *copyright* con el de derechos de autor.

La normativa europea sobre derechos de autor y programas de ordenador encuentra su base en la **Directiva 91/250/CEE del Consejo, de 14 de mayo de 1991**, relativa a la protección jurídica de programas de ordenador, modificada por la **Directiva 93/98/CEE del Consejo de 29 de octubre de 1993**.

La Directiva de programas de ordenador establece que:

- a) Los programas de ordenador se protegerán por los derechos de autor como "obras literarias", de manera acorde con la Convención de Berna.
- b) La protección abarca la expresión, pero no las ideas o principios.
- c) No se reconoce al titular un derecho exclusivo de uso, si bien se incluyen en el derecho de reproducción los actos de uso que comporten reproducción total o parcial.
- d) El sistema de patentes y los derechos de autor son compatibles.

Asimismo, también es relevante en la materia la **Directiva 96/9/CE** sobre protección jurídica de bases de datos, que armoniza la protección de las bases de datos a nivel europeo y que estudiaremos en el apartado 10 de esta unidad.

Últimamente, la labor legislativa se ha acelerado con la **Directiva 2001/29/CE, de 22 de mayo del 2001** relativa a la armonización de determinados aspectos de los derechos de autor y derechos afines a los derechos de autor en la sociedad de la información (Directiva de Derechos de Autor o de la Sociedad de la Información).

La Directiva de Derechos de Autor pretende actualizar toda la normativa vigente, manteniendo un elevado nivel de protección y cumpliendo los compromisos asumidos en virtud del Tratado de la OMPI de Derechos de autor (TODA). El nacimiento de esta directiva ha sido realmente difícil y ha implicado un largo debate. Actualmente, aún no ha sido implementada en muchos estados de la Unión Europea incumpléndose el plazo de 19 meses establecido (22 de diciembre del 2002). De hecho, a noviembre del 2003, sólo ha sido transpuesta en Inglaterra, Austria, Dinamarca, Alemania, Grecia e Italia.

En el artículo 1 de la Directiva se revalida expresamente la vigencia de la Directiva 91/250 para el software. No obstante, esto no significa que esta directiva no sea relevante para el software. Pensemos, por ejemplo, en las medidas tecnológicas de protección de los artículos 6 y 7 de la directiva y en la utilización del software como mecanismo para desactivar esas medidas.

Lo más relevante de la **Directiva de Derechos de autor** es que:

- a) Se amplían los conceptos de reproducción y de comunicación pública (ahora también derecho de poner a disposición del público) que se aplican tanto al software como a la documentación complementaria distribuida por Internet;
- b) En los artículos 6 y 7 se establece la protección legal de las medidas tecnológicas de protección que estudiaremos más adelante.

## Derechos de autor en el Derecho Español

El Código civil español, en sus artículos 428 y 429, remite la regulación de la propiedad intelectual a una ley especial, y declara la aplicación supletoria de las reglas generales establecidas en el mismo sobre la propiedad para lo no específicamente previsto en dicha ley especial.

Esta ley actualmente es la Ley de Propiedad Intelectual, cuyo Texto Refundido fue aprobado por el Real Decreto Legislativo 1/1996, de 12 de abril (en adelante, LPI). El citado texto –modificado por la Ley 5/1998, de 6 de marzo– constituye, junto con las normas reglamentarias de desarrollo parcial, el marco legal por el que se regula la propiedad intelectual en España, sin perjuicio de lo establecido en la materia por los convenios y tratados internacionales de los que nuestro país es parte y que hemos estudiado en los apartados anteriores.

Existe un **Proyecto de Ley de Propiedad Intelectual** para implementar la Directiva de Derechos de Autor en la Sociedad de la Información. A día de hoy (febrero del 2004), la propuesta de noviembre del 2002 es la única versión pública del proyecto. Una segunda versión, fechada el 23 de enero del 2003, incluye algunas modificaciones que no han sido comunicadas oficialmente. Aunque en este momento el debate sobre el proyecto parece haberse detenido, fuentes del Ministerio de Cultura Español informan que la aprobación de una nueva Ley de Propiedad Intelectual será una de las prioridades del nuevo ejecutivo.



Los programas de ordenador en España se regulan por la Ley de Propiedad Intelectual, desde que la derogada Ley 16/1993, de 23 de diciembre, incorporó al Derecho español de la Directiva 91/250/CEE, de 14 de mayo de 1991 sobre la protección jurídica de los programas de ordenador. Actualmente, la materia se regula en los **artículos 95 y siguientes de la LPI de 1996**.

### 2.2.3. Sumario de conocimientos

El software está protegido por el sistema de los derechos de autor, de modo que dicho objeto legal se asimila a una obra literaria. La elección de este sistema de protección se fundamentó en las características de automatismo, armonización internacional, simplicidad y no incompatibilidad con el sistema de patentes, del sistema de derechos de autor.

A nivel internacional, los principales tratados sobre la materia son:

- Convención de París para la Protección de la Propiedad Industrial de 1883.
- Convenio de Berna para la Protección de Obras Literarias y Artísticas de 1886.
- Acuerdo sobre los Aspectos de la Propiedad Intelectual Relacionados con el Comercio de 1994 (ADPIC, o TRIPS en inglés).
- Tratado de la OMPI sobre Derecho de Autor de 1996 (Tratado TODA o WCT en inglés).

A nivel europeo, la normativa más importante sobre la materia comprende:

- Directiva 91/250/CEE del Consejo, de 14 de mayo de 1991, relativa a la Protección Jurídica de Programas de Ordenador (Directiva de los Programas de Ordenador).
- Directiva 96/9/CE sobre Protección Jurídica de Bases de Datos.
- Directiva 2001/29/CE, de 22 de mayo del 2001 relativa a la armonización de determinados aspectos de los derechos de autor y derechos afines a los derechos de autor en la sociedad de la información (Directiva de Derechos de Autor o de la Sociedad de la Información).

En España, la legislación aplicable es la Ley de Propiedad Intelectual, cuyo Texto Refundido fue aprobado por el Real Decreto Legislativo 1/1996, de 12 de abril (LPI), pendiente de modificación.

### 2.3. Objeto y requisitos de protección

En este apartado vamos a intentar comprender cuál es el objeto concreto de protección de los derechos de autor, en especial en su aplicación al software. Es decir, qué componentes del software se protegen y qué requisitos deben cumplir éstos para poder ser protegidos.

#### 2.3.1. Objeto de protección



Son objeto de propiedad intelectual las creaciones humanas originales, literarias, artísticas o científicas, expresadas por cualquier medio o soporte, tangible o intangible, actualmente conocido o que se invente en el futuro.

Para estudiar la protección del software –como cualquier otra obra susceptible de protección– por derechos de autor, debemos tener en cuenta las características de su protección:

- a) Únicamente va a ser protegible la **expresión** de un programa de ordenador, y no las ideas que contiene.
- b) Sólo va a protegerse al programa en cuanto **bien inmaterial**, independientemente del soporte en el cual se fije el mismo.

Para continuar avanzando en el estudio, se hace imprescindible delimitar el objeto de protección y conocer cuáles van a ser los elementos del software susceptibles de protección por vía del derecho de autor. Debemos, por lo tanto, adentrarnos en sus componentes y etapas de creación.

Aunque en el lenguaje habitual los términos software y programa de ordenador sean utilizados como sinónimos, jurídicamente el concepto de “programa de ordenador” está comprendido en el concepto de “software”.

La mejor ilustración de la afirmación anterior la encontramos en las disposiciones tipo de la OMPI para la protección del software que definen al **software** como el conjunto de tres elementos:

- a) **El programa de ordenador:** definido como el conjunto de instrucciones capaces –cuando se incorporan en un medio técnico legible por el ordenador– de hacer que una máquina con capacidad de procesar la información indique, actúe u obtenga una función, tarea o resultados concretos.
- b) **La descripción del programa:** definida como una presentación procedimental completa en forma verbal, esquemática o de cualquier otro modo, de forma suficientemente detallada para determinar un conjunto de instrucciones que constituyan el programa de ordenador correspondiente.
- c) **El material de apoyo:** entendido como cualquier material creado para ayudar a la comprensión de un programa de ordenador, como por ejemplo las descripciones de un problema o las instrucciones del usuario.

De la anterior definición se deriva que el programa de ordenador es el elemento principal del software y por eso nosotros vamos a centrarnos en el estudio de los programas de ordenador, al que consideraremos sinónimo del término *software*, aunque a veces haremos referencia a otros elementos más accesorios.

Otra definición la encontramos en el artículo 96.1 de la LPI, que establece: “se entenderá por programa de ordenador toda secuencia de instrucciones o indicaciones destinadas a ser utilizadas, directa o indirectamente, en un sistema informático para realizar una función o una tarea o para obtener un resultado determinado, cualquiera que fuere su forma de expresión y fijación”.

Es muy relevante la frase final del artículo 96.1 de la LPI “cualquiera que fuera su forma de expresión y fijación”, pues gracias a ella, la doctrina incluye entre las obras protegidas todas las fases de elaboración del programa desde el momento en que hay una descripción del mismo, en forma gráfica (ordinograma o diagrama de flujos) o verbal, suficientemente detallada para determinar un conjunto de instrucciones; incluye la expresión en cualquier lenguaje de progra-



mación y permite, asimismo, pensar en la protección del conjunto de instrucciones fijado en un producto semiconductor (microcódigo y *firmware*). La expresión “directa o indirectamente” abarca tanto los programas escritos en lenguaje de alto nivel (código fuente), como su versión en código-máquina (código objeto).



En resumen, podemos concluir que el derecho de autor protege al software en cualquiera de sus expresiones posibles y en su totalidad. Lo que no protege son las ideas y los principios.

Así, los derechos de autor protegen principalmente los siguientes elementos del software:

- El programa de ordenador
- La documentación preparatoria
- Los manuales de uso y documentación técnica de apoyo
- El código fuente
- El código objeto
- La arquitectura del programa: incluye diagrama de flujo, modelos de datos, diagramas en UML, etc.
- Las interfaces (incluyendo elementos gráficos, sonidos, tipografías y otros elementos audiovisuales).
- Las bases de datos también son protegidas.

### 2.3.2. Requisitos para la protección

Para que el software, como cualquier otra obra, sea protegible por los derechos de autor, debe cumplir unas determinadas condiciones enumeradas en el artículo 10.1 LPI y que pueden resumirse en la si-

#### Nota

Veremos la protección de las bases de datos en el apartado 9 de esta unidad.

guiente sentencia: “las obras son aquellas creaciones humanas originales expresadas en cualquier medio o soporte”.

Tres son entonces las condiciones:

#### a) Creación humana

El programa debe ser fruto del intelecto de un autor, consecuencia de su actividad y por lo tanto calificable de creación. Los programas de ordenador creados por máquinas no son considerados protegibles a excepción del software compilado (creado por el compilador), que se asimila al software creado por los seres humanos.

#### b) Expresada por cualquier medio o soporte

Los bienes inmateriales precisan de un instrumento o medio a través del cual hacerse perceptibles en el mundo exterior. Por lo tanto, según el art. 10.1 LPI, para que sea posible obtener derechos de autor para una obra, ésta deberá estar “contenida en un medio de expresión tangible o intangible, conocido actualmente o que se invente en el futuro”. En el software, el medio de expresión puede ser un disquete, un disco duro, un CD-ROM, un *flash card*, etc.



En otros países, por ejemplo Estados Unidos, se exige la fijación en un medio tangible de expresión del cual pueda ser percibido, reproducido, o de otra manera comunicado, o bien directamente o bien con ayuda de una máquina o dispositivo.

#### c) Original

Para ser protegida, una obra debe ser original. No importa el mérito, la destinación, el grado de elaboración, la licitud o ilicitud, ni la prioridad temporal. Como es obvio, no es fácil definir originalidad. De hecho, supone la cuestión más controvertida en los últimos años en la tradición continental. La tradición más ortodoxa del sistema de derechos de autor exige una “huella de la personalidad”, aunque no existe unanimidad en las legislaciones nacionales en el nivel de ori-

ginalidad exigible. Actualmente, se admite que el concepto es de geometría variable.

En el software, es problemático delimitar cuándo existe originalidad al ser una creación utilitaria, donde el margen de maniobra es escaso. Es importante definir qué amplitud tendrá el criterio aplicado en la apreciación de la originalidad de un programa, pues ello incide no sólo en el establecimiento del nivel que deberá alcanzar para acceder a la protección de la ley, sino también en la comparación entre programas con el fin de determinar la existencia o no de una copia no autorizada.

Tal dificultad va acompañada de un temor a que un criterio demasiado exigente obstaculice la creación de nuevos programas y suponga una traba para la libre competencia. La aplicación de un criterio objetivo como el de novedad, exigido para las invenciones patentables, provocaría la desprotección de muchos programas cuyas funciones básicas se basan en el estado de la ciencia (*state of the art*). Por otra parte, un criterio excesivamente amplio podría dificultar la apreciación de reproducciones no autorizadas en las obras que no constituyan una copia literal o servil.

Para solucionar esta dificultad, se ha optado por un **criterio de nivel bajo** y en general se considera que es suficiente que el software sea el resultado de un esfuerzo personalizado, o sea, que no esté copiado para ser considerado original, excluyendo los criterios cualitativos y estéticos en la comprobación de la originalidad.

La prueba de originalidad de una obra suele tener dos aspectos. El primero es que la obra deberá ser original del autor, en el sentido de que realmente haya sido creada por éste de modo independiente, no copiada de otras obras. El segundo aspecto es que la obra debe contener la cantidad suficiente de creatividad para que no se la pueda considerar como algo trivial.



El *copyright*, por su parte, se conforma con un nivel de originalidad inferior, exigiendo sólo que la creación resulte de un esfuerzo personalizado, o sea, que no sea copia. En la medida en que el software debe su origen

a su autor, esto es, que fue independientemente creado y no copiado de otras obras, será considerado original.

## 2.4. Autores y titulares de derechos

En este apartado vamos a explicar los diferentes actores que pueden estar implicados en el proceso de creación y distribución de una obra, y las diferentes tipologías de autoría reconocidas. Dada la multitud de intervinientes en el proceso de creación de un software, y especialmente del software libre, este apartado adquiere una relevancia especial.

### 2.4.1. El autor



Se considera **autor** a la persona física (o jurídica, como veremos más adelante) que crea una obra y, en general, la propiedad de la misma corresponde al autor por el solo hecho de su creación original por el principio de automatismo. Como hemos visto, no será necesario cumplir ningún tipo de formalidad ni requisito adicional, la creación original implica la protección por derechos de autor.

#### Nota

Los derechos morales se estudian en el apartado 2.5.1.

Además, como veremos al estudiar los derechos morales, la condición de autor tiene un carácter irrenunciable; no puede transmitirse *inter vivos* ni *mortis causa*, no se extingue con el transcurso del tiempo, así como tampoco entra en el dominio público ni es susceptible de prescripción.

La obra, el software en nuestro caso, puede ser creada por el esfuerzo y labor de una persona/autor o puede ser el producto del esfuerzo de varias. Este segundo supuesto es el más habitual en la práctica y, en estos casos, la atribución de autoría puede ser más compleja. A continuación, vamos a estudiar diferentes tipologías de autorías múltiples.

## Obra en colaboración



Los artículos 7 y 97.3 de la LPI establecen que será una obra en colaboración aquella obra que sea el resultado unitario de la colaboración de varios autores, siempre que sea posible separar las aportaciones de cada uno y explotarlas independientemente.

En este caso, los derechos sobre la misma corresponden a todos ellos en la proporción que determinen. A falta de pacto, los autores son titulares a partes iguales. La divulgación y modificación de la obra precisa el consentimiento de todos los autores. No obstante, una vez que la obra ha sido divulgada, ningún coautor puede rehusar injustificadamente su consentimiento para su explotación en la forma en se divulgó. Los coautores pueden explotar separadamente sus aportaciones, siempre que no se cause perjuicio a la explotación común.

## Obra colectiva



La obra colectiva es la creada por iniciativa y bajo la coordinación de una persona natural o jurídica, que la edita y divulga bajo su nombre. Constituye la reunión de aportaciones de diferentes autores de tal forma que es imposible la atribución a cualquiera de ellos de un derecho sobre el conjunto de la obra y cada aportación se funde en una creación única y autónoma.

El resultado es una obra que tiene un valor añadido a la simple unión de aportaciones, como sería una obra en colaboración.

La regla general es que se consideran autores originarios a la persona física que crea una obra, pero este supuesto se ve desbancado por el artículo 97.2 LPI al establecer que será autor de la obra colectiva, salvo pacto en contrario, la persona física o jurídica que la edite o divulgue bajo su nombre. Por lo tanto, de producirse el supuesto de que sea una persona jurídica la que edite o divulgue la obra bajo su nombre, la persona jurídica será la titular original de los derechos.

### Obra creada por asalariado o por encargo

Un supuesto específico es el de los programas de ordenadores creados en el seno de una relación laboral. Salvo pacto en contrario en el contrato de trabajo, los derechos patrimoniales de los programas de ordenador creados por un programador asalariado, serán titularidad del empresario de manera automática y exclusiva.



Así, el artículo 97 de la LPI establece que:

“cuando un trabajador asalariado crea un programa de ordenador, en el ejercicio de las funciones que le han sido confiadas o siguiendo las instrucciones de su empresario, la titularidad de los derechos económicos correspondientes al programa de ordenador así creado, ... corresponderán, exclusivamente, al empresario, salvo pacto en contrario.”

Para que el empleador pueda ser considerado el titular de los derechos sobre ese software, son necesarios dos requisitos:

- a) que la relación laboral esté materializada en un contrato de trabajo, y
- b) que el programa haya sido creado por el empleado en el ejercicio de sus funciones o siguiendo instrucciones del empresario.

Como veremos más adelante, los derechos patrimoniales no son los únicos que puede ostentar un autor. Así, esta cesión legal de los derechos patrimoniales al empresario no impide que los derechos morales continúen siendo titularidad inalienable del programador en los sistemas continentales.



La excepción al principio de autoría por persona física establecido en el artículo 97.2 LPI ha propiciado que en la actualidad muchas compañías de software utilicen la

figura de la obra colectiva para el software desarrollado por sus trabajadores para intentar evitar que la autoría recaiga en los programadores y éstos tengan derechos morales sobre la obra. De esta manera, se posibilita crear una figura asimilable al sistema de *copyright* que permite negar los derechos morales a los programadores asalariados. En este supuesto, el empresario será el titular de **todos los derechos** sobre las obras (patrimoniales y morales).

Otro supuesto específico es el de los **programas de ordenador creados por encargo**. La directiva y la legislación española han optado por no regular el tema y ha sido la jurisprudencia la que ha establecido que, salvo pacto en contrario, no existirá transmisión de derechos y la titularidad será del creador y no de la persona (física o jurídica) que realiza el encargo (el cliente). Excepcionalmente, si el cliente se encuentra involucrado en el proceso creativo del software, la autoría podría convertirse en colectiva.

### Autoría en el software

Según lo estudiado en los apartados anteriores, la **autoría en el software tradicional** permite básicamente tres modalidades de autoría:

- a) La empresa-titular en el supuesto de software creado en el marco de una relación laboral entre el titular/empleador y los programadores.
- b) La consultoría-titular en el supuesto de compañías que crean software por encargo.
- c) Finalmente, están todos los supuestos de autoría independiente (obra en colaboración, colectiva, autoría individual...) según el concreto modelo de desarrollo del software.

Las particulares del **modelo de desarrollo del software libre**, añaden dificultades a las modalidades tradicionales y nos obligan a realizar un análisis específico de su aplicación al software libre.

De esta manera, existen diferentes posibilidades de autoría en el software libre:

- a) Algunas aplicaciones pueden considerarse **obras en colaboración** si el concreto modelo de desarrollo del software implica que varios autores contribuyan a la creación de una sola aplicación (obra), y las contribuciones de cada autor puedan distinguirse (por ejemplo, con avisos de autoría en los módulos).

Como hemos visto, bajo ese régimen cada autor es titular de su contribución, aunque la explotación de la obra se realiza conjuntamente. Esto implica que si se opta por un modelo de desarrollo del software de este tipo, es esencial acordar entre los autores, tan pronto como sea posible, la licencia que se aplicará al software y el régimen de explotación de la obra resultante.



El concepto de obra en colaboración, obra donde las contribuciones pueden distinguirse, no existe en el sistema del *copyright*. Las posibilidades son de co-titularidad bajo la figura del *joint-ownership* o la autoría independiente.

- b) Otras aplicaciones de software serán **obras colectivas** si no pueden diferenciarse las contribuciones de los diferentes programadores.

#### Ejemplo

Ejemplos de aplicaciones de software libre que podrían considerarse como obras colectivas (si fueran desarrolladas y diseminadas bajo el sistema de derechos de autor) incluyen el programa Mozilla, que es coordinado por Mozilla.org, o las aplicaciones de la Free Software Foundation, que coordina el desarrollo de varias aplicaciones distribuidas por la misma bajo la licencia GPL (por ejemplo, GNOME), en aquellos programas de ordenador que los contribuyentes no desmarcan o identifican su contribución.



Para prevenir posibles problemas futuros, los proyectos de desarrollo de software libre que opten por esta opción deben asegurarse de que cada autor-contribuyente transfiere por escrito todos los derechos a la entidad coordinadora para que ésta pueda administrar correctamente los derechos de propiedad intelectual de la aplicación, en concreto, determinar el régimen de licencia, asegurar la defensa de cualquier infracción, etc. Un ejemplo de esta política preventiva son las condiciones de la FSF para cualquier contribución realizada por programadores que aportan más de 10 líneas de código a un proyecto coordinado por ella: deben transferir la titularidad del código a la FSF.

La práctica de algunas iniciativas de software libre de crear un software sin la coordinación de una organización o persona producirá una obra de incierta clasificación en el Derecho español que requerirá de esta figura, como hemos visto al estudiar las obras colectivas.



En la creación del software libre, es usual la aportación de diferentes programadores. Para saber quién es el autor, quién ostenta los derechos sobre la obra y quién está legitimado para ejercitar acción por copia, es necesario delimitar si el resultado es una obra colectiva o una obra en colaboración. La respuesta dependerá del concreto modelo de desarrollo utilizado, de la existencia o no de un coordinador, de la separabilidad o no de las diferentes aportaciones y por supuesto de los pactos concretos establecidos.

## Obra original y obra derivada

Como hemos estudiado, una obra original es aquella obra que constituye una creación autónoma e independiente, aunque se publique conjuntamente con otras.



Una **obra derivada**, en cambio, es aquella obra nueva que incorpora una obra preexistente, sin la colabora-

ción del autor de esta última, aunque sí con su autorización (para ser obra derivada lícita).

Existirán, por lo tanto, dos obras diferenciadas y con dos titularidades: la obra original y la obra derivada. Aquella obra que incorpora una obra preexistente pero sin autorización (expresa o implícita a través de la colaboración del autor) produce una infracción de los derechos de autor del titular de la obra original.

En el software libre éste es un tema polémico, por las dificultades de realizar una diferenciación entre obra derivada, obra en colaboración y obra colectiva. Las dificultades se plantean básicamente en dos entornos:

- a) En relación con el **sistema de creación** del software libre, esta diferenciación es muy relevante, pues como hemos visto, él mismo se basa en la incorporación, agregación y modificación de obras preexistentes. El modelo usual de desarrollo de software libre implica una cantidad importante de contribuyentes a la creación de una misma aplicación. Algunos autores aportan código nuevo (por lo tanto, crean obras en colaboración o colectivas), otros corrigen o mejoran un código existente (por lo tanto, creando una obra derivada). Asimismo, algunos autores agregan un aviso de autoría, otros no.

La clave para la prevención de conflictos de autoría en un proyecto de desarrollo de software libre se encuentra en la “conciencia de contribución” de cada autor: a falta de un acuerdo explícito sobre las condiciones de colaboración, los que gestionan el software deben basarse en una cesión implícita de derechos (no permitido legalmente) o ausencia de reclamo por parte de los programadores contribuyentes contra cualquiera que distribuya públicamente el software (Source-Forge, etc.) o realice otro acto de explotación restringido.

- b) En cuanto al **uso y la explotación** del software libre, esta diferenciación entre obra original y derivada también es importante.

La mayoría de las licencias de uso permiten la modificación (adaptación, traducción, etc.) de las aplicaciones y, por lo tanto, la creación de obras derivadas. Algunas licencias de software libre controlan la difusión y el uso de estas obras derivadas (por ejemplo, la GPL), mientras que otras no (la BSD).

Es importante entender bien la definición de obra derivada para saber si un desarrollo realizado sobre la base de una aplicación libre (por ejemplo, integrándola, modificándola, usándola, etc.) puede considerarse:

- una obra derivada de la original, o
- una obra original que utilice la misma pero no la modifique.

**Nota**

Este tema se estudia con detenimiento en las unidades 6 y 7, y, en relación con la licencia GPL, veremos que no condiciona las aplicaciones que simplemente usan o se vinculan con la obra original.

### 2.4.2. Titular original y titular derivado

Como hemos estudiado, cuando se crea una obra, software en nuestro caso, el **titular original** de los derechos de explotación es su **autor** (ver artículo 97 de la LPI). No obstante, algunos o todos los derechos de explotación sobre una obra pueden ser cedidos por su autor y una tercera persona (física o jurídica) deviene titular derivado de los derechos de explotación de manera exclusiva o no.

Es importante tener presente que la cesión de derechos, en los sistemas continentales, no afecta a los derechos morales, que como hemos visto son indisponibles. No obstante, en los sistemas anglosajones, los autores de software no se benefician de derechos morales –y todos los derechos sobre un programa pueden cederse.

No vamos a estudiar aquí las particularidades de la cesión de derechos de explotación, pero es importante tener en cuenta que es el titular de cada derecho (y solamente él) quien puede otorgar una licencia sobre aquel derecho. Esto implica que una persona que sólo

**Nota**

La cesión de derechos de explotación se estudia en la unidad 5.

es titular de los derechos de distribución (por ejemplo, una tienda de software), no puede ceder derechos que no posea, como los de modificación o de difusión pública.

### **2.4.3. Identificar al titular**

Si, como hemos visto en el apartado anterior, sólo los titulares de los derechos de explotación de una obra, pueden otorgar licencias de uso sobre la misma, es obvia la importancia de conocer los métodos para poder identificar a los titulares.

Los símbolos ©, el registro de propiedad intelectual y las notas de reconocimiento de autoría incluidas en la misma obra son los instrumentos más usuales para identificar al titular de los derechos de autor de una obra, aunque estas formalidades no son más obligatorias desde el Convenio de Berna.

Además, el artículo 6 de la LPI establece unas reglas muy útiles en la práctica:

- a) Se presumirá autor a quien aparezca como tal en la obra mediante su nombre, firma o signo que le identifique.
- b) La cesión de derechos por parte del autor, como regla general, no se presume.
- c) Si la obra fue divulgada de manera anónima o con seudónimo, el ejercicio de los derechos corresponderá a quien la divulgue con el consentimiento del autor.

En el software libre, la identificación del autor puede resultar problemático, por ejemplo en el caso de que existan varios autores anónimos que hayan contribuido con código a una aplicación particular (Mozilla, GNOME, KDE, etc.). Esta dificultad de identificación del autor puede causar problemas, por ejemplo, a la hora de pedir su consentimiento a un cambio de licencia.

Para lidiar con este tipo de problemas, los coordinadores de ciertas aplicaciones libres mantienen listas de contribuyentes y han establecido listas de autores no identificados. Por ejemplo, la Free Software

Foundation exige que cada contribuyente se identifique al coordinador del proyecto, y los que envían más de 10 líneas de código deben ceder todos sus derechos a la Fundación. La empresa MySQL AB sólo acepta contribuciones a su motor de base de datos libre de los que hayan firmado su “Developers Agreement” (acuerdo de desarrollador).

Además, una exigencia común a todas las licencias de software libre es la **obligatoria mención de la autoría** en relación con la obra original. No obstante, cada licencia establece unos requisitos diferenciados. Así, la licencia Apache obliga a mantener en las obras derivadas la mención de la autoría del software original, la licencia GPL obliga a dar indicaciones de qué ha sido modificado y por quién, mientras que la licencia BSD no permite mencionar la autoría original si se realizan determinadas modificaciones.

#### Nota

Ver en la unidad 7 los distintos diferenciados de cada licencia.

## 2.5. Contenido de los derechos de autor

En este apartado vamos a estudiar el núcleo de toda la teoría de derechos de autor: los derechos que tiene el autor o titular de los derechos sobre una obra.



El artículo 2 de la LPI sintetiza el tema perfectamente:

“La propiedad intelectual está integrada por derechos de carácter personal y patrimonial, que atribuyen al autor la plena disposición y el derecho exclusivo a la explotación de la obra, sin más limitaciones que las establecidas en la ley.”



En el sistema continental se distinguen dos tipos de derechos: los **derechos de carácter personal o morales** (no reconocidos por el sistema anglosajón del *copyright* con relación a algunas obras, como el software) y los **derechos de carácter patrimonial o económico**.

**Nota**

La regulación de esta materia en Europa está pendiente de la transposición de la directiva de derechos de autor a la mayoría de países, incluido España, por lo que vamos a hacer referencia a las disposiciones de la directiva en casi todos los apartados.

**2.5.1. Derechos de carácter personal o derechos morales**

El sistema de derechos de autor continental sostiene la conexión íntima entre el autor y la obra y declara que el autor tiene un “derecho moral para limitar la alteración y la manifestación de su obra”, incluso después de que éste haya transferido sus derechos económicos a un tercero. Por el contrario, los sistemas de *copyright* de los países anglosajones no reconocen tradicionalmente los derechos morales a los autores sobre algunos tipos de obras, como por ejemplo el software.

El derecho moral del autor sobre su obra se compone de diferentes facultades o derechos que enumera el artículo 14 de la LPI (también es conveniente leer el artículo 113 para las bases de datos) y que, simplificando, sirven para “proteger su nombre y la integridad de la obra”, prohibiendo cualquier modificación sin el consentimiento previo del mismo.

Destacan los siguientes derechos que se conceden al creador:

- a decidir la divulgación de la obra;
- a determinar en qué forma debe hacerse la divulgación, con su nombre, bajo seudónimo o anónimamente;
- al reconocimiento de su nombre y al respeto de su estatus de autor;
- a exigir el respeto a la integridad de su trabajo e impedir cualquier alteración contra ella que suponga perjuicio a sus legítimos intereses o menoscabo a su reputación;

- a modificar la obra respetando los derechos adquiridos por terceros y las exigencias de protección de bienes de interés cultural;
- a retirar la obra del comercio, por un cambio en sus convicciones intelectuales o morales, previa indemnización de daños y perjuicios a los titulares de derechos de explotación (derecho de arrepentimiento), y
- a acceder al ejemplar único o raro de la obra, cuando se halle en poder de otro, a fin de ejercitar el derecho de divulgación o cualquier otro que le corresponda.

Estos derechos de carácter personal son **irrenunciables**, **intransmisibles** e **inalienables**, y algunos incluso **perpetuos**. Acompañan al autor durante toda su vida y algunos derechos también a sus herederos o causahabientes al fallecimiento de aquéllos.

Al regular los derechos de autor aplicados al software, la LPI y la Directiva de Programas de Ordenador de 1991 sólo mencionan los derechos de explotación y no mencionan los otros derechos patrimoniales del autor (de simple remuneración como comentaremos) ni los derechos morales.

No obstante, la doctrina mayoritaria entiende que los derechos morales sí que se aplican al software, aunque la tradición del *copyright* (iniciadora en la protección del software) no los reconozca. Así, se aplica el artículo 14 de la LPI a los programas de ordenador, con las únicas limitaciones establecidas en el artículo 100.4 de la LPI respecto a la integridad de la obra y la discusión doctrinal si los derechos morales se aplican a las personas jurídicas titulares de la obra en supuestos de obras colectivas.

El reconocimiento de los derechos morales que el sistema continental otorga a los autores de programas de ordenador puede resultar inconveniente para las compañías dedicadas a la producción de software (pensemos en el derecho al arrepentimiento). La doctrina defiende que una manera de evitarlos es privar a las personas físicas de la autoría de los programas.

Mientras que en el desarrollo de software propietario las empresas pueden adjudicarse todos los derechos (como empleadores o como editor/coordinador de una obra colectiva), en el desarrollo de software libre el

**Nota**

Ver los artículos 15, 16 y DT 6º de la LPI.

tema es más complejo. Para determinar la existencia y los titulares eventuales de derechos morales, deberá estudiarse el modelo de desarrollo en concreto y los acuerdos a los que han llegado los diferentes programadores que intervienen en la creación.

### 2.5.2. Derechos de carácter económico o patrimonial

Los derechos económicos o patrimoniales se basan en la convicción de que los autores deben ser compensados por sus habilidades creativas únicas.



Los **derechos económicos** consisten en el reconocimiento de ciertas facultades que permiten al titular obtener la utilidad pecuniaria que la obra reporte.

Estos **derechos**, que pertenecen en principio al autor, pueden ser **transferidos** o **cedidos** para permitir la explotación de una obra. A cambio de la transferencia de los derechos, el autor recibirá una remuneración, que generalmente será proporcional a los ingresos generados por la explotación de dicha obra.



Los **derechos patrimoniales** se dividen en derechos de explotación de la obra y en derechos meramente compensatorios. A continuación vamos a estudiarlos con más detenimiento.

### Derechos de explotación

Los derechos de explotación de la obra se subdividen, a su vez, en derechos de explotación exclusiva y derechos de simple remuneración:



Los **derechos de explotación exclusiva** son aquellos que la ley reconoce al autor o titular derivado de los



mismos y que confieren a su titular el poder jurídico de autorizar previamente o prohibir ciertas formas o actos de explotación respecto de su obra o prestación protegida, con la posibilidad de obtener una retribución por la autorización.



El artículo 17 LPI establece para las obras en general que:

“corresponde al autor el ejercicio exclusivo de los derechos de explotación de su obra en cualquier forma y en especial los derechos de reproducción, distribución, comunicación pública y transformación, que no podrán ser realizadas sin su autorización, salvo en los casos previstos en la Ley.”

El artículo 99 LPI establece con relación a los programas de ordenador:

“Sin perjuicio de lo dispuesto en el artículo 100 de esta Ley, los derechos exclusivos de explotación de un programa de ordenador, por parte de quien sea su titular con arreglo al artículo 97, incluirán el derecho de realizar o de autorizar:

- “a) La reproducción total o parcial, incluso para uso personal, de un programa de ordenador, por cualquier medio y bajo cualquier forma, ya fuere permanente o transitoria. Cuando la carga, presentación, ejecución, transmisión o almacenamiento de un programa necesiten tal reproducción, deberá disponerse de autorización para ello, que otorgará el titular del derecho.”
- “b) La traducción, adaptación, arreglo o cualquier otra transformación de un programa de ordenador y la reproducción de los resultados de tales actos, sin perjuicio de los derechos de la persona que transforme el programa de ordenador.”
- “c) Cualquier forma de distribución pública incluido el alquiler del programa de ordenador original o de sus copias.”

Como apreciamos de la comparación de los artículos 17 y 99 de la Ley de la propiedad intelectual, los derechos de explotación del autor de un programa de ordenador son básicamente los mismos que para cualquier otra obra, aunque deben adaptarse a sus características propias.

A continuación vamos a estudiar los medios habituales que la Ley de Propiedad Intelectual señala para que tal explotación pueda tener lugar, aunque la enumeración es ejemplificadora y no existe una lista cerrada o exhaustiva.

#### a) Derecho de reproducción



La **reproducción** consiste en fijar la obra en un medio que permita comunicarla y obtener copias. Es reproducción, incluso la efímera.

La legislación señala que autores o titulares de los derechos de explotación tienen un derecho exclusivo en relación con la reproducción de sus obras, ya sea directa o indirectamente, provisional o permanentemente, por cualquier medio o de cualquier manera, de la totalidad o parte de ésta.



Concretamente, el artículo 99 a) de la LPI reconoce al autor o al titular de los derechos de explotación el derecho exclusivo a realizar o autorizar la:

“reproducción total o parcial, incluso para uso personal, de un programa de ordenador, por cualquier medio y bajo cualquier forma, ya fuere permanente o transitoria. Cuando la carga, presentación, ejecución, transmisión o almacenamiento de un programa necesiten tal reproducción deberá disponerse de autorización para ello, que otorgará el titular del derecho.”

Aunque el tema está pendiente de la transposición de la Directiva de derechos de autor, en principio, también **precisarán autorización**, en cuanto constituyan reproducción, los siguientes actos:

- La **carga**: la introducción del programa en la memoria RAM.
- La **presentación**: la visualización en pantalla.
- La **ejecución**: la puesta en funcionamiento y utilización del programa.
- La **transmisión**: el envío a otro sistema mediante telecomunicación, red local, etc.
- El **almacenamiento**: la instalación en el disco duro o cualquier otro soporte.



El simple uso de un programa de ordenador requiere su copia en el equipo y, consecuentemente, su reproducción; por lo tanto, es necesaria la autorización del titular del derecho a la que hace referencia el artículo 99.

De esta manera se configura un derecho tal que permite al titular de los derechos de explotación de un software controlar el uso del programa por terceros aprovechando la circunstancia técnica de que apenas es posible hacer funcionar un ordenador mediante un programa sin que se produzcan copias en el ordenador.



El **derecho de copia/reproducción** (diferente a las copias de respaldo o de seguridad y a las copias privadas, permitidas las primeras y prohibidas las segundas, como veremos) es fundamental en las licencias, pues se debe disponer de él para poder ejecutar/utilizar el programa.

**Ejemplo**

Quien hurta un libro ajeno y lo lee, lesiona el derecho de propiedad del dueño del libro, pero no la propiedad intelectual del autor. Por el contrario, quien utiliza un software que no ha adquirido legítimamente viola los derechos de reproducción sobre el programa. ¿Por qué? Pues porque para servirse del programa y hacer con él aquello para lo que sirven los programas, necesariamente ha de reproducirlo (ejecutarlo), y el derecho de reproducción está reservado al titular de los derechos de explotación.

La cuestión de la reproducción es muy controvertida y difícil de resolver en relación con el software, debido a la naturaleza múltiple de un programa: se constituye de los elementos de diseño (su arquitectura y estructura), las líneas de código (que pueden ser código objeto o fuente) y el resultado de su ejecución. Una reproducción podría ocurrir a cualquier nivel o en relación con cualquier elemento.

La **reproducción literal** de las líneas de código sin autorización es la infracción más común en relación con los programas de ordenador. Las siguientes acciones son ejemplos de reproducción o copia literal:

- La instalación de un programa en un equipo.
- La grabación de un programa sobre un soporte físico (CD-ROM, disquete, disco duro).
- La “subida” de archivos a un servidor web o su descarga en un equipo local (por ejemplo, las transferencias en P2P).
- Adjuntar un programa a un correo electrónico y enviarlo.
- La integración o la incorporación de líneas de código de un tercero en un desarrollo propio.

Hay otras formas de reproducción, llamadas **no literales**, que también pueden constituir infracciones del derecho de copia o repro-

ducción. Los elementos no literales de un programa incluyen la estructura y la arquitectura de un programa, los formatos de entrada y salida de datos, el formato de las API, el interfaz gráfico, etc. La cuestión de delimitar si hay copia o no, en caso de dos programas similares en elementos no literales, es complicada.

Un supuesto de estudio interesante en relación con la copia no literal es el caso de dos programas diferentes que producen un resultado similar. Puede ocurrir, por ejemplo, por la reingeniería del primero (la creación de una nueva expresión de las ideas subyacentes) que es generalmente permitida por las leyes europeas siempre que el resultado de la reingeniería no sea sustancialmente similar al original.

Otro supuesto sería aquel caso en el que, aunque no haya ninguna copia literal de las líneas de código del original en el desarrollo del segundo programa, se puede argumentar que el segundo es una copia del primero porque las funcionalidades, la estructura, la organización de datos y/o el resultado de su proceso (por ejemplo, la interfaz gráfica) son demasiado similares.

Existe bastante jurisprudencia sobre la copia no-literal de programas de ordenador, sobre todo en Estados Unidos, aunque desafortunadamente por el momento no encontramos una respuesta definitiva y unánime a la cuestión. En el Reino Unido, por ejemplo, el test actualmente aceptado por los tribunales es que existe copia si hubo, en el desarrollo del segundo programa, “un uso excesivamente abusivo de las competencias, labores y criterios de los autores originales” referentes a la estructura de datos, la arquitectura de sistema, los métodos de desarrollo y de implementación, la interfaz gráfica, etc. Por lo tanto, para defenderse de la copia no autorizada, no es necesario probar la reproducción literal (una tarea normalmente difícil).

Estos supuestos son especialmente relevantes en el desarrollo de software libre dado que muchas de las aplicaciones libres imitan, y muchas veces mejoran, aplicaciones propietarias. Además, la prueba de la copia literal es más fácil con el software libre, ya que su código se distribuye de manera abierta.

## b) Derecho de distribución



Se entiende por **distribución pública** la puesta a disposición del público del original o copias de la obra mediante su venta, alquiler, préstamo o de cualquier otro modo.



El artículo 99 c) establece respecto a los programas de ordenador:

“El derecho a realizar o autorizar cualquier forma de distribución pública incluido el alquiler del programa de ordenador original o de sus copias.”

“A tales efectos, cuando se produzca cesión del derecho de uso de un programa de ordenador, se entenderá, salvo prueba en contrario, que dicha cesión tiene carácter no exclusivo e intransferible, presumiéndose, asimismo, que lo es para satisfacer únicamente las necesidades del usuario. La primera venta en la Unión Europea de una copia de un programa de ordenador por el titular de los derechos o con su consentimiento agotará el derecho de distribución de dicha copia, salvo el derecho de controlar el subsiguiente alquiler del programa o de una copia del mismo”.



No existe distribución sin ejemplares físicos, por lo que en el ciberespacio el concepto de distribución se desdibuja y, como veremos, el concepto de comunicación pública recobra relevancia.

Uno de los temas más complejos en la materia es el **agotamiento del derecho de distribución**: El derecho exclusivo de distribución del titular “se agota” con la “primera venta en la Unión Europea de una copia por el titular de los derechos o con su consentimiento”, salvo para los alquileres/arrendamientos del programa. Por tanto, si el ti-

**Nota**

Este derecho no está previsto en el Convenio de Berna y, sin embargo, a nivel Europeo ha sido armonizado para los programas de ordenador y las bases de datos.

tular de los derechos de explotación de un software “vendiera” copias, el comprador podría distribuirla libremente a terceros, por lo que el titular perdería los posibles ingresos por nuevas copias.

Ante esta posibilidad, los proveedores de software se cuidan de dejar muy claro en las licencias de uso que no “venden” ninguna copia al usuario, porque ello les haría perder su derecho exclusivo de controlar la distribución de las copias, y el usuario podría libremente distribuir su copia (si bien, no podría realizar copias sucesivas para distribuirlas), al menos en la Unión Europea.



Así pues, en la mayoría de licencias el titular/proveedor “vende” el soporte del software (el CD-ROM), pero respecto al software sólo concede al usuario el derecho a usarlo mediante una licencia, prohibiéndole que transmita su derecho de uso.

#### Nota

Este tema será estudiado con más detenimiento en la unidad 4.

Existen numerosas discusiones doctrinales sobre este tema, aquí sólo vamos a resumirlas:

- a) Al dilucidar si la venta de ejemplares de un programa agota el derecho de distribución sólo respecto de las ventas sucesivas (como dice el artículo 19. LPI para las obras protegidas en general) o lo agota respecto de cualquier acto posterior de distribución, excepto el arrendamiento/alquiler (como dice el artículo 99.c para los programas de ordenador).
- b) Es generalmente aceptado que la definición de alquiler comprende la “licencia” por tiempo determinado y el *leasing*. Queda la duda de si las licencias por tiempo indeterminado habrán de tratarse más bien como compraventa.



Se discute si el titular de los derechos de explotación puede impedir a ciertos usuarios “prestar” la copia del software. El préstamo se diferencia del alquiler en que, si bien en ambos casos el usuario cede temporalmente la copia a un tercero, en el préstamo se hace de forma gratuita.

Ocorre que la LPI contempla una excepción general (para obras literarias, artísticas, etc.) al derecho de distribución. El artículo 37.2 permite a determinadas instituciones culturales de interés general (museos, bibliotecas, filmotecas, instituciones docentes) realizar préstamos sin tener que recabar la autorización de los titulares de las obras ni satisfacerles remuneración alguna.

La doctrina entiende que, en puridad, esta excepción no rige para el caso del software. Las normas específicas de la LPI para el software se aplican preferentemente a las genéricas, y los artículos 99 y 100 contemplan el derecho exclusivo del proveedor del software a controlar cualquier tipo de distribución, sin recoger excepción alguna a favor del préstamo por parte de dichas instituciones. Por tanto, aunque creamos que tal prohibición es muchas veces excesiva, con la "LPI en mano", las bibliotecas o instituciones docentes no pueden prestar copias de software sin autorización expresa.

### c) Derecho de comunicación pública



El artículo 20 LPI para las obras en general establece:

"Se entenderá por comunicación pública todo acto por el cual una pluralidad de personas pueda tener acceso a la obra sin previa distribución de ejemplares a cada una de ellas. No se considerará pública la comunicación cuando se celebre dentro de un ámbito estrictamente doméstico que no esté integrado o conectado a una red de difusión de cualquier tipo."

El concepto de *comunicación pública* está originalmente concebido para el teatro, el cine o la radio tradicional, o sea, para un acto de difusión pública de una obra sin distribución de ejemplares físicos.



Ni la Directiva de Derechos de Autor sobre los Programas de Ordenador ni el artículo 99.c de la LPI mencionan el derecho de comunicación pública para los programas de ordenador. Aunque el hecho de que la enumeración de derechos del artículo 99 no sea cerrada, no permite excluir su aplicación al software.

De hecho, el artículo 3 de la Directiva de los Derechos de Autor (recordemos que está pendiente de implementar en España), sí que hace mención a este derecho y además lo amplía al crear una subcategoría dentro del mismo: el **derecho de poner a disposición** (o *making available*, en inglés), pensado para la distribución de obras por Internet. La Directiva, admitiendo las complicaciones jurídicas que la distribución de obras por entornos interactivos e Internet provocaba (no poder ser considerados distribución al no tratarse de distribución ejemplares físicos ni existir acción de comunicación), pretende clarificar la situación y reconoce expresamente a los autores el derecho exclusivo de comunicación dentro y fuera de estos entornos.



El artículo 3 de la Directiva de los Derechos de Autor en la Sociedad de la Información menciona un “derecho de comunicación al público de obras y derecho de poner a disposición del público prestaciones protegidas” y en su apartado primero establece:

“Los Estados miembros establecerán en favor de los autores el derecho exclusivo a autorizar o prohibir cualquier comunicación al público de sus obras, por procedimientos alámbricos o inalámbricos, incluida la puesta a disposición del público de sus obras de tal forma que cualquier persona pueda acceder a ellas desde el lugar y en el momento que elija.”

Por lo tanto, la autorización del autor también será necesaria para poner al alcance del público obras o prestaciones a través de Internet (en abierto o bajo la fórmula “a la carta”). Concretamente, estamos hablando de Internet y del mercado de servicios “a la carta” (cuando es el usuario quien escoge el momento y el lugar

de acceso) que es una de las principales áreas de crecimiento en la actualidad. Estos servicios se caracterizan justamente por el hecho de que el material almacenado en formato digital (textos, películas, música, software, o bases de datos) es puesto a disposición del público de tal forma que puede obtenerse una transmisión o acceso individualizado en el tiempo y el espacio.

**Ejemplo**

Un ejemplo práctico de violación de este derecho es colgar un programa en Internet en un sitio determinado (por ejemplo, warez) o ponerlo a disposición de terceros desde el propio ordenador a través de programas como e-donkey, e-mule, Kazaa, etc, o para el software libre, servidores de distribución como Source-Forge. Los titulares de los derechos de explotación tendrán, por lo tanto, un derecho de puesta a disposición y será necesaria su autorización para “colgar” un programa en Internet.

Es necesario hacer dos precisiones:

- a) Este derecho no cubre las comunicaciones privadas, por lo que no se considera comunicación pública ni puesta a disposición cuando se celebre dentro de un ámbito estrictamente doméstico que no esté integrado o conectado a una red de difusión de cualquier tipo, y
- b) No existe agotamiento (artículo 3.3 de la Directiva).

**Nota**

El problema es cómo separar en Internet lo que son comunicaciones públicas de privadas. Ejemplo: ¿existe comunicación privada si enviamos un documento a 50 amigos? ¿Y si utilizamos programas *peer to peer* y abrimos nuestro ordenador a la red?

#### d) Derecho de transformación



El artículo 99 b) de la LPI dispone que entre los derechos exclusivos de explotación sobre el software se encuentra el derecho exclusivo de realizar o de autorizar:

“la traducción, adaptación, arreglo o cualquier otra transformación de un programa de ordenador y la reproducción de los resultados de tales actos, sin perjuicio de los derechos de la persona que transforme el programa de ordenador.”

La transformación de una obra comprende su traducción, adaptación y cualquier otra modificación en su forma de la que se derive una obra diferente.



Los derechos de propiedad intelectual de la obra resultado de la transformación corresponden al autor de esta última, sin perjuicio del derecho del autor de la obra preexistente de autorizar, durante todo el plazo de protección de sus derechos sobre la obra original, la explotación de esos resultados en cualquier forma y en especial mediante su reproducción, distribución, comunicación pública o nueva transformación.

Aquella obra que incorpora una obra preexistente pero sin autorización (expresa o implícita a través de la colaboración del autor de la obra preexistente) producirá una infracción de los derechos de autor del titular de la obra original.

La mera existencia del reconocimiento del derecho exclusivo de transformación sobre la obra al titular del software podría ser suficiente para impedir las modificaciones de la obra sin el consentimiento del autor también con referencia a los programas de ordenador, si no fuera por los límites específicos que este derecho encuentra en el artículo 100 de la LPI.



Las “versiones sucesivas” de un programa de ordenador no constituyen obras derivadas, distintas, por tanto, de la originaria.

#### Nota

El termino **versión sucesiva** o *release* responde a la característica evolutiva del software dada la progresividad de mejoras en los programas de ordenador cada cierto tiempo por parte del titular de los derechos de explotación para mantenerlos en el mercado.



En concreto, el artículo 100.4 de la LPI señala que:

“El autor, salvo pacto en contrario, no podrá oponerse a que el cesionario titular de derechos de explotación realice o autorice la realización de versiones sucesivas de su programa ni de programas derivados del mismo.”

La doctrina interpreta que esta disposición no se aplica a los usuarios de una licencia de software, sino a aquéllos a quienes el autor haya cedido todos sus derechos de explotación sobre el software **en exclusiva** (titular derivado, por ejemplo un cliente). Y, aunque este precepto fuera aplicable también a las licencias de software (a los usuarios legítimos de un software), únicamente supondría que la prohibición de modificar el software se ha de establecer **expresamente** en la licencia (“pacto en contrario”), lo que ya sucede en la práctica en las licencias de software propietario.

La existencia de este derecho exclusivo de transformar el programa a favor del titular de los derechos de explotación (y, consecuentemente, las habituales prohibiciones de modificación que al usuario se le imponen en las licencias de uso), se encuentra estrechamente vinculada con lo siguiente:

- a) Ante todo, con un **derecho moral**: exigir el respeto a la integridad de la obra, en este caso el software. En virtud de tal dere-

cho que, al ser “moral”, en los sistemas continentales no es renunciable ni transmisible por el autor, éste puede impedir “cualquier deformación, modificación, alteración o atentado contra ella que suponga perjuicio a sus legítimos intereses o menoscabo a su reputación”.

De este modo, debemos entender que el autor del software nunca consiente que los usuarios desarrollen un software derivado que empeore las prestaciones técnicas del original hasta hacerlo desmerecer en la reputación que tenga en el mercado; y menos aún que por ejemplo, le agregue un “virus” informático.

- b) Si, de alguna manera, el titular de los derechos de explotación del software permite al usuario realizar modificaciones en el mismo (siempre que no perjudiquen a sus legítimos intereses o menoscaben su reputación), éste podrá pasar a ser autor de una **obra derivada**, con derechos análogos al autor/proveedor de software original. En tal caso, la exclusividad originaria del proveedor del software se verá perjudicada, lo que éste evita excluyendo el derecho de modificación.



El artículo 96.3 de la LPI establece que los derechos de autor sobre el software se extienden:

“a cualesquiera versiones sucesivas del programa, así como a los programas derivados, salvo aquellas creadas con el fin de ocasionar efectos nocivos a un sistema informático.”

Como hemos explicado en el apartado de obras originales y obras derivadas, el modelo usual de desarrollo de software libre implica muchas veces la transformación de obras preexistentes.



Por lo tanto, las licencias de software libre consideran la cesión de este derecho de transformación como una de las bases de la libertad incorporada en ellas. Es decir, si una licencia no otorga ningún derecho de transformación, no es una licencia libre.

#### Nota

Ver el apartado 4.1 de obras originales y obras derivadas.

Una vez finalizado el estudio de los derechos de explotación exclusiva, vamos a realizar una breve referencia al segundo componente de los derechos de explotación: los derechos de simple remuneración. Los **derechos de simple remuneración**, también conocidos bajo la denominación de **licencias obligatorias**, son aquellos que la ley concede a determinados titulares, en virtud de los cuales éstos pueden exigir a la persona que explota su obra o prestación protegida el pago de una suma de dinero, bien determinada en la ley (licencia legal obligatoria), o fijada por cualquier otro procedimiento (negociación a nivel sectorial, tarifas generales de la entidad de gestión, etc.).

### Derechos meramente compensatorios

Los derechos meramente compensatorios son básicamente el derecho de participación en el precio de la reventa de obras plásticas y el derecho de remuneración por copia privada. El derecho de remuneración por copia privada (art. 25 LPI) es un derecho irrenunciable a favor de autores y artistas, intérpretes o ejecutantes que compensan los derechos de propiedad intelectual dejados de percibir por razón de las reproducciones de las obras o prestaciones protegidas para uso exclusivamente privado del copista.



En el software está expresamente prohibida la copia privada, a excepción de las copias de respaldo y de ciertos actos de reproducción, y por lo tanto no existe derecho de remuneración por copia privada. Luego, sólo es relevante conocer el derecho de remuneración por copia privada por las consecuencias que va a tener la aplicación de un canon a los CD vírgenes para los usuarios de programas de ordenador.

Como sabemos, en España el canon se ha impuesto para retribuir las copias privadas que los particulares puedan realizar de música. Pero esos mismos CD se utilizan para realizar copias de seguridad o de respaldo de programas de ordenador, por lo que se ha argumentado que esta solución es injusta. Además, en algunos países europeos también se imponen cánones a otros dispositivos más relacionados con el software, como los discos duros o los módems, y ésta puede ser la tendencia futura en España.

## 2.6. Límites a los derechos de autor

Como hemos podido observar, el autor o titular de los derechos de explotación tiene un control casi absoluto sobre la obra. No obstante, el interés general impone unos límites temporales y sustantivos a este monopolio del autor o titular de los derechos de explotación que han tenido reconocimiento en la legislación sobre los derechos de autor. A continuación, vamos a estudiar las excepciones o limitaciones que la Ley reconoce a los derechos estudiados en el apartado anterior.

### 2.6.1. Límites temporales: la duración



La propiedad es eterna, no obstante la propiedad intelectual está sujeta a plazo. En toda la Unión Europea rige la regla general de que a los **70 años de la muerte** del autor o declaración de fallecimiento, la obra pasa a ser de **dominio público**. Las obras de dominio público pueden ser utilizadas por cualquiera, siempre que se respete la autoría y la integridad de la obra.



Dado que hay obras en que aún podría aplicarse por las disposiciones transitorias, es relevante tener en cuenta que, antes de la armonización europea, en España, la duración era de 80 años y en Gran Bretaña, de 50 años. La duración de los derechos de autor en Estados Unidos es compleja, pues depende de diferentes factores, incluyendo si ha sido publicada o no. La regla general para las obras creadas después del 1 de enero de 1978 está protegida por derechos de propiedad intelectual durante la vida del autor, más 70 años adicionales. Para las obras anónimas, con seudónimo o hechas por encargo, los derechos de *copyright* tendrán una duración de 95 años desde su primera publicación o 120 años desde su creación, cualquiera que sea el plazo que expire primero. Para los programas de ordenador, la duración de los derechos de autor en Estados Unidos es de 50 años después de acabada la vida del autor o de 75 desde la publicación en obras realizadas por encargo.

No obstante la regla general de los 70 años, existen plazos especiales para determinados tipos de obras:

- a) Los derechos de explotación de obras anónimas o seudónimas duran 70 años desde su divulgación lícita. Sin embargo, se aplica la regla general cuando, antes de cumplirse este plazo, sea conocido el autor, bien porque el seudónimo adoptado no deje dudas sobre su identidad, bien porque el mismo autor la revele.
- b) Los derechos de explotación de las obras no divulgadas lícitamente duran 70 años desde la creación de éstas, cuando el plazo de protección no sea computado a partir de la muerte del autor.
- c) Los derechos de explotación de las obras en colaboración duran toda la vida de los coautores y 70 años desde la muerte del último coautor superviviente.
- d) Los derechos de explotación sobre las obras colectivas duran 70 años desde la divulgación lícita de la obra protegida.

Además, debe tenerse en cuenta que los derechos conexos tienen plazos más breves y que algunos derechos morales, como la paternidad y la integridad, son perpetuos.

### 2.6.2. Límites sustantivos: las excepciones

Como hemos visto, los derechos de explotación no son absolutos, ya que con carácter excepcional, la ley enumera situaciones en las que se permite realizar actos que entran en la esfera de exclusividad del autor prescindiendo de la **autorización, e incluso, a veces, de la compensación**.

El objetivo de las excepciones en la mayoría de obras es asegurar intereses generales como son la educación, el acceso a la cultura, la libertad de información y de crítica y la libre competencia.



En los programas de ordenador, de manera muy novedosa en el sistema de derechos de autor, el objetivo es asegurar que el usuario legítimo (el que ha adquirido

#### Nota

Ver artículos 31 a 40 de la LPI para obras en general y artículo 100 LPI para programas de ordenador.



válidamente la licencia) de un software pueda utilizarlo de acuerdo a su destino.

En relación con el software, estas excepciones se conceptúan como una protección mínima a favor del usuario legítimo del programa, permitiéndole realizar ciertas acciones con el software sin que el titular de los derechos de explotación pueda prohibírselas. De lo contrario, se entendería que el derecho de uso queda desnaturalizado y no corresponde con lo que legítimamente puede esperar el usuario.



**En las licencias de software propietario**, el titular de los derechos de explotación puede controlar prácticamente el uso que se haga del programa. Las excepciones juegan por lo tanto un papel fundamental, aunque la exclusión absoluta de la utilización del programa no tiene excepción alguna respecto de quien no sea el “usuario legítimo” del programa. En cambio, en las **licencias de software libre**, la configuración de los derechos de explotación es tan amplia que rompe el monopolio del autor y hace muchas veces irrelevantes estas excepciones.

Las **excepciones a los derechos de autor** aplicables al software se desarrollan en el artículo 100 de la LPI y se pueden resumir como la autorización al usuario legítimo para:

- 1) La reproducción y transformación del programa cuando sea necesario para el uso del mismo, incluyendo la corrección de errores.
- 2) La realización de copias de seguridad o respaldo.
- 3) La realización de actos de ingeniería inversa o descompilación que permitan el análisis del programa y aseguren su interoperabilidad.



El sistema de derechos de autor ha optado por establecer un sistema cerrado con una enumeración concreta de excepciones, en cambio, el sistema del *copyright* contiene una cláusula abierta de **uso leal** o *fair use*, que deja cierto margen a los jueces para cada caso concreto. Este tema será explicado en adelante en el apartado dedicado al *copyright*.

Vamos a estudiar estas excepciones con un poco más de detenimiento.

### 1) El uso del programa de ordenador

La excepción de “uso del programa de ordenador” deroga la aplicación del artículo 99.a del TRPLI sobre la reproducción y el 99.b sobre la transformación, y establece que el usuario legítimo de un programa de ordenador no necesita autorización del proveedor para reproducir (cargar, ejecutar, transmitir y almacenar) y transformar el software cuando sea necesario para la utilización del mismo, con arreglo a su finalidad propuesta.



Art. 100.1 LPI:

“No necesitarán autorización del titular, salvo disposición contractual en contrario, la reproducción o transformación de un programa de ordenador incluida la corrección de errores, cuando dichos actos sean necesarios para la utilización del mismo por parte del usuario legítimo, con arreglo a la finalidad propuesta.”

Resulta obvio: si el derecho de uso sobre el software es la facultad esencial en toda licencia, el usuario debe tener asegurada la posibilidad de utilizar el software una vez acepte la licencia, durante la vigencia de la misma y mientras cumpla sus términos y condiciones.

Se establecen tres condiciones para que la excepción sea aplicable:

- 1) Esta derogación ampara actos de reproducción (ejecución, carga, presentación en pantalla, transmisión o almacenamiento) y transformación efectuados para la utilización del programa de ordenador, incluyendo explícitamente la corrección de errores.
- 2) Sólo el “usuario legítimo” se beneficia de esta excepción, o sea el que ha obtenido el software mediante una licencia del titular.
- 3) El uso del programa debe adecuarse a la finalidad propuesta.



Aunque la LPI prevé que esta facultad del usuario no necesita de autorización “salvo disposición contractual en contrario”, no es lógico ni racional pensar que una licencia puede obligar al usuario a pedir al proveedor autorizaciones diferentes y sucesivas para cada acto necesario para usar el software, y para cada momento en que desee usarlo. Por tanto, debemos entender que mientras esté vigente la licencia, el usuario puede reproducir el software (cargar, ejecutarlo y almacenar) siempre que quiera.

Sin embargo, éste no sólo es un supuesto excepcional, sino que además en la práctica el usuario tiene muy difícil beneficiarse del consiguiente derecho a modificar el software, porque el titular de los derechos de explotación del software no le entrega (ni tiene la obligación, en principio, de entregarle) una versión del programa en código fuente.

## 2) La copia de seguridad

El usuario del software puede hacer una copia de seguridad o respaldo sin necesidad de autorización del titular, siempre que le sea necesaria para la utilización del software.



En concreto, el artículo 100.2 de la LPI dispone que:

“La realización de una copia de seguridad por parte de quien tiene derecho a utilizar el programa no podrá

impedirse por contrato en cuanto resulte necesaria para dicha utilización.”

El reconocimiento de esta excepción implica que, para los demás supuestos, la copia privada es considerada como ilícita en materia de programas de ordenador. Con ello, se introduce una excepción a la regla general contenida en el artículo 31 de la LPI que permite las reproducciones de las obras en general sin autorización del autor para uso personal del copista.

Evidentemente, esta copia de seguridad no puede destinarse a otro usuario y sólo puede utilizarse cuando la copia original se haya deteriorado, destruido o eliminado.

Seguramente producto de las presiones recibidas por la industria del software propietario, la LPI limita el derecho a hacer una copia de seguridad cuando “resulte necesaria para la utilización del programa”. De ello se deriva que el usuario puede hacer una copia de seguridad cuando la copia original se le ha proporcionado directamente en el propio elemento hardware (por ejemplo, software preinstalado) o, en todo caso, cuando el soporte en el que se le ha proporcionado es necesario para utilizar el software (por ejemplo, el software viene en un CD-ROM que ha de estar en el lector del ordenador cuando el programa se ejecuta).

Por el contrario, si el soporte en que se proporciona el software no es necesario para utilizar el software una vez instalado, el proveedor puede prohibir al usuario realizar una copia de seguridad (atribuyendo al soporte original la condición de “copia de seguridad”).



En el caso de obras literarias o artísticas (libros, música, etc.), la LPI permite al adquirente de los derechos de explotación o usuario realizar copias privadas, siempre que no sea objeto de utilización colectiva ni lucrativa (por ejemplo, quien compra un disco de música puede válidamente hacerse una copia de la obra para usarla

en su reproductor del coche, *walkman*, etc.). Sin embargo, la LPI no concede este derecho a realizar una copia privada a los usuarios de software, salvo para tener una copia de seguridad, o la propia copia en el dispositivo hardware de los archivos que sean necesarios para la ejecución del programa.

Esto se debe, con toda seguridad, a la presión de las grandes empresas desarrolladoras de software en el momento de redactar y promulgarse la directiva europea; en concreto, al deseo de obtener la máxima protección a su derecho exclusivo de poner en circulación copias del software (derecho de distribución), la base de su modelo de negocio. Por tanto, el usuario ni siquiera puede hacer copias del software licenciado, cuando pretenda destinarlas para sí mismo y no para distribuirlas a terceros.

### 3) Los actos de ingeniería inversa

La ingeniería inversa puede definirse como todas aquellas operaciones que son necesarias para que se pueda tomar conocimiento de las ideas que subyacen en el programa de ordenador.

El establecimiento de una autorización para realizar actos de ingeniería inversa pretende asegurar el desarrollo de una industria no monopolística y la interoperabilidad, y al mismo tiempo pretende luchar contra la ingeniería inversa que intenta descubrir el código fuente a través del código objeto.

En el ordenamiento jurídico español existen tres disposiciones que tratan de este tema, los artículos 100.3, 100.5 y 100.6 de la LPI. Vamos a estudiarlos a continuación.

#### a) Un derecho al análisis

El artículo 100.3 LPI autoriza al usuario legítimo de un programa de ordenador para **observar, estudiar o verificar su funcionamiento**, durante el uso normal del mismo, con el fin de determinar las ideas y principios implícitos en cualquier elemento del programa.



Concretamente, el artículo 100.3 de la LPI establece:

“El usuario legítimo de la copia de un programa estará facultado para observar, estudiar o verificar su funcionamiento, sin autorización previa del titular, con el fin de determinar las ideas y principios implícitos en cualquier elemento del programa, siempre que lo haga durante cualquiera de las operaciones de carga, visualización, ejecución, transmisión o almacenamiento del programa que tiene derecho a hacer.”



Este derecho a observar y estudiar el funcionamiento del programa es inherente al propio derecho de uso. Además, es consecuencia lógica de lo previsto por el artículo 96.4 de la LPI: el titular del software no tiene protegidos por los derechos de autor “las ideas y principios en los que se basan cualquiera de los elementos de un programa de ordenador, incluidos los que sirven de fundamento a sus interfaces”.

No obstante, con el mero estudio del software durante su ejecución, muchas veces le resultará al usuario imposible acceder en la práctica a dichas ideas y principios subyacentes al programa. Necesitaría para ello el código fuente (que el proveedor de software propietario no entrega ni tiene obligación de entregar), o el derecho a descompilar el programa, que la LPI le permite en el supuesto puntual de obtener la interoperabilidad con otro programa, tal y como veremos a continuación.

#### b) La descompilación

La descompilación consiste en obtener, a partir de los archivos ejecutables en código binario que componen el programa (inteligibles solamente para el equipo hardware), una versión aproximada del código fuente (comprensible para usuario, si dispone de los conocimientos de programación suficientes), que le permite modificar y adaptar el software.

Como excepción al derecho exclusivo de modificación del titular de los derechos de explotación del software, el usuario no necesita la autorización del titular para la “reproducción del código y la traducción de su forma” si ello es indispensable para “obtener la información necesaria para la interoperabilidad de un programa creado de forma independiente con otros programas”. De esta manera, se permite la descompilación si tiene como objetivo posibilitar que un programa funcione en conjunción con otros.

Actualmente la descompilación es aceptada explícitamente por nuestra legislación, subordinada al cumplimiento de tres condiciones:

- Que sólo el **usuario legítimo** o cualquier persona facultada para utilizar un programa de ordenador, o en su nombre, una persona debidamente autorizada (por ejemplo, la empresa), puede llevar a cabo los actos de descompilación.
- Que el titular de los derechos de explotación no haya puesto una **versión del código fuente** (al menos, de la parte que puede permitir la interoperabilidad) **a disposición del usuario**, es decir, que la información necesaria no debe haber sido puesta a disposición de personas legitimadas de manera rápida y fácil.
- Que la descompilación debe limitarse a aquellas **partes del programa** que resultan **necesarias** para conseguir la interoperabilidad.



Cuando la LPI permite la descompilación, presupone que ésta debe ser indispensable, significa que únicamente debe recurrirse a esta técnica si la información necesaria para la interoperabilidad no se puede obtener estudiando las “ideas y principios” del software mientras éste se ejecuta.

Una vez obtenida lícitamente la información, se establecen **restricciones** al uso de la información obtenida:

- La información no puede destinarse a **propósitos distintos** a los de adquirir interoperabilidad para un programa de ordenador

creado independientemente. Así, sólo se podrá utilizar la información relativa a las ideas y principios que son la base de las interfaces, y sólo puede emplearse para desarrollar un programa que tenga la posibilidad de ser conectado y empleado con cualesquiera elementos de un sistema informático.

- La información no puede ser comunicada a terceros, excepto si fuera necesario para la consecución de la interoperabilidad del programa creado de forma independiente. Así, la persona que ha descompilado un programa tiene una obligación legal de **confidencialidad**.
- La información obtenida no podrá ser utilizada para el desarrollo, producción o comercialización de un **programa sustancialmente similar** en su expresión o para cualquier otro acto que infrinja los derechos de autor. No se puede copiar la forma de expresión de las interfaces del programa descompilado, ya que ellos pueden estar protegidos por sí mismos siempre que cumplan la condición de originalidad.

Finalmente, existe una **limitación general** que señala que las disposiciones relativas a la descompilación no pueden interpretarse de manera que su aplicación cause “un perjuicio injustificado a los intereses legítimos del titular de los derechos o sea contraria a la explotación normal del programa de ordenador”.

Esta limitación, aparentemente teórica, tiene cierta aplicación práctica: cuando las operaciones de descompilación para obtener la interoperabilidad con otros programas puedan perjudicar los derechos y expectativas del titular de los derechos de explotación (comercializar en exclusiva el software, realizar versiones nuevas, etc.), el usuario debe también abstenerse de realizarlas, por lo que en muchas ocasiones este derecho del usuario será impracticable.

#### Nota

Cualquiera puede analizar una obra literaria o cinematográfica e intentar establecer sus principios o ideas. A eso se dedican los críticos o los especialistas en provecho de otros escritores. En cambio, en el ámbito del



software hallamos una gran resistencia en el establecimiento de condiciones para llevar a cabo la operación (art. 100.5 LPI: legitimados, inexistencia de alternativas, limitación del objeto de estudio, etc.) y en las aplicaciones que se puede dar a la información obtenida de ese modo (artículo 100.6 LPI: sólo para garantizar la interoperabilidad y no para competir con un programa igual). ¿Quizás porque en el software lo esencial es la invención o la idea?

Consideramos importante destacar varios **actos secundarios**, en el sentido de que no son infracciones directas de derechos exclusivos, sino que atentan indirectamente contra ellos, asociados con el software, que pueden constituir violaciones de la propiedad intelectual:

- La fabricación, comercialización o distribución de artículos destinados a la copia ilegal de obras protegidas (*rippers*).
- Facilitar una infracción por la provisión de servicios de transmisión, difusión o distribución de obras ilegales (por ejemplo, empresas que basan su negocio en sistemas P2P).
- La elusión de medidas tecnológicas de protección: es un tema importante, que vemos a continuación en el siguiente apartado.

### 2.6.3. Sumario de conocimientos



“Los derechos de autor están integrados por derechos morales y de explotación, que atribuyen al autor la plena disposición y el derecho exclusivo a la explotación de la obra, sin más limitaciones que las establecidas en la ley.”

Derechos de explotación del titular:

- Derecho de reproducción (copia)
- Derecho de distribución pública de ejemplares físicos

- Derecho de comunicación pública y de poner a disposición
- Derecho de transformación

Límites a los anteriores derechos:

- La duración: 70 años
- Reproducción y transformación para el uso del programa
- Copias de seguridad o respaldo
- Actos de ingeniería inversa o descompilación para asegurar la interoperabilidad

## 2.7. Medidas de protección frente a la infracción de derechos

Conocemos ya que la legislación sobre derechos de autor reconoce al autor del software, o titular de sus derechos de explotación, una serie de importantes derechos en exclusiva: derecho a usar, modificar, y distribuir el software. Él es quien decide a quién, cuándo y cómo autoriza a utilizar el software. De este modo, cualquier utilización del software por alguien que no sea su titular, sin que éste lo haya permitido expresamente, es un uso prohibido por la propia ley.

En este apartado vamos a estudiar las medidas legales de protección de las obras y las reacciones ante posibles infracciones de derechos de autor. Para clarificar el tema, vamos a dividirlos en medidas preventivas, y éstas en legales y tecnológicas, y en medidas reactivas o soluciones a infracciones de derecho de autor. Asimismo, realizaremos una breve referencia a la legislación penal y administrativa.

### 2.7.1. Medidas de protección

#### Medidas legales de protección



Las medidas legales de protección son los mecanismos que la legislación reconoce para proteger los derechos de los titulares de derechos de autor.

Como hemos estudiado, en los países parte en el Convenio de Berna y el Acuerdo ADPIC, la protección por derecho de autor no requiere ninguna formalidad. Se considera que, por su mera existencia, toda obra creada queda protegida por el derecho de autor. No obstante, existen formalidades o mecanismos que aunque no son obligatorios, sí que son muchas veces convenientes para prevenir infracciones de derechos de autor o ser utilizadas posteriormente como prueba de la autoría.

#### a) El registro de la propiedad intelectual

No obstante no ser obligatorio el registro de la obra, en la mayoría de los países es un procedimiento muy directo y económico y que confiere beneficios adicionales importantes en algunos países. En España, por ejemplo, el registro genera una presunción legal, según la cual quien figura como autor del programa inscrito lo es en realidad.



Algunos estados, como Estados Unidos, pueden requerir que sus propios ciudadanos, pero no los autores extranjeros, registren sus obras en la Oficina de Derechos de Autor del país. Por ejemplo, para presentar una demanda en los tribunales federales. Además, en algunos países, un registro de derechos de autor constituye una prueba *prima facie* de la validez y propiedad de derechos de autor.

#### b) El depósito notarial

Como medida alternativa o complementaria a la inscripción en el Registro de la Propiedad Intelectual, a veces es aconsejable el depósito notarial del código fuente completo, de soportes magnéticos conteniendo el código objeto, del manual de uso, diseño de las pantallas, análisis, documentación preparatoria y demás elementos identificativos del programa que pueden ser de vital importancia para la práctica de una prueba pericial.

El depósito notarial puede ser una prueba fehaciente en juicio sobre autoría. Asimismo, se ha revelado como un sistema más adecuado

que el registro por su mayor confidencialidad y rapidez, menor burocracia y mejor descripción del programa. No obstante, la aceptación del depósito es voluntaria para los notarios, por lo que es mejor introducir el programa en una maleta de aluminio o de plástico que el notario precintará. El interesado se convierte entonces en depositario del código fuente.

### c) Uso del símbolo © o indicación de reserva de derechos

Por aplicación de la Convención de Berna, en la mayoría de los países, no se requiere ya un aviso de derechos de autor para que la obra esté protegida por los derechos de autor. No obstante, es común la utilización de símbolos o avisos para advertir a los usuarios que la información está protegida. La manera correcta de elaborar dicho aviso es: “[Derechos de Autor [fecha] por [autor/dueño]”. A veces se usa el símbolo © en referencia al sistema del *copyright*.

## Medidas tecnológicas de protección

Cuando se concibió la figura de los derechos de autor como método de protección de las obras, e incluso actualmente en el mundo de la distribución en formato papel, las editoriales y los derechos de autor tradicionales tenían su función porque un autor necesitaba de su infraestructura para difundir sus obras y la copia era casi inexistente o limitada al ámbito privado del copista.

A lo largo de los años, la evolución tecnológica fue ampliando este panorama y aparecieron otro tipo de obras, lenguajes y medios de explotación a los que fue necesario que los derechos de autor se adaptaran. Sin embargo, la sociedad de la información y las nuevas tecnologías han producido un cambio radical al poner al alcance de todos, en un nuevo medio, las obras tradicionales (texto, música, fotografía, etc.) y al capacitar a cualquier persona, sobre todo gracias a Internet, para ser editor y distribuir una obra sin necesidad de intermediarios.

Además, esa misma revolución tecnológica implica que la capacidad de copia y de reproducción, con o sin ánimo de lucro, se ha generalizado. Hace unos cuantos años, una grabadora de CD estaba dis-

ponible únicamente para un estudio de grabación, mientras que hoy la tenemos casi todos en casa, en nuestro ordenador personal. Por no hablar del *peer-to-peer*. Para la mayoría de la población la violación de los derechos de autor no es percibida como una conducta anti-social e incluso podríamos decir que se ha convertido en una conducta habitual. Comprar un CD musical en un puesto callejero, bajarse música de Internet o instalar programas de ordenador sin pagar licencias son ejemplos de prácticas que pueden violar los derechos de propiedad intelectual de terceros.

Por ello, todo el sistema de distribución de las obras y los elementos, sobre la base de los cuales los autores y los intermediarios basan sus beneficios, ha quedado en entredicho. Esto ha supuesto un cambio irreversible, frente al cual toda la legislación sobre propiedad intelectual debe ser redefinida para devolver el equilibrio a las partes en conflicto. Los titulares de los derechos de explotación afectados (principalmente la industria de la música, el cine y el software) pretenden utilizar los derechos de autor no ya como arma de un empresario contra otro, como fueron concebidos en su origen, sino como defensa de un empresario/titular contra el público que vulnera los derechos de esa compañía, y para ello han confiado principalmente en la misma tecnología. Así, la protección de estas obras parece que, en un futuro muy cercano, estará confiada principalmente a las medidas tecnológicas de protección.

Asimismo, estas medidas tecnológicas son la base tecnológica para los nuevos **Sistemas de Gestión de Derechos de Propiedad Intelectual** (*Digital Rights Management Systems* o DRMS, en inglés) que revolucionarán las relaciones entre los usuarios y los titulares de los derechos de explotación sobre las obras musicales, audiovisuales y software, principalmente. Los DRMS son procesos tecnológicos de gestión de los derechos de autor que permiten un control casi total del titular sobre su obra. Los DRMS están empezando a ser utilizados para identificar las obras, a los titulares de las mismas, para solicitar consentimientos previos a los titulares por parte de usuarios o para realizar micro pagos por servicios “a la carta”, entre otras funcionalidades. Como veremos, el artículo 7 de la Directiva de Derechos de Autor hace expresa referencia a ellos.

Por supuesto, estos sistemas de gestión también tienen implicaciones para la comunidad del software libre, pues no sólo es contrario a su

propia filosofía, sino que puede llegar a criminalizar programas libres como potenciales mecanismos de elusión de estos sistemas. Un ejemplo de máxima actualidad sería un programa libre lector de DVD que podría permitir la captura de los datos originales de un DVD para su copia al disco duro. Sin embargo, no se pueden desestimar las nuevas tecnologías sólo porque algunas personas puedan utilizarlas de forma fraudulenta.

A continuación vamos a comentar las principales medidas tecnológicas de protección de las obras y su tratamiento jurídico.

#### a) Tecnologías de protección

Frente a las infracciones masivas de derechos de propiedad intelectual cometidas en el mundo digital, y la facilidad con la cual se puede copiar y distribuir obras sin autorización de los titulares de sus derechos, se han inventado varias tecnologías y métodos para proteger las obras, incluidos los programas de ordenador y objetos informáticos asociados (bases de datos, etc.).



Desde el punto de vista legal, se puede definir como medida de protección “cualquier dispositivo o medio destinado a prevenir o restringir la copia de una obra o a reducir la calidad de cualquier copia realizada”.

Algunos ejemplos que encontramos ya en el mercado son:

- Claves de acceso: muchas veces, para la instalación de un programa, se requiere insertar una clave de identificación única. Asimismo, para acceder a bases de datos, sobre todo en línea, se suele requerir insertar un nombre de usuario y una clave (*password*).
- Sistemas de identificación única: marcas de agua (o *watermarks*).
- El cifrado y la encriptación de obras.
- Sistemas anticopia: son medidas para impedir que se pueda copiar una obra digital. Por ejemplo, el sistema CSS para los DVD.

- Los sistemas que Adobe utiliza en sus e-book, que impiden imprimir o copiar.

Actualmente para hacer frente a la proliferación de tecnologías y grupos que pretenden evitar o romper estos dispositivos (DeCSS, etc.), las medidas tecnológicas de protección de los derechos de propiedad intelectual están siendo protegidas por la ley, como vemos a continuación.

#### b) Protección legal de las medidas tecnológicas

Como hemos visto, la tecnología supera y sustituye el derecho en su protección de las obras, pero esta tecnología para ser lícita y realmente útil también requiere de la protección y del reconocimiento del derecho.

La digitalización y el avance de las nuevas tecnologías no sólo aporta nuevos riesgos para los titulares de los derechos de propiedad intelectual, también hace más fácil administrar y controlar los actos de explotación por medio de sistemas de control de acceso, identificación y prevención de copias. La introducción a gran escala de estos sistemas de protección dependerá de la protección legal que exista contra los actos de neutralización o manipulación de los mismos.

Para hacer frente a esta situación, la Organización Mundial de la Propiedad Intelectual aprobó los tratados OMPI de 1996: el **Tratado OMPI de Derechos de Autor (TODA)** y el Tratado OMPI de Interpretaciones y Fonogramas en diciembre, que estudiamos al inicio de esta unidad.

El Tratado TODA traduce este pensamiento al requerir a los países que lo suscriban que modifiquen sus legislaciones en el sentido de proporcionar dos tipos de protección legal de las medidas tecnológicas para garantizar que los titulares de derechos puedan utilizar eficazmente las tecnologías para proteger sus derechos y conceder licencias sobre sus obras en línea:

- a) El primero de estos complementos tecnológicos, conocido como la disposición contra la elusión, aborda el problema de la “piratería” al exigir que los países suministren una protección jurídica adecuada y prevean recursos eficaces contra la elusión de las medidas tecnológicas utilizadas por los titulares para proteger sus derechos.

- b) El segundo vela por la fiabilidad e integridad del mercado virtual, al exigir que los países prohíban la alteración o supresión deliberada de información electrónica sobre gestión de derechos, es decir, la información que acompaña a cualquier material protegido y que permite identificar la obra, su autor, artista intérprete o ejecutante, o propietario y las condiciones de su uso.

Éste parece ser el futuro, pues esta corriente de pensamiento es la que influye a las últimas novedades legislativas en Estados Unidos y en Europa.



En concreto, la sección 1201 de la Digital Millenium Copyright Act (DMCA) de 1998 en Estados Unidos y, posteriormente, los artículos 6 y 7 de la Directiva Europea de Derechos de Autor en la Sociedad de la Información están optando por legitimar y otorgar protección legal a los sistemas DRMS mediante la prohibición de los dispositivos y prácticas que permiten deshabilitarlos.

Estados Unidos fue el primero en implementar el Tratado TODA, a través de la DMCA que incorpora nuevas prohibiciones para evitar la evasión de las medidas tecnológicas a fin de proteger las obras amparadas por derechos de *copyright* contra estas medidas tecnológicas.

A nivel europeo, también se ha implementado el Tratado TODA, concretamente en los artículos 6 y 7 de la Directiva de Derechos de Autor, que siguen en buena parte lo establecido por la DMCA.



#### Protección de medidas tecnológicas e información para la gestión de derechos

Artículo 6. Obligaciones relativas a medidas tecnológicas

“1. Los Estados miembros establecerán una protección jurídica adecuada contra la elusión de cualquier



medida tecnológica efectiva, cometida por una persona a sabiendas, o teniendo motivos razonables para saber que persigue ese objetivo.”

“2. Los Estados miembros establecerán una protección jurídica adecuada frente a la fabricación, importación, distribución, venta, alquiler, publicidad para la venta o el alquiler, o posesión con fines comerciales, de cualquier dispositivo, producto o componente o la prestación de servicios que:”

“a) sean objeto de una promoción, de una publicidad o de una comercialización con la finalidad de eludir la protección, o”

“b) sólo tengan una finalidad o uso comercial limitado al margen de la elusión de la protección, o”

“c) estén principalmente concebidos, producidos, adaptados o realizados con la finalidad de permitir o facilitar la elusión de la protección de cualquier medida tecnológica eficaz.”

“3. A efectos de la presente Directiva, se entenderá por ‘medidas tecnológicas’ toda técnica, dispositivo o componente que, en su funcionamiento normal, esté destinado a impedir o restringir actos referidos a obras o prestaciones protegidas que no cuenten con la autorización del titular de los derechos de autor o de los derechos afines a los derechos de autor establecidos por ley o el derecho *sui generis* previsto en el capítulo III de la Directiva 96/9/CE. Las medidas tecnológicas se considerarán ‘eficaces’ cuando el uso de la obra o prestación protegidas esté controlado por los titulares de los derechos mediante la aplicación de un control de acceso o un procedimiento de protección, por ejemplo, codificación, aleatorización u otra transformación de la obra o prestación o un mecanismo de control del copiado, que logre este objetivo de protección.”

“4.No obstante la protección jurídica prevista en el apartado 1, en caso de que los titulares de los derechos no adopten medidas voluntarias, incluidos los acuerdos con otros interesados, los Estados miembros tomarán las medidas pertinentes para que los titulares de los derechos faciliten al beneficiario de una excepción o limitación establecida por el derecho nacional de conformidad con las letras a), c), d), y e) del apartado 2 del artículo 5 o con las letras a), b) y e) del apartado 3 del mismo artículo, los medios adecuados para disfrutar de dicha excepción o limitación, en la medida necesaria para ese disfrute, siempre y cuando dicho beneficiario tenga legalmente acceso a la obra o prestación protegidas.”

“Un estado miembro podrá adoptar asimismo tales medidas respecto del beneficiario de una excepción o limitación establecida en virtud de la letra b) del apartado 2 del artículo 5, a menos que los titulares de los derechos hayan hecho ya posible la reproducción para uso privado en la medida necesaria para el disfrute de la excepción o limitación contemplada y de conformidad con lo dispuesto en la letra b) del apartado 2 del artículo 5 y en el apartado 5 del mismo artículo, sin impedir a los titulares de los derechos la adopción de medidas adecuadas respecto del número de reproducciones de conformidad con tales disposiciones.”

“Tanto las medidas tecnológicas adoptadas voluntariamente por los titulares de los derechos, incluidas las derivadas de acuerdos voluntarios, como las adoptadas en aplicación de medidas adoptadas por los Estados miembros, disfrutarán de la protección jurídica prevista en el apartado 1.”

Lo dispuesto en los párrafos primero y segundo no será de aplicación a obras o prestaciones que se hayan puesto a disposición del público con arreglo a lo convenido por contrato, de tal forma que personas concretas del público puedan acceder a ellas desde un lugar y en un momento que ella misma haya elegido.

Cuando el presente artículo se aplique en el contexto de las Directivas 92/100/CEE y 96/9/CE, el presente apartado se aplicará *mutatis mutandis*.

#### Artículo 7. Obligaciones relativas a la información para la gestión de derechos

“1. Los Estados miembros establecerán una protección jurídica adecuada frente a todas aquellas personas que a sabiendas lleven a cabo sin autorización cualquiera de los siguientes actos:”

“a) supresión o alteración de toda información para la gestión electrónica de derechos;”

“b) distribución, importación para distribución, emisión por radiodifusión, comunicación o puesta a disposición del público de obras o prestaciones protegidas de conformidad con lo dispuesto en la presente Directiva o en el capítulo III de la Directiva 96/9/CE en las que se haya suprimido o alterado sin autorización la información para la gestión electrónica de derechos, sabiendo, o teniendo motivos razonables para saber que al hacerlo inducen, permiten, facilitan o encubren una violación de los derechos de autor o los derechos afines a los derechos de autor establecidos por una norma jurídica, o el derecho *sui generis* previsto en el capítulo III de la Directiva 96/9/CE.”

“2. A efectos de la presente Directiva, se entenderá por ‘información para la gestión de derechos’ toda información facilitada por los titulares de los derechos que identifique la obra o prestación protegida contemplada en la presente Directiva, o a que se refiere el derecho *sui generis* previsto en el capítulo III de la Directiva 96/9/CE, al autor o cualquier otro derecho habiente, o información sobre las condiciones de utilización de la obra o prestación protegida, así como cualesquiera números o códigos que representen dicha información.”

El párrafo primero se aplicará cuando alguno de estos elementos de información vaya asociado a una copia de una obra o prestación o aparezca en conexión con la comunicación al público de una obra o prestación protegida prevista en la presente Directiva o cubierta por el derecho *sui generis* previsto en el capítulo III de la Directiva 96/9/CE.

La Directiva exige a los Estados miembros que creen un entorno legal adecuado para perseguir cualquier actividad de fabricación o distribución de sistemas que neutralicen o faciliten la inutilización de sistemas de protección sin la autorización del titular de los derechos. Al mismo tiempo, los Estados miembros dispensarán una protección legal adecuada que permita perseguir a cualquier persona que sin autorización retire o altere actos o información sobre la gestión electrónica de derechos, o distribuya, importe, difunda, comunique públicamente, o ponga a disposición del público copias de obras o cualquier otro elemento del cual se haya retirado la información relativa a la gestión de derechos.

Es un tema polémico, pues el resultado de la protección legal de estas medidas tecnológicas puede ser “abusivo” si los titulares de derechos de explotación pueden controlar completamente el uso de los contenidos y no sólo los derechos de propiedad intelectual. Además, las medidas tecnológicas protegidas por la ley no sólo pueden aumentar ilimitadamente la capacidad de control del titular de los derechos sobre la obra, sino que puede afectar a los derechos de los usuarios en lo que respecta a la capacidad de interoperabilidad de los programas de ordenador.

Quizás la mejor prueba de la relevancia del tema y de los debates que está provocando es la lentitud de implementación de la Directiva de Derechos de Autor en los diferentes países europeos.

### **2.7.2. Medidas de reacción frente la infracción de derechos**

Una vez que las medidas preventivas anteriores han resultado infructuosas, y la infracción de los derechos de autor sobre una obra se ha

producido, las que hemos denominado como “medidas de reacción” entran en juego.

Frente a estas violaciones de derechos de autor, la legislación, tanto española como de la mayoría de los estados firmantes de los tratados internacionales, ofrece una serie de mecanismos de protección de los derechos de propiedad intelectual, existiendo la posibilidad de acudir a acciones **administrativas, civiles y penales**.

En España, la LPI ha optado por una **tutela civil** de la propiedad intelectual, basada en la idea de reparación de un derecho privado, en lugar de articular una exclusiva tutela por el derecho penal, aunque sin perjuicio de cualquier otra acción que corresponda al titular.



El artículo 102 de la LPI regula la infracción de los derechos en programas de ordenador y establece que se consideran infractores de los derechos de autor quienes, sin autorización del titular de los mismos, realicen los actos previstos en el artículo 99 y en particular:

- “a) Quienes pongan en circulación una o más copias de un programa de ordenador conociendo o pudiendo presumir su naturaleza ilegítima.”
- “b) Quienes tengan con fines comerciales una o más copias de un programa de ordenador, conociendo o pudiendo presumir su naturaleza ilegítima.”
- “c) Quienes pongan en circulación o tengan con fines comerciales cualquier instrumento, cuyo único uso sea facilitar la supresión o neutralización no autorizadas de cualquier dispositivo técnico utilizado para proteger un programa de ordenador.”

Frente a las mencionadas infracciones, la LPI ofrece acciones y procedimientos que no sólo pueden plantearse en los supuestos de infracción de los derechos exclusivos de explotación, sino que también amparan y comprenden los derechos morales; del mismo modo, se

ofrece la protección tanto si los citados derechos corresponden al autor o a un tercero adquirente de los mismos.

El titular de los derechos podrá incluso exigir el **cese de la actividad** ilícita del infractor, solicitar medidas cautelares e incluso exigir una **indemnización** por los daños materiales y morales causados, en los términos previstos en los artículos 134 y 135 de la LPI.

### **2.7.3. Otras medidas de reacción**

Aparte de la tutela de la LPI, el titular de los derechos de explotación de una obra que ve menoscabados sus derechos por la violación de un tercero tiene otras vías de actuación. Las más relevantes son:

#### **a) Derecho contractual**

Frente a una violación de los derechos de autor derivada de un incumplimiento del contrato de licencia de uso de software, también es posible acudir al Derecho de Obligaciones tal y como recuerda el artículo 100.4 de la LPI. Eso significa que el titular, aparte de demandar al infractor por una violación de los derechos de autor, también puede demandarle por incumplimiento del contrato de licencia.

#### **b) Derecho penal**

La doctrina ha considerado tradicionalmente que los derechos de autor reconocidos al autor o titular eran también un bien jurídico protegible penalmente y la infracción de derechos de autor está tipificada en el Código penal español castigándose con penas de prisión y multa.

Concretamente, el artículo 270 del Código penal castiga con penas de hasta dos años de prisión a los que “con ánimo de lucro y en perjuicio de tercero, reproduzca, plagie, distribuya o comunique públicamente una obra, sin la autorización de los titulares de los correspondientes derechos”. Se castiga con la misma pena a quien “importe, exporte o almacene ejemplares de dichas obras”, y también quien “fabrique, ponga en circulación o posea medios específicamente destinados a facilitar la supresión no autorizada o la neutralización de cualquier dis-

positivo técnico que se haya utilizado para proteger programas de ordenador”.

El anterior Código penal exigía que esta actividad tuviera carácter de habitualidad, refiriéndose a la figura del perista profesional. No obstante, el Código penal actual (de 1995) requiere sólo la presencia de ánimo de lucro, es decir, el deseo de obtener una ventaja patrimonial, y esta ventaja puede ser, simplemente, no tener que pagar el precio de mercado de dichos productos. Cabe añadir que tampoco es necesario el pago de un precio para incurrir en el delito. Puede tratarse de un simple intercambio o de una cesión gratuita.

Éste es un tema que debe ser considerado muy relevante pues los poderes públicos están empezando a actuar contra las prácticas infractoras de derechos de autor por la presión que las agrupaciones de titulares de derechos están ejerciendo. Por ejemplo, recientemente se han producido en España 30 detenciones de compradores de programas ilegales distribuidos en CD-ROM. La operación ha sido el resultado de investigar las listas de clientes obtenidas por la policía tras la desarticulación de una red que ofrecía copias no autorizadas de software a través de Internet y que fue denunciada por la BSA (*Business Software Alliance*).

## 2.8. El sistema de derechos de autor anglosajón: el *copyright*

Como hemos visto, la normativa de propiedad intelectual está dividida en dos sistemas o concepciones diferentes, los derechos de autor y el *copyright*, que conviven en el mundo actual. Durante este curso nos hemos centrado en el análisis del sistema de derechos de autor por ser la tradición imperante en la Europa continental.

No obstante, creemos hacer una breve referencia al sistema anglosajón del *copyright*, más si tenemos en cuenta que el movimiento del software libre tiene su origen en Estados Unidos y que la mayoría de licencias y de casos que vais a estudiar están pensadas originariamente para este sistema jurídico. De hecho, las decisiones de los tri-

### Nota

Ver los dos sistemas o concepciones diferentes de la normativa de propiedad intelectual en el apartado 2.1.3.

bunales norteamericanos son actualmente las principales impulsoras de la evolución del derecho sobre el software libre, baste citar como ejemplo el caso SCO.

El sistema del *copyright* –derecho de copia en español– se basa en la concepción según la cual lo más importante no es el autor, sino la obra y los beneficios que su protección conlleva para el interés general. Como simplificación, podríamos decir que el sistema *copyright* es colectivista y protege la creación para estimular a los creadores como beneficio de interés general.

### 2.8.1. Marco legal

Si queremos estudiar el sistema del *copyright* debemos dirigirnos principalmente a estudiar el sistema legal norteamericano y el sistema legal inglés.

En Estados Unidos, la legislación más importante sobre *copyright* aplicado al software es:

- La Copyright Act, de 1976
- La Computer Software Copyright Act, de 1980.

Recientemente se aprobó la polémica Digital Millenium Copyright Act (DMCA) como desarrollo del Tratado TODA.

En el Reino Unido, la legislación en vigor actualmente es la Copyrights, Designs and Patents Act, de 1988, que consolidó las disposiciones legales anteriores recogidas en varias leyes y jurisprudencia.

A su vez, esta ley ha sido modificada:

- En 1992, por la Copyright Computer Programs Regulations, que incorpora la Directiva europea sobre los programas de ordenador.
- Recientemente en el 2003 se ha incorporado la Directiva Europea de Derechos de Autor en la Sociedad de la Información.



### 2.8.2. Características diferenciadoras principales

Es imposible resumir todo un sistema legal en unas pocas líneas, por eso vamos a proporcionaros unas breves pinceladas de las características más importantes del sistema del *copyright*, especialmente aquellas que más difieren del sistema de derechos de autor estudiado:

- a) El **objetivo**: el sistema del *copyright* se centra en la obra y el beneficio que la misma proporciona al interés común. El autor y sus derechos pasan, por lo tanto, a un segundo plano, al contrario que en la tradición de los derechos de autor.
- b) **Ausencia de formalidades**: tradicionalmente, el sistema del *copyright* exigía formalidades para que una obra fuese protegida, concretamente su registro, y no fue hasta el Convenio de Berna cuando se eliminaron estos requisitos. Actualmente, el derecho de *copyright* goza de una casi total ausencia de formalidades y requerimientos procedimentales, similar al sistema de derechos de autor. No obstante, Estados Unidos continúa exigiendo a sus propios ciudadanos, aunque no a los autores extranjeros, que registren sus obras en la oficina de derechos de autor para poder ser protegidos.
- c) **Nivel bajo de originalidad**: el umbral de originalidad (novedad, creatividad) requerido para que una obra esté protegida por el sistema del *copyright* es muy bajo: solamente se requiere que la obra sea nueva y creada por su autor, es decir, que no sea una copia.
- d) **Derechos morales**: en el sistema del *copyright* no se reconoce la existencia de derechos morales en los programas de ordenador, por lo que se permite la transferencia total de los derechos de autor sobre una aplicación o línea de código a un tercero. En cambio, como hemos estudiado, bajo el derecho continental, un contribuyente de código a un programa de software libre mantendrá siempre ciertos derechos sobre ella.
- e) **Duración**: en el sistema de *copyright* la duración de la protección de las obras es, en general, más larga. En Estados Unidos, por ejemplo, la duración es un tema complejo que puede ser 70, 95 o 125 años. En el Reino Unido al haberse armonizado la materia a nivel europeo, la duración es la misma que en España: 70 años desde la muerte del autor.

- f) **Sistemas de autoría:** en el sistema del *copyright*, cuando los autores son varios y no se puede distinguir las contribuciones (equivalente a obra colectiva), los autores se consideran como cotitulares bajo un régimen de *joint ownership*. No obstante, no existe el concepto de obra en colaboración cuando las contribuciones a una obra pueden distinguirse, y en estos supuestos nos encontramos ante una simple compilación de obras individuales.
- g) **Actos permitidos:** como excepción a los derechos de explotación, el derecho del *copyright* recoge los mismos supuestos que hemos estudiado para los derechos de autor:
- La descompilación por los fines de interoperabilidad entre programas.
  - La realización de una copia de salvaguarda.
  - La adaptación para la corrección de errores.
  - La realización de copias necesarias para el uso legítimo del programa.

Pero, aparte, existe un supuesto específico: el **uso justo**, o *fair use*, para fines de estudio, investigación no comercial (académica, principalmente), crítica o información periodística.

- h) **Transmisión o transferencia de derechos:** en el sistema del *copyright* se permiten varias maneras de transmisión de derechos en una obra protegida:
- La **cesión**: contrariamente a lo establecido en el derecho continental, en el *copyright* se pueden transferir todos los derechos del titular de una obra (morales y de explotación). La cesión debe ser por escrito y firmado por el cedente. El cesionario gozará de todos los derechos en la obra sin restricción.
  - Asimismo, la **cesión de obras futuras** (por ejemplo, en un contrato de prestación de servicios entre cliente y programador autónomo): en el momento de la creación, la obra automáticamente será titularidad del cliente.
  - La **licencia**: es la forma más usual de transmisión de derechos.

#### Nota

En el apartado 2.8.3 comentaremos brevemente la especialidad del uso justo.

#### Nota

Vamos a estudiar la licencia en las unidades 4 a 7.

- i) **Bases de datos:** hasta el momento, en Estados Unidos se ha rechazado una legislación similar a la europea para la protección de las bases de datos, por la presión de la comunidad científica y educativa que alegaban que una protección como la europea impediría el libre acceso a la información.

### 2.8.3. Doctrina del “uso justo”

Uno de los derechos básicos que otorga todo sistema de derechos de autor para proteger al titular de los derechos de explotación es el derecho a impedir o autorizar a otras personas a que reproduzcan la obra (Derecho de Reproducción).

Como hemos visto, este derecho está sujeto normalmente a ciertas excepciones. En el sistema del *copyright* norteamericano, una de las limitaciones más importantes a este derecho es la doctrina del “uso justo” o *fair use*. Así es, los derechos exclusivos que se otorgan al dueño de los derechos de autor no incluyen el derecho de impedir que otros usen en forma justa el trabajo registrado.

La doctrina se ha desarrollado mediante un número sustancial de decisiones judiciales a través de los años y ha sido codificada en la sección 107 de la actual ley de derechos de autor. La sección 107 contiene una lista de los diversos fines por los cuales se puede considerar “justa la reproducción de una obra en particular, como los propósitos de crítica, comentario, noticias, información, enseñanza, estudio académico e investigación”.

La sección 107, desarrolla la doctrina jurisprudencial norteamericana sobre la materia y establece cuatro factores que es necesario considerar para determinar si un uso en particular es justo o no, y por lo tanto, se excluye la violación de los derechos de *copyright*:

- 1) El **propósito y carácter del uso**, incluso si el mismo es de naturaleza comercial o si tiene propósitos educativos sin fines de lucro.
- 2) La **naturaleza de la obra** protegida con derechos de autor.
- 3) La cantidad y el grado en que la porción utilizada sea importante en relación con la totalidad de la obra protegida con derechos de autor.

#### Nota

Ver las excepciones a los derechos de explotación en el apartado 2.6.2.

**Lectura complementaria**

Para mayor información consultar: Alain Strowel (1993). *Droit d'auteurs et copyright*. Bruselas/París.

- 4) El efecto de dicho uso sobre el mercado potencial o el valor de la obra protegida con derechos de autor.

**2.9. Derechos de autor y software libre**

Como hemos visto a lo largo de esta unidad, la legislación vigente sobre derechos de autor y *copyright* permite que los autores o titulares de derechos de explotación de un programa de ordenador puedan estipular en las licencias de uso del software unas cesiones de derechos de explotación tan restrictivas que, de hecho, limiten al máximo sus usos. El movimiento del software libre pretende invertir esta situación y romper el monopolio de control que la ley proporciona al titular de los derechos de explotación de una obra, para permitir que el usuario del software libre tenga un amplio espectro de posibilidades de uso. El nacimiento del movimiento del software libre está estrechamente vinculado a lo que sus defensores llaman “la tiranía de los derechos de autor”.



Mediante las licencias de software libre, el titular de los derechos de explotación del programa de ordenador no pretende proteger sus derechos exclusivos de explotación que le reconoce la legislación sobre derechos de autor, sino asegurarse el disfrute de éstos por los usuarios del software.

Por el contrario, el titular permite expresamente a los usuarios **usar, modificar y redistribuir** el software, con o sin modificaciones. Este tipo de licencias de software libre no suponen que el proveedor renuncie a su condición de “autor” o titular del software, pero sí que significan una puesta a disposición de la comunidad de los derechos de explotación.

Como podemos imaginar, el movimiento del software libre ha supuesto una revolución en la conceptualización y uso de los derechos de autores tradicionales. Quizás el concepto más radical y controvertido dentro del movimiento del software libre es el *copyleft* defendido

por la Free Software Foundation, y por eso vamos a estudiarlo con más detenimiento, sin perjuicio de las referencias que se hagan en las próximas unidades, especialmente en la unidad 7 al estudiar la licencia GPL del proyecto GNU.

### 2.9.1. Copyleft: el concepto

El término *copyleft* hace referencia a una modalidad de distribución del software que fue iniciada por Richard Stallman en 1983 cuando comenzó el proyecto GNU de creación de software libre.

Como sabemos, el software libre se caracteriza por tener el código fuente y el código objeto accesible, lo cual lo hace potencialmente controlable, modificable, mejorable y copiable por los usuarios. La meta del movimiento defensor del *copyleft* es desarrollar software libre con las características esenciales que hemos estudiado en la unidad 1 (básicamente las cuatro libertades) pero impidiendo que se transforme en software propietario o cerrado. Como parte de esta meta, la Free Software Foundation creó la Licencia Pública General, o GPL en su abreviatura en inglés de *General Public License*.

De esta manera, el *copyleft* se puede conceptualizar como un aviso de reserva de derechos de propiedad intelectual con una peculiaridad, obliga a que los términos de distribución que permiten las libertades de utilización, modificación y distribución de una obra o sus derivaciones deban ser conservadas para todos los usuarios y licenciarios futuros. Así, sólo se garantiza la eliminación de las restricciones legales si los términos de distribución no son cambiados y no se agregan restricciones propias. De esta manera, el código y las libertades cruciales que definen el “software libre” se hacen legalmente inseparables. Las libertades quedan garantizadas para cualquiera que tenga una copia y se transforman en derechos inalienables.

Según los defensores de este movimiento, para que la filosofía del software libre sea realmente efectiva, las versiones derivadas también deben ser libres. El *copyleft* asegura que toda obra basada en el software libre distribuido con *copyleft* quedará disponible para los destinatarios del código.

El *copyleft* afecta sobre todo al derecho del usuario de software libre a distribuir programas derivados que éste haya desarrollado a partir del software original. Si la licencia de software libre es *copyleft* (como la GNU-GPL), el usuario sólo puede redistribuir el software original, o incluso el software derivado que haya desarrollado, también de forma libre. Por lo tanto, el usuario que pasa a ser proveedor-licenciante de un software derivado debe permitir a los nuevos usuarios el libre uso, modificación y distribución del software; con el mismo grado de libertad –como mínimo– contemplado en la licencia del software libre originario. Así, uno de los requisitos básicos de la redistribución del software con *copyleft*, con o sin cambios, es asegurar la libertad de copiarlo y modificarlo a los futuros usuarios o redistribuidores.



La idea central del *copyleft* es que se le da permiso a cualquiera para ejecutar, copiar, modificar el programa y redistribuir versiones modificadas –pero no se le da permiso para agregar restricciones propias. De esta manera, las libertades que definen el “software libre” quedan garantizadas para cualquiera que tenga una copia y se transforman en derechos inalienables.

El sistema de *copyleft* no es contrario ni modifica el marco normativo de los derechos de autor, sino que su utilización implica una inversión del uso habitual de los derechos de autor en las licencias propietarias.

En definitiva, coloquialmente podríamos decir que el *copyleft* es un uso nuevo de los derechos de autor tradicionales para adaptar sus características a la voluntad de unos creadores muy peculiares que desean romper el monopolio exclusivo del autor y ceder todos los derechos a los usuarios, garantizando al mismo tiempo que estos usuarios mantendrán sus características de libertad. Así, es gracias al sistema de derechos de autor como el creador de software libre puede otorgar una licencia donde se “abre y protege” el código del software y se impide que un tercero lo “cierre”.



Juego de palabras: el *copyleft* usa la ley de *copyright*, pero le da la vuelta para servir a lo opuesto a su propósito usual; en lugar de ser un medio para privatizar el software, se transforma en un medio de mantener libre el software.

Hay que tener en cuenta que *copyright* en inglés significa “derecho de copia”. Por el contrario, *left* es el participio del verbo *to leave* que significa: dejar, autorizar, ofrecer. Por lo tanto, *copyleft* podría traducirse como “derecho/autorización para copiar”. Desarrolladores de software propietario usan el *copyright* para restringir la libertad a los usuarios y restringir la libre reproducción; el movimiento GNU utiliza los derechos reservados para garantizarles su libertad y por eso invirtieron el nombre.

La definición de *copyleft* en el definitivo “Lexicon hacker”, el Jargon File, dice:

“*copyleft* /kop’ee-left/ [juego de palabras con ‘derecho de reproducción’] 1. El aviso de derechos reservados (‘Licencia Pública General’) llevado por GNU EMACS y otro software de la Fundación del Software Libre, concediendo reutilización y derechos de reproducción a todos los usuarios (aunque es necesario ver también virus públicos generales) 2. Por extensión, cualquier anuncio de derechos reservados intencionados para alcanzar metas similares.”

### **2.9.2. ¿Por qué no dominio público? Asegurar la libertad del usuario**

Sin lugar a dudas, la forma más simple de hacer que un programa sea libre es convertirlo en dominio público sin derechos reservados. Esto permitiría al creador compartir el programa y sus mejoras con el mundo entero. Pero esta solución abriría la posibilidad a que terceros utilizaran el software libre para fines contrarios a los de su filosofía de origen y lo convirtieran en software propietario o cerrado.

Para evitar esta posibilidad, el proyecto GNU creó el concepto *copyleft* y protegió el software GNU contra futuros intermediarios que intentaran restringir la libertad de los usuarios de redistribuirlo y cambiarlo.

Además, como hemos estudiado, en los sistemas continentales los derechos morales son indisponibles, por lo que tampoco es posible poner voluntariamente un software bajo el dominio público, renunciando a los derechos morales sobre el mismo. No obstante, incluso en los países de derecho anglosajón (en los que no existe la figura de los “derechos morales” aplicada al software, y sí que se puede poner un software voluntariamente en el dominio público), que es de donde proceden las licencias de software libre, también se ha querido dejar bien claro que el autor originario del software libre no renuncia a su condición de tal. Por lo tanto, es común de las diferentes modalidades de licencias libres mantener el aviso de autoría.

### 2.9.3. Funcionamiento del *copyleft*

Software libre y *copyleft* son conceptos generales. Para proteger actualmente un programa libre con *copyleft*, es necesario usar un conjunto específico de términos de distribución. La implementación específica de *copyleft* que se usa para la mayoría del software GNU es la Licencia Pública General de GNU (*GNU General Public License* o GNU GPL para abreviar) y la GFDL para los manuales GNU (utilizando un *copyleft* adaptado a documentos). También existe la Licencia Pública General Menor GNU (GNU LGPL), que se aplica a algunas librerías GNU.

Las obras originales distribuidas con *copyleft* añaden un texto como el siguiente:

*Copyright (c) 20yy < nombre de autor >*

*Esta información es libre. Usted puede copiarla, redistribuirla y/o modificarla bajo los términos de la Licencia Pública General GNU publicada por la Fundación del software libre, versión .. de la licencia o cualquier versión posterior.*

Aspectos concretos del *copyleft*:

- **Autoría:** el *copyleft* obliga a dejar bien claro en el documento reproducido quién es el autor original, desde la mera mención a incluirlo en la portada del documento.

#### Nota

Los elementos específicos de las licencias con *copyleft* se estudiarán en las unidades 7 y 8.



- **Modificaciones de la obra original:** si se hacen modificaciones en la obra original, el *copyleft* especifica que debe dejarse claro cuál es el contenido original y cuáles las modificaciones realizadas por los autores posteriores.
- **Libre no es igual a gratis:** que una obra/software sea de tipo *copyleft* no significa que sea gratuita. Se puede cobrar por obras *copyleft* y puede hacerlo tanto el autor original como otra persona. El *copyleft* se limita a establecer la libertad de explotación del contenido. O sea, se puede vender un software *copyleft*, pero quien lo adquiere puede copiarlo y modificarlo tantas veces como desee y debe garantizar que si lo distribuye, lo hará con *copyleft*.
- **Compatibilidad *copyleft* / software propietario:** un tema importante es la combinación de software libre con código de software no-libre. Tal combinación será inevitablemente no-libre; cualesquiera libertades que falten a la parte no-libre le faltarán también al todo resultante y ello podría arruinar los esfuerzos de la comunidad GNU para crear un software libre. Por ello, un requerimiento crucial para el *copyleft* es que cualquier agregación o combinación con software libre debe ser tal que la versión resultante sea también libre y bajo *copyleft*. Sin embargo, hay diferentes grados de combinación.

**Nota**

Los diferentes grados de combinación se van a comentar en la unidad 7.

#### 2.9.4. Software libre sin *copyleft*: licencias permisivas

Como hemos visto, no todo el software libre se distribuye con *copyleft*, de hecho sólo incluye *copyleft* el software distribuido bajo la licencia GPL de GNU (y otras licencias que estudiaremos en la unidad 7) y el concepto de *copyleft* es lo que distingue el “software libre” (concepto restringido de la FSF) de resto de software que nosotros llamamos también libre distribuido bajo otras licencias (ejemplo BSD, Mozilla) conocido como software abierto o *open source*.

Según la Free Software Foundation, máxima defensora del software distribuido con *copyleft*, liberar el código de un programa de ordenador bajo alguna otra licencia no *copyleft* no impide que hablemos de software libre ni de contribución a la comunidad, pero sí que impide asegurar la futura libertad de los usuarios de compartir y cambiar el software.

**Nota**

La licencia GNU GPL se estudia en la unidad 7.

### 2.9.5. Copyleft en obras que no son software

El *copyleft* fue inicialmente concebido para ser aplicado al software, pero también se puede aplicar a otro tipo de obra. Al estudiar con atención la licencia GNU GPL, se observa que dicha licencia puede ser fácilmente aplicada a información que no es software.



La GNU GPL afirma que:

“se aplica a cualquier **programa o trabajo** que contenga una nota ubicada por el propietario de los derechos diciendo que éste puede ser distribuido bajo los términos de la Licencia Pública General.”

Así, el “programa” puede no ser necesariamente un programa de ordenador –a cualquier trabajo de cualquier tipo al que se le puede aplicar derechos de reproducción, puede ser *copylefted* libremente con la GNU GPL.

La licencia GNU GPL hace referencia al “código fuente” de un trabajo; este “código fuente” significará diferentes cosas para diferentes tipos de información, pero la definición de “código fuente” –como lo establece la GNU GPL– se mantiene auténtica en cualquier caso: “el código fuente para un trabajo significa la forma preferida del trabajo para hacerle modificaciones”.

Además, la FSF ha creado una serie de **licencias libres para documentación**, sobre todo porque el software va acompañado de una documentación técnica a menudo necesaria para su uso. No tendría sentido distribuir el software libre sin distribuir la documentación correspondiente bajo términos similares. Por ello, se creó la *General Free Document License* para acompañar sus programas.

**Nota**

Observad que en ciertos regímenes legales, por ejemplo en España, el software debe ir acompañado de la documentación correspondiente y la licencia del primero cubre la segunda. Éste no es el caso de los países anglosajones, donde la obligación se limita a los materiales de diseño (y no a los manuales de uso, etc.)

La iniciativa de *Creative Commons* (en adelante, CC) es un proyecto originado por expertos en derechos de autor de las Universidades de Harvard y Stanford, en Estados Unidos. Su propósito es ayudar a los autores y creadores a distribuir libremente sus obras para uso del público y ampliar, por tanto, el número de obras creativas accesibles a todos. Se dirige sobre todo a las creaciones literarias y artísticas y no al software, y recomienda expresamente la GFDL para cualquier documentación y aplicación informática. Además, la CC propone un sistema privado, bajo derecho americano, para limitar la duración de la protección de *copyright* a 14 años en vez del plazo acordado por ley sobre la base de una declaración pública. Finalmente, permite dedicar obras al dominio público, también bajo las condiciones del derecho de autor de Estados Unidos.

Pero además de las mencionadas, existen otras muchas iniciativas, de entre las que destacamos las siguientes:

- Ram Samudrala ([www.ram.org](http://www.ram.org)) ha escrito un manifiesto filosófico de la música libre y crea música *copylefted*.
- El *copyleft* se experimentó en el Forum Social Mundial celebrado en Porto Alegre (Brasil) en enero del 2001. Los medios de comunicación vinculados a organizaciones sociales compartieron la información mediante autorización mutua para reproducir sus trabajos.

## 2.10. Derechos de autor y derecho *sui generis* de las bases de datos

Para tener una visión más general de todos los aspectos de derechos de autor relacionados con el software, consideramos conveniente introducirnos brevemente en la protección jurídica de las bases de datos. La Directiva 96/9/CE del Parlamento Europeo y del Consejo, sobre la protección jurídica de las bases de datos regula la materia a nivel europeo.

Para entender las diferentes protecciones existentes, es necesario tener en cuenta que las bases de datos pueden ser consideradas obras originales o no.

### Nota

La iniciativa de CC se estudia con más detenimiento en la unidad 7.

### 2.10.1. Bases de datos que son obras originales

La protección de las bases de datos encuentra su justificación en que el derecho de autor no sólo protege la originalidad absoluta, sino también la **originalidad derivada**, o sea, la creación a partir de otras creaciones. El esfuerzo intelectual y la originalidad pueden estar en la selección de los contenidos y en su disposición.

Por lo tanto, una base de datos puede ser considerada obra original y se le aplican el derecho de autor (el Libro I de la LPI que estudiamos en los apartados 2.5 y 2.6). En este caso, se protege la estructura (selección y disposición) que expresa la creatividad del autor, y no su contenido.

### 2.10.2. Bases de datos que no son obra original

¿Existe originalidad en un listado de teléfonos o en el listado de asociados a una organización? Difícilmente, si la selección responde a criterios de exhaustividad y la disposición a criterios funcionales (alfabético, cronológico, etc.). Ello no significa que estas bases de datos creadas con esfuerzo no deban ser protegidas, pero parece claro que no por los derechos de autor.

La protección puede venir por la normativa de competencia desleal o por el otorgamiento de derechos exclusivos argumentando que el riesgo de copia es demasiado elevado y que es necesario tener derechos para poder explotarlas.



La opción de crear un derecho especial, *sui generis*, a favor de los “fabricantes” de bases de datos es la elegida por la **Directiva 96/9 sobre la Protección Jurídica de las Bases de Datos** (incorporada en la LPI en 1998 en el título VIII).

Este derecho también beneficia a las bases de datos, que son obras originales siempre que cumplan con algunos de los requisitos.



Hasta el momento, en Estados Unidos se ha rechazado una legislación similar a la europea por la presión de la comunidad científica y educativa que alegaban que una protección como la europea impediría el libre acceso a la información. Actualmente, Europa está defendiendo en la OMPI un tratado de alcance mundial al derecho *sui generis* de las bases de datos, aunque cuenta con la oposición de Estados Unidos y de los países en vías de desarrollo.

Este derecho tiene las características siguientes:

- a) **Objeto de protección:** no protege la creatividad, sino una inversión sustancial, ya sea económica o en esfuerzo, que realiza el fabricante de una base de datos. La normativa de competencia desleal no protege actos de particulares sin finalidades comerciales o de empresas no competidoras, no obstante el derecho *sui generis* de las bases de datos y la propiedad intelectual no están afectados por esta limitación.
- b) **Derechos e infracciones:** al fabricante de la base de datos se le atribuye el denominado derecho *sui generis*, consistente en poder prohibir la extracción y/o reutilización de la totalidad o de una parte sustancial de su contenido, evaluada cuantitativa o cualitativamente. También puede impedir la extracción y/o reutilización repetida o sistemática de partes no sustanciales del contenido que representen actos contrarios a una explotación normal de la mencionada base de datos o que causen un perjuicio injustificado a los intereses legítimos del fabricante.
- c) **Duración:** en apariencia nos encontramos ante un derecho de corta duración (15 años desde la finalización de la base de datos), aunque si tenemos en cuenta el artículo 136.3 LPI, podemos llegar a la conclusión contraria, pues cualquier nueva inversión sustancial atribuye un nuevo plazo de 15 años.

## 2.11. Conclusión

A lo largo de esta unidad hemos realizado un recorrido completo por la figura de los derechos autor y por su aplicación concreta al software como obra con unas características muy particulares. Hemos introducido conceptos, analizado modelos de autoría y estudiado con detalle los derechos que el sistema otorga a los autores. Para hacer frente al monopolio del autor o titular de los derechos, hemos analizado las excepciones propias del software. Asimismo, hemos comentado las medidas para hacer frente a las posibles infracciones de los derechos de autor, del mismo modo que hemos comparado el sistema de derechos de autor con *copyright* y, finalmente, enumerado un supuesto muy específico, la protección de las bases de datos.

Durante todo el recorrido, se ha realizado en paralelo un análisis de las implicaciones que los derechos de autor tienen para el modelo de desarrollo del software libre y nos hemos detenido en el tema para estudiar una de las grandes innovaciones del software libre a toda la teoría de derechos de autor, el concepto de *copyleft*.

Advertimos que ésta no es la última vez que se van a estudiar los derechos de autor en este curso, pues en las unidades 4 a 8, se va a conocer la aplicación concreta de los derechos de autor a las licencias de software, y en las unidades 9 y 10, a través de unos estudios de casos, su aplicación práctica.

Esperamos que, al finalizar esta unidad, se hayan cumplido los objetivos que nos proponíamos al inicio del curso: adquirir una visión completa del actual sistema de protección del software por los derechos de autor. Aunque esto no es todo, es de especial relevancia para nosotros que no sólo se hayan adquirido los conocimientos teóricos, sino que también se haya comprendido la filosofía que subyace al método de protección del software por el sistema de derechos de autor, y la enorme relevancia que ello tiene para el estudio de todo el movimiento de software libre.

Finalmente, creemos que el lector debe tener presente que los derechos de autor no están aislados de los enormes cambios que afectan al mundo actual. Como hemos visto a lo largo de toda la unidad,

pero especialmente en el apartado de protección legal de medidas tecnológicas y sistemas de gestión de derechos, los derechos de autor se están adaptando a las novedades de difusión en entornos interactivos y la tecnología se convierte en un elemento esencial para proteger y gestionar los derechos de los titulares sobre las obras.

El futuro de los derechos de autor como sistema de protección del software es un tema difícil de predecir. Como estudiaremos con más detenimiento en la unidad 3, las grandes multinacionales (principales titulares de los derechos de explotación) consideran que el sistema es débil al permitir a los usuarios (mediante las excepciones que hemos estudiado) limitar los derechos de manera excesiva y están creando *lobbies* de defensa del sistema de patentes de software. Según ellos, el problema se halla en la naturaleza misma del sistema de derechos de autor que, como hemos visto, no protege las ideas e invenciones subyacentes en un programa de ordenador y permite la coexistencia en el mercado de programas de ordenador con código fuente distinto, pero con funcionalidades idénticas.





### 3. Propiedad industrial. Patentes de software y software libre. Marcas y secretos industriales

Hemos visto cómo el software puede ser protegido bajo la figura jurídica de los derechos de autor o *copyright*. La protección bajo el derecho de autor o *copyright* ha sido hasta ahora la figura por excelencia para proteger los programas de ordenador.

#### Nota

En el ordenamiento español, los **derechos de autor** son lo que se conoce simplemente como *propiedad intelectual*. El término *copyright* se corresponde con los “derechos de autor” en derecho anglosajón, con sus diferencias específicas.

En esta unidad vamos a ver en qué medida el software puede ser protegido también bajo la propiedad industrial, en concreto las **patentes**, y de una manera más tangencial, las **marcas** y los **secretos industriales**, pues dichas figuras ofrecen protecciones complementarias y, por tanto, adicionales a las que el derecho de autor y el *copyright* ofrecen a los creadores o titulares del software frente a los usuarios finales. De hecho, en su análisis veremos también cómo las distintas formas de protección no son excluyentes las unas de las otras.

Si bien la protección por marcas y secretos industriales no ofrece mayores polémicas, veremos los problemas que suscita la protección de los programas de ordenador como patentes, en especial frente al software libre, y cómo los titulares de los derechos sobre un programa propietario (las empresas informáticas más importantes) presionan a los gobiernos para que los programas de ordenador puedan ser también protegidos por la figura jurídica de las patentes, no ya sólo cuando estén implementados en una invención de la que forman parte, sino también cuando la protección se confiere al programa de ordenador en sí mismo considerado.

#### Nota

En los países anglosajones, bajo el término *propiedad intelectual* también se incluye la propiedad industrial, si bien en nuestro derecho son áreas complementarias.

Lo anterior ha dado lugar a una serie de posiciones doctrinales y legislativas contrapuestas y, aunque aún no puede asegurarse que esta cuestión haya quedado definitivamente zanjada, la opinión más común es que el software debería protegerse por el derecho de autor, y solamente se ampliará la patente también al programa de ordenador en la medida en que el programa de ordenador forme parte de una invención.

Sin embargo, la práctica demuestra que muchos programas de ordenador se han “colado” en varias oficinas de patentes de los distintos países del mundo como invenciones aisladas y, por tanto, en los que la invención sólo consiste en un programa de ordenador aislado, a pesar de las formas en que se trata de revestir la descripción de la patente. Si bien dichas patentes pueden ser anuladas en algunos casos, el gasto y el tiempo que ello requiere es considerable.

Por todo ello, en esta unidad nos centraremos en las patentes de software y la polémica que conllevan. Las marcas y los secretos industriales se tratarán de forma más breve.

Con el estudio de esta unidad, el lector alcanzará los objetivos siguientes:

1. Conocer el concepto de *patente*, su objeto de protección, los procedimientos de solicitud de la patente y cómo queda reflejada en los documentos oficiales.
2. Saber determinar cuáles son los titulares de una patente, su duración, el contenido y las diferencias de las patentes respecto a los derechos de autor o *copyright*.
3. Conocer el origen y el marco normativo de la protección del software por patente y las razones para excluir en algunos casos la patentabilidad del software.
4. Distinguir los aspectos relevantes de las patentes de software frente a los desarrollos de software libre.
5. Ser capaz de analizar en qué medida afectan las marcas al software y sus diferencias con las patentes y los derechos de autor y *copyright*.

6. Conocer las ventajas de proteger el software mediante la figura jurídica del “secreto industrial” y sus diferencias con las marcas, las patentes y los derechos de autor y *copyright*.

### 3.1. Definición y objeto de las patentes

Hemos visto cómo la propiedad intelectual –es decir, los derechos de autor y *copyright*–, protegen un conjunto de creaciones espirituales que forman parte de la cultura, el arte y la belleza. Sin embargo, estas figuras jurídicas también se han tenido que adaptar para proteger los programas de ordenador, puesto que los derechos de autor y *copyright* estaban pensados básicamente para evitar la copia de las obras literarias.

Los **derechos de autor** o *copyright* en un programa de ordenador protegen básicamente la manifestación o “expresión formal” del algoritmo considerando únicamente su contenido intelectual, pero no las concepciones o “ideas” que hay detrás del algoritmo y que están desvinculadas de su expresión.



En cambio, como enseguida veremos, la **patente** protege las aportaciones técnicas que enriquecen las posibilidades del hombre en el dominio de las fuerzas naturales para la satisfacción de sus necesidades sociales.

Una forma de plasmar esta idea sería decir que el derecho de invención regula las creaciones técnicas y, por tanto, protegería la “idea” en sí misma, por lo que tiene de solución técnica, y ello con total independencia de la “manera” en que la idea pueda ser precisada. Es decir, el derecho de invención supone un monopolio en el que se protege un pensamiento en abstracto y no la “expresión” concreta de la “idea”.

Por su esquema de jurisprudencia, en Estados Unidos desde 1981 es posible patentar “todo lo hecho bajo el sol por el hombre”. Sin em-

**Nota**

El concepto *invenciones implementadas por ordenador* se refiere a un sistema que usa programas para funcionar, pero que cumple la restricción de ejercer control sobre fuerzas de la naturaleza.

bargo, en Europa se ha discutido la posibilidad de exigir que un invento debe hacer un uso controlado de las fuerzas de la naturaleza para que pueda ser considerado parte de un campo de la tecnología y, por lo tanto, patentable. Esto no implica necesariamente que en Europa no pueda ser patentado nada que incluya software.

**Ejemplo**

En Europa, el concepto de **invenciones implementadas por ordenador** puede ser patentado como un todo. A pesar de ello, las invenciones implementadas por ordenador abren un espacio limítrofe en el que se podría argumentar que un programa que de alguna manera causa un efecto visual en una pantalla debiera ser patentable, con lo cual cualquier software sería susceptible de ser patentado, siempre y cuando la formulación de la patente incluya algún aspecto artificial de efectos físicos (como así ocurre con muchas de las patentes de software concedidas por la Oficina de Patentes Europea).

Las patentes de software abarcan desde las muy técnicas hasta las patentes sobre puros métodos sociales. Como luego veremos, las limitaciones actuales en la legislación europea sobre patentes, bajo una interpretación literal, hacen muy difícil patentar nada que no sea “máquinas que incluyan software, tengan un efecto técnico y aplicación industrial”. Esto deja fuera de juego las patentes de Internet (excepto para dispositivos de pago) y las patentes sobre métodos de educación.

Sin embargo, si el software se convirtiera en algo patentable, sería muy sencillo adquirir patentes sobre métodos de Internet, métodos de educación, métodos de consultoría, etc., mediante una ingeniosa formulación en la solicitud de patente que encapsulara esos métodos en técnicas de software. Tal vez, la Oficina Europea de Patentes deje que estas patentes contravengan las excepciones legislativas dando una interpretación diferente del mismo modo como ha hecho con el software.

**Ejemplo**

**a) Patentes sobre técnicas de software.** Se pueden encontrar unos cuantos ejemplos de patentes de Es-

tados Unidos con técnicas tan obvias como “XOR”, “Uso de instrucciones nulas para ralentizar un proceso” o “Hacer correcciones a un documento usando dos colores adicionales distintos” en:

<http://www.base.com/software-patents/examples.html>

b) **Patentes de Internet.** Se pueden encontrar unos cuantos ejemplos de patentes de la Oficina de Patentes de Estados Unidos (USPTO) o de la Oficina Europea de Patentes (OEP, EPO bajo sus siglas en inglés) sobre métodos obvios como “un solo clic (*one-click*)”, “Subastas de Internet”, “Referencia de negocios”, “Publicación de bases de datos en la web”, “Fork & ping” o “Técnicas de caché de Internet para WAP” en:

<http://www.freepatents.org/examples/>

c) **Patentes sobre métodos educativos.** Se pueden encontrar unos cuantos ejemplos de patentes sobre métodos educativos principalmente obvios que requerirían de las escuelas el pago de licencias de patentes a compañías multinacionales para poder implementar ciertas prácticas educativas basadas en ordenadores en:

<http://www.freepatents.org/examples/education.html>

d) **Colección de documentos imprimibles de la FFII.** En el enlace siguiente se incluye un gran número de patentes de software de la Oficina Europea de Patentes:

<http://swpat.ffii.org/vreji/prina/index.en.html>

### 3.1.1. Naturaleza jurídica

La patente es un derecho subjetivo que concede al titular la explotación en exclusiva de su invento con determinadas limitaciones, en especial, la temporal.

**Nota**

Este “monopolio” sólo nace desde la solicitud de la patente, claro está, si finalmente es concedida ante la correspondiente oficina de patentes.

**3.1.2. Procedimiento**

Las ideas que no sean registradas inmediatamente por sus creadores ante una oficina de patentes no serán protegidas por la ley como patente, perdiendo así el inventor su derecho al monopolio. Los derechos otorgados al inventor se hacen valer mediante un documento, un certificado otorgado por el Estado, llamado *patente*.

Ésta es una gran diferencia entre los derechos de autor y *copyright*, los cuales gozan de protección directa y automática en todo el mundo por el simple hecho de su creación, sin que sea necesario registro alguno. Si bien, como ya hemos visto en la unidad anterior, en algunos casos es conveniente registrar los derechos de autor o *copyright* a efectos de tener una prueba de la fecha de la creación del programa de ordenador.

**3.1.3. Definición de la patente**

Generalizando, podríamos decir que las patentes son “monopolios de explotación” sobre una “idea” limitados en un territorio y en el tiempo, tradicionalmente otorgados a alguien que ha creado una “invención” “novedosa” y de la cual puede sacarse “provecho a nivel industrial”.

**Nota**

Normalmente, las patentes se otorgan por un tiempo corto: 20 años en Europa y Estados Unidos; y, por tanto, inferior al largo periodo del derecho de autor y *copyright*: hasta 70 años *auctoris post mortem* en Europa por lo que se refiere a los derechos de explotación y hasta 120 años prorrogables en Estados Unidos, respectivamente.

De manera parecida a como sucedía con los derechos de explotación, en los derechos de autor y *copyright* (amén de la subsistencia de los “derechos morales”), una vez acabado el periodo de protección, la invención se incorpora al acervo de soluciones técnicas a disposición de todos, y cualquiera puede explotarla libremente.

Las legislaciones modernas no suelen definir el concepto de **invención**, simplemente se limitan a constatar que las invenciones que reúnen todos los requisitos formales y materiales establecidos legalmente son patentables. Así, el concepto *invención* es vago.



En **Estados Unidos** una invención tiene que ser: “novedosa”, “no obvia”, y “útil”. En cambio, en **Europa**, una invención tiene que tener: “novedad”, “altura inventiva” y “aplicación industrial”.

Por tanto, una diferencia importante estriba en que en Estados Unidos no se exige legalmente que las invenciones tengan aplicación industrial, ni que supongan “contribución técnica” o que tengan un “efecto técnico”, como luego veremos. En este sentido, la jurisprudencia de Estados Unidos ha llegado a admitir que los métodos de negocio y las fórmulas matemáticas pueden ser patentados en la medida en que tengan un resultado útil, concreto y tangible.

Generalizando, podríamos decir que una **invención** es la solución de un “problema técnico” “aplicable a la industria” que proporciona la posibilidad de obtener un cierto “resultado útil”, definida como la idea que tiene una persona sobre cómo combinar y disponer una materia o energía determinadas para que, mediante la utilización de las fuerzas naturales, se obtenga un resultado que sirva para satisfacer una necesidad humana y originar la solución de un problema técnico que no había sido resuelto hasta entonces.

Vemos, pues, que en el Derecho de patentes, los requisitos de protección se refieren a un factor de “creatividad” (esto es parecido al derecho de autor y *copyright*), si bien en cuanto a “novedad inventiva”, entendida ésta como “adelanto tecnológico” sobre el conocimiento previo y con “aplicación industrial” determinada. Por tanto, lo realmente protegido es la superación técnica o tecnológica de un estado de conocimiento (el “estado de la técnica”, que luego veremos). Complementario a lo anterior se encuentra el requisito de la inscripción de la invención en el registro público correspondiente, como antes hemos mencionado.

#### Nota

La Oficina Europea de Patentes ha llegado a sustituir el requisito de “aplicación industrial” en la medida en que ha llegado a entender que una invención es una solución técnica a un problema técnico.

#### Nota

La novedad inventiva implica un grado distinto al de la “originalidad” visto en el derecho de autor o *copyright*, según el criterio que se adopta en cada país.



Aplicada a un programa de ordenador, la patente protegerá no sólo la forma de expresión (tal y como también protege el derecho de autor y el *copyright*) del programa, sino las ideas caracterizadoras del mismo y sus equivalentes. Ello significa que, cuando se extinga la patente (normalmente en 20 años), cualquier tercero podrá explotar un programa basado en esas ideas y que se exprese de manera diversa (es decir, con distinto código fuente) a la del programa del titular del derecho de autor que continúa vigente (normalmente 70 años para las personas jurídicas).

#### 3.1.4. Clases de patente

Como veremos, existen dos modalidades de patentes de software (si bien dicha clasificación pueda resultar inútil en Estados Unidos o Japón en la medida en que las dos modalidades son patentables):

- a) Patentes de las invenciones implementadas por ordenador.
- b) Patentes por sí mismas (es decir, aisladamente consideradas).

#### Ejemplo

A pesar de que las patentes por sí mismas no se permiten en Europa, a continuación relacionamos:

- a) un ejemplo de patente de invención implementada por ordenador en el que, en realidad, lo único novedoso es el programa de ordenador en sí:

<http://swpat.ffii.org/patents/samples/ep538888/index.en.html>

- b) Un ejemplo de patente de software en sí mismo concedida por la Oficina de Patentes Europea:

<http://swpat.ffii.org/pikta/txt/ep/0903/889/>



- c) Otras patentes de software por sí mismo que se “han colado” en la Oficina Europea de Patentes se pueden encontrar en:

<http://swpat.ffii.org/patents/txt/ep/2003.en.html>

La invención siempre debe indicar lo que debe ser una determinada materia o energía, o cómo debe actuarse sobre ella para que produzca el resultado previsto. Así, podemos hablar de:

- a) **Inventiones de producto:** señalan cómo debe ser una materia o energía (por ejemplo, una máquina) determinadas. En la medida en que se permite patentar un ordenador programado, una red informática programada u otro aparato programado, hablaríamos de “inventiones implementadas por ordenador reivindicadas como patentes de producto”, es decir, de **aparato programado**.

#### Ejemplo

En el siguiente enlace tenemos un ejemplo de patente de hardware programado:

<http://swpat.ffii.org/pikta/txt/ep/0660/223/>

- b) **Inventiones de procedimiento:** señalan cómo debe ser la actividad en relación con la realidad material, es decir, cómo debe actuar el hombre con respecto a ella. En la medida en que se permita patentar un procedimiento realizado por un ordenador, una red informática o un aparato dirigido mediante la ejecución de un programa, hablamos de “inventiones implementadas por ordenador reivindicadas como patentes de procedimiento”, es decir, de **procedimiento técnico de producción**.

#### Ejemplo

Encontramos ejemplos de patentes de procedimiento en las patentes de Acacia Media Technologies, relativas a la transmisión y recepción de ficheros de audio y ví-

deo comprimido sobre Internet. Para más información, podéis consultar el siguiente enlace:

<http://www.informationweek.com/story/showArticle.jhtml?articleID=6504680&pgno=1>

### 3.1.5. Fundamentos de la patente

En términos generales, la patente encierra una relación contractual entre el autor de una invención y el Estado. La prestación del Estado consiste en otorgar al inventor un derecho de explotación de la invención en exclusiva (normalmente durante veinte años), en los que este último puede obtener el retorno de la inversión efectuada para producir la invención y los beneficios que compensen el riesgo asumido.

A cambio de ese derecho –y esas son, por así decirlo, las prestaciones del inventor–, el Estado divulga la invención para enriquecer el patrimonio tecnológico del país y obliga al inventor a explotarla en unas determinadas condiciones para asegurar que la tecnología patentada se utiliza realmente.



La pretensión que hay tras la patente es fomentar que las invenciones se comuniquen a la sociedad y no permanezcan secretas u ocultas, es decir, que el inventor se las reserve para sí. No se trata tanto de recompensar al inventor, como de incitarle a que no se guarde sus inventos en secreto. Aplicado al software, deberíamos entender que en una patente de software se debería pretender dar a conocer públicamente la idea que hay detrás de un algoritmo determinado.

En la explotación de una invención sin protección legal, el inventor mantendría en secreto su invención, ya que no la divulgaría más que en sus resultados (es decir, en los productos que sitúa en el mercado).

Por otra parte, para evitar proteger invenciones obvias que no impliquen actividad inventiva alguna y fomentar la investigación, se exigen una serie de requisitos legales y se otorgan una serie de protecciones al inventor. Cuanto más banal sea la invención aceptada como patente, más oneroso será para el inventor, con el grave perjuicio para la sociedad. En este supuesto se daría al titular de la patente un monopolio no justificado. Por ello, es necesario que la Administración (la correspondiente Oficina de Patentes) verifique y valore la petición de patente solicitada por el inventor antes de su concesión.

En definitiva, se trata de otorgar una patente sólo a una “verdadera invención” y dotar a su inventor de las “acciones legales” necesarias para que se pueda defender de modo efectivo frente a posibles usurpadores.

¿Por qué la patente es necesaria y conveniente? La esencia de los derechos que la patente confiere es impedir que los terceros exploten la invención protegida sin el consentimiento del titular.

En este sentido, podemos decir que las patentes confieren al inventor un monopolio de explotación que difícilmente puede obtener por el sistema de mantener en secreto su invención sin ninguna protección legal, dadas las características de la sociedad actual. La práctica nos demuestra que el derecho que confiere una patente compensa lo suficiente las obligaciones que conlleva, como lo demuestra el hecho de que los principales agentes de innovación en el mundo siguen una política sistemática de proteger sus innovaciones mediante patentes.

Veremos que esto ocurre también sistemáticamente en la práctica respecto al software y, en la medida en que alguna legislación no lo facilita, la industria informática que desarrolla software propietario hace presión sobre sus gobernantes para que acepten la patentabilidad del software.



Por otra parte, la divulgación de la invención, aunque debe ser lo suficientemente detallada como para que

**Nota**

Por *know-how* nos referimos a todo conocimiento reservado a ideas, productos o procedimientos empresariales (industriales o comerciales) que el empresario, por su valor competitivo para la empresa, desea mantener ocultos.

un experto medio en la materia pueda ejecutarla, no implica que se facilite con ello la usurpación de la patente, pues para explotar la invención de forma adecuada suele ser preciso, además, un determinado *know-how* que sólo posee el inventor.

Por tanto, alguien interesado en la tecnología patentada probablemente considerará más conveniente obtener una licencia del inventor a cambio de **cánones** o **regalías** (es decir, una compensación económica para el titular), ya no sólo en concepto de pago por la patente en sí, sino también por el *know-how* adicional no descrito en los documentos de patente (que también necesita para implementar la patente) y por la asistencia técnica necesaria; que intentar por sí mismo la explotación y, sobre todo, asumir el riesgo de infringir un derecho.



La reserva del *know-how*, mantenido en secreto por los titulares de las patentes, tiene sus efectos en los programas de ordenador que se registran en las correspondientes oficinas de patentes, en la medida en que el código fuente o los algoritmos no suelen ser publicados en la descripción de la patente, precisamente para mantener en secreto lo más valioso de los programas de ordenador: el código fuente.

En este sentido podríamos decir que la divulgación de la patente es un elemento de difusión de la misma de cara a su comercialización, si bien no es ése el sentido de fondo del legislador para justificar el porqué del monopolio conferido por patente.

Por otra parte, hemos de destacar que la decisión de proteger una invención mediante una patente encierra un cálculo de costes y beneficios porque, además de las contrapartidas mencionadas, obtener una patente tiene un coste considerable, y una vez obtenida, es preciso el pago de anualidades a la correspondiente oficina de patentes para mantener vigente el derecho sobre el registro de la patente.

Al igual que los derechos de autor o *copyright*, la protección que proporciona una patente tiene un carácter territorial y está limitada en el tiempo.



El hecho de que la patente tenga carácter territorial hace que el inventor tenga que pedir la concesión de una patente en todos los estados en donde quiere proteger su invento.

Precisamente para estimular la invención, existe una internacionalización de normas, incluso en su contenido material, por lo que podemos decir que el régimen jurídico de las patentes es, en muchos aspectos, común en los distintos países. Así, surgen una serie de tratados internacionales que han cambiado la antigua concepción exclusivamente territorial de las patentes.

### 3.1.6. Requisitos de patentabilidad

Para que una invención sea objeto legal de patente, debe cumplir con los siguientes requisitos comunes a la mayoría de las legislaciones:

- 1) **Novedad.** En términos vulgares, el vocablo *novedad* puede tener varios significados. La ley suele definir que una invención es novedosa cuando no está comprendida en el “estado de la técnica” (así lo define legalmente la ley española). La fecha de solicitud de la patente será la fecha en la que debe ser apreciada la novedad.
- 2) **Actividad inventiva.** La actividad inventiva exige a la invención una cierta entidad. Legalmente se suele decir que una invención implica una **actividad inventiva** cuando no resulta del “estado de la técnica” de una manera evidente para un “experto” en la materia.



En Europa, como veremos, no puede ser objeto de patente una invención que comprenda únicamente un programa de ordenador. Cuestión distinta es el caso de una invención que incorpore un programa de ordenador, pues la invención sí que es patentable. En la ac-

tualidad, difícilmente existen invenciones que no incorporen un programa de ordenador.

Sin embargo, en la práctica ocurre que la Oficina Europea de Patentes (OEP, o en su denominación inglesa: European Patent Office, EPO), está admitiendo el registro de patentes de software en sí mismo considerado (es decir, de forma independiente), tal y como también lo hacen las correspondientes oficinas de patentes de Estados Unidos (United States Patent and Trademark Office, USPTO), Japón y Australia. Al fin y al cabo, todo programa de ordenador se sirve finalmente de una máquina para ejecutar sus instrucciones y conseguir un determinado resultado.

El **estado de la técnica** está constituido por todo lo que se ha hecho accesible al público antes de la fecha de presentación de la solicitud de la patente en el país en el que se solicita, o en el extranjero por una descripción escrita u oral, por una utilización o por cualquier medio.

#### Nota

En España se entiende que no se considerará estado de la técnica una invención que, dentro de los seis meses anteriores a la solicitud de patente, haya sido divulgada:

- a) con un abuso frente al solicitante o causante;
- b) por haberse exhibido en una exposición oficialmente reconocida por parte del solicitante o causante, o
- c) por los ensayos efectuados por el solicitante o causante, siempre que no impliquen explotación comercial del invento.

Por otra parte, el “**experto**” al cual la invención no le resulte “evidente” debe tener una educación media en el sector de la invención y poseer conocimientos y cualidades “medios”.

- 3) **Carácter industrial.** Finalmente, y como último requisito de fondo de las patentes, se entiende que la invención es susceptible de “aplicación industrial” cuando su objeto puede ser fabricado o utilizado en cualquier clase de industria.



En definitiva, para poder obtener la concesión de una patente mediante el registro correspondiente de una oficina de patentes, las distintas legislaciones nacionales y tratados internacionales suelen exigir el cumplimiento de tres requisitos: la “novedad”, la “actividad inventiva” y el “carácter industrial” de la patente.

### 3.1.7. Excepciones de patentabilidad

Después de haber expuesto los requisitos de patentabilidad, consistentes en la necesaria concurrencia de la novedad, actividad inventiva y susceptibilidad de aplicación industrial, en Europa existen una serie de excepciones o supuesto de no patentabilidad. A los efectos del software libre destacaremos las siguientes excepciones a la patentabilidad:

- a) Los descubrimientos científicos y los métodos matemáticos (en la medida en que todos éstos ya existían o no pueden considerarse como industrializables).
- b) Las obras literarias o artísticas o cualquier otra creación estética, así como las obras científicas, los planes, reglas y métodos para el ejercicio de actividades intelectuales, para juegos o para actividades económico-comerciales, así como los programas de ordenador y las formas de presentación de información (en la medida en que todos éstos ya están protegidos por la propiedad intelectual, si bien, por lo que al software se refiere, la prohibición *strictu sensu* es en la medida en que se considere al programa aislado, pero no en conjunto y combinado con el hardware –más adelante entraremos en más detalle sobre esta cuestión).
- c) Las invenciones cuya publicación o explotación sea contraria al orden público o a las buenas costumbres.

Más adelante desarrollaremos con detalle la excepción de patentabilidad (que no existe en Estados Unidos o Japón, pero sí en Europa) referida al software aisladamente considerado (es decir, al software en sí mismo), así como los requisitos de la patentabilidad del software implementado en un ordenador (es decir, en combinación del hardware).

### **3.1.8. Procedimiento para la obtención de una patente**

Ya hemos dicho que, a diferencia de los derechos de autor y *copyright*, para poder gozar de la protección de una patente en un territorio determinado hay que solicitarla en la correspondiente oficina de patentes de ese territorio.

Veamos cómo se comprueban los ya mencionados requisitos de patentabilidad en los procedimientos para obtener la concesión de una patente, según exigen normalmente las legislaciones nacionales.

Primero hay que proceder a la solicitud de la patente ante el registro correspondiente (es decir, ante la oficina de patentes correspondiente al país o zona en la que se pretende la protección). Y a continuación veremos cómo es posible ampliar esa misma protección a otros territorios bajo lo que se conoce como *prioridad internacional* durante un primer plazo de preferencia frente a otros solicitantes. Una vez comprobados los requisitos mínimos, se procede a clasificar el contenido del documento mediante la Clasificación Internacional de Patentes resultante del Convenio de Estrasburgo de 1971.

En la **solicitud de patente** se procede a comprobar los requisitos de patentabilidad según las siguientes posibilidades (que suelen variar en cada país):

- 1) **Requisitos de forma:** la oficina de patentes correspondiente suele entender que este tipo de requisitos se cumplen cuando el peticionario aporta los documentos que la ley establece y cumple con las formalidades determinadas por dicha oficina.
- 2) **Requisitos de fondo (o de patentabilidad):** se comprobará el cumplimiento de los requisitos de “novedad”, “actividad inventi-



va” y “carácter industrial”. Podemos distinguir las variantes siguientes:

- a) **Sistema de simple depósito:** procedimiento por el que la Administración acepta (sin valoración en esta fase) las manifestaciones del petionario y su afirmación de que la invención reúne todos los requisitos para la concesión de la patente. En este caso, las patentes se conceden sin previo examen de la novedad ni utilidad, que se deja bajo la plena responsabilidad del petionario, lo cual hace que este tipo de sistemas de patentes sea barato y rápido. Las patentes concedidas bajo este sistema se conocen como **patentes “blandas”**.

**Nota**

En la medida en que exista una patente anterior o dicha invención ya existiera en el estado de la técnica, dichas patentes podrán ser atacadas con éxito fácilmente (por ejemplo, solicitando su nulidad). Así ocurre con las patentes españolas concedidas bajo el sistema antiguo de patentes español.

- b) **Sistemas de simple depósito con examen formal (o de examen de fondo parcial):** se debe presentar una instancia en la oficina de patentes correspondiente acompañada de una serie de documentos; se considera el momento de la solicitud aquél en el que se presenten todos los documentos ante la mencionada oficina.
- c) **Procedimiento general de concesión:** la oficina examina todos los requisitos formales de patentabilidad establecidos, salvo los de novedad y actividad inventiva, excepto cuando la invención carezca de novedad de manera manifiesta y notoria. A partir de entonces, el petionario deberá instar a la realización del informe sobre el estado de la técnica (si no lo hubiera hecho anteriormente). Dicho informe mencionará los elementos que pueden ser tomados en cuenta para apreciar la novedad y la actividad inventiva. El informe se publica en el boletín correspondiente para que terceros puedan expresar observaciones a la solicitud de patente. Finalizado el periodo de observaciones, la oficina concederá la patente y lo anunciará en el correspondiente boletín. La concesión de la patente se hace sin perjuicio de terceros y sin otorgar ninguna garantía al petionario sobre la validez de la patente.

- d) **Sistema de concesión con examen previo:** se comprueban todos los requisitos de patentabilidad (en Europa: “novedad”, “actividad inventiva” y “aplicación industrial”) y la patente no se concede hasta el examen de fondo de todas las posibles oposiciones (en vigor en España desde el R.D. 996/2001).

Con algunas diferencias relativas a las legislaciones de patentes de los distintos países, en la práctica, la fase de verificación del “estado de la técnica” se suele hacer en dos pasos:

- 1) **Examen de “novedad”.** En este paso se detectan de forma rápida y barata las ideas carentes de innovación y se consigue descongestionar la labor de los ingenieros que trabajan en la oficina de patentes correspondiente; se entiende que una idea es nueva si no era accesible ni conocida ni practicada por el público, desde el punto de vista del experto en la materia. Sólo los conocimientos anteriores a la fecha de la solicitud de patente destruyen la novedad, incluida, con ciertas excepciones en algunos países, una simple publicación local de una pequeñísima universidad en cualquier país del mundo apenas unos minutos antes, o su divulgación a la prensa por parte del inventor el día antes. Algunos países otorgan un “periodo de gracia” para evitar que el inventor se vea desposeído de su invención cuando la haya hecho pública un poco antes de solicitar una patente, normalmente para poner a prueba la atracción que la invención puede suponer en el mercado. Pero ello es siempre una excepción.

En un primer momento se suele hacer lo que se denomina la **búsqueda de anterioridades**. A continuación se procede a hacer una valoración, o lo que se denomina **examen técnico de la solicitud**, que culmina en la decisión sobre si existe novedad o no. Las personas que elaboran el examen técnico están mucho más cualificadas que las que realizan la búsqueda de anterioridades. La novedad de las invenciones se valora a nivel mundial, lo que ha llevado a la homologación internacional de los trámites.

- 2) **Examen de la “actividad inventiva”.** En esta etapa, una vez se ha visto que la idea no pertenece al estado de la técnica, se comprobará si es sustancialmente nueva y si tiene una cierta entidad inventiva, un cierto valor creativo y personal.

**Nota**

La actividad inventiva también se denomina así en terminología francesa (*activité inventive*). En terminología germanica se llama *altura inventiva* (*erfindungshöhe*); o *nivel de invención* (*standard of invention*), en terminología anglosajona.



La **actividad inventiva** que requieren las oficinas de patentes suele tener un nivel razonable, no demasiado alto para no descorazonar a los pequeños inventores. Así, la actividad inventiva se ha venido entendiendo como aquello que no es evidente para un experto en la materia.

Finalmente, para que la invención tenga un “carácter industrial”, es necesario que tenga una “aplicación (utilidad) industrial” clara y definida, lo cual también tendrá en cuenta la correspondiente oficina de patentes.

En principio, una patente “concedida por error” será nula cuando no cumpla los requisitos anteriores. Sólo un juez podrá declarar nula una patente tras interponer el correspondiente recurso ante los tribunales de justicia, y se publicará en el correspondiente *Boletín Oficial de Patentes* para conocimiento formal de todos los ciudadanos.

En definitiva, las etapas básicas se producen en todos los procedimientos de todos los países, con sus propias variaciones:

- presentación de la solicitud,
- examen formal preliminar de la solicitud,
- clasificación de la idea en un campo técnico,
- búsqueda de anterioridades,
- elaboración del informe,
- examen técnico de la novedad,
- decisión final (a favor o en contra),
- publicación (en el correspondiente boletín de patentes), y
- oposición o plazo de alegaciones (por parte de terceros).

**Ejemplo**

En Estados Unidos, las patentes sólo se publican después de haber sido concedidas, lo cual hace que la tramitación sea rápida, menos de 12 meses, dado que durante el procedimiento de concesión no hay alegaciones u observaciones por parte de terceros. Éste es uno de los sistemas de tramitación más perfectos.

Por el contrario, en la Oficina Europea de Patentes (OEP), las solicitudes se publican en el momento de hacer el informe técnico, es decir, antes de su concesión por parte de la Oficina Europea de Patentes, lo cual alarga su publicación unos 2 años por las alegaciones u observaciones de terceros. Ello supone una traba al monopolio corto de las patentes, pues alarga bastante el procedimiento hasta la concesión final de la misma. Así, el procedimiento español y el europeo son de los más lentos por su mayor burocracia, sin que ello aumente necesariamente la calidad del examen técnico.

### **3.1.9. Homologación internacional de trámites en la solicitud de patentes**

Como ya hemos apuntado, la homologación de trámites ha permitido la firma de una serie de tratados internacionales de cooperación administrativa entre los países en relación con las patentes cuya finalidad es diversa:

- a) Partiendo de una sola documentación ("solicitud internacional"), se pueden obtener simultáneamente varias patentes en varios países a la vez. Así se sigue en el procedimiento de patentes europeas del **Convenio de Munich, de 5 de octubre de 1973**, si bien el resultado es la obtención de los distintos títulos de propiedad simultáneos en los que se haya solicitado específicamente protección sobre el mismo objeto.



Cada uno de estos títulos tiene validez en su territorio como si se tratase de sus propios títulos nacionales.



En definitiva, bajo el procedimiento del Convenio de Munich, de 5 de octubre de 1973, existirá una patente por cada estado designado. En el Convenio de Munich se regula el procedimiento completo de tramitación de las solicitudes. Los Estados miembros del Convenio son los siguientes 27 países: Alemania, Austria, Bélgica, Bulgaria, Chipre, Dinamarca, España, Eslovaquia, Eslovenia, Estonia, Finlandia, Francia, Grecia, Holanda, Hungría, Irlanda, Italia, Liechtenstein, Luxemburgo, Mónaco, Portugal, Reino Unido, República Checa, Rumania, Suecia, Suiza, y Turquía.

La oficina que tramita dicha “solicitud internacional” de patentes es la Oficina Europea de Patentes (OEP).

- b) Para reducir los costes fijos de procedimientos, la búsqueda de anterioridades a nivel mundial sólo se hace una vez. Ésta es la finalidad del **Convenio de Cooperación en Materia de Patentes (PCT, Patent Cooperation Treaty)**, firmado en Washington, el 19 de junio de 1970. En este Tratado de Washington, sólo se incluye una regulación parcial del procedimiento: la “búsqueda” y el “examen”.



No todas las búsquedas y exámenes son aceptados, sino sólo los de los países más grandes y mejor dotados de medios técnicos y humanos, como Estados Unidos, Japón, la Oficina Europea de Patentes, etc., en los que la comprobación, el prestigio y la credibilidad de sus oficinas de patentes es muy grande, a fin de conseguir un sistema de “patentes fuertes”. Con ello se intenta evitar las búsquedas y exámenes de los sistemas de patentes que admiten la simple declaración del interesado sobre la novedad y actividad inventiva, sin una comprobación de tales notas. Del mismo modo, el dictamen sobre la “novedad”, la “actividad inventiva” y la “aplicación industrial” efectuado por una de las oficinas autorizadas en relación con la valoración de la solicitud, servirá para todas las demás.

Los países miembros del Tratado, a 6 de marzo del 2003, son 118, pertenecientes a los cinco continentes. La lista completa se puede obtener en inglés en la dirección de Internet siguiente:

<http://www.wipo.org/cfdpct/en/index.html>



Al igual que el Convenio de Munich, el Tratado de Washington no crea una “patente internacional”, sino una “solicitud internacional” que ofrece la posibilidad de tener tantas patentes nacionales como designaciones efectuadas a los estados comprendidos en dicho tratado.

- c) Finalmente, hay que destacar la promoción (aún no está en vigor) en Europa de un título único para varios estados, tal y como ocurre en Estados Unidos para todos los estados federales. En estos casos hablamos de una verdadera “patente internacional” y no de una “solicitud internacional”.

La Unión Europea trata de evitar su situación actual en relación con la concesión de patentes, caracterizada con simultáneos títulos de patentes nacionales que pueden bloquear las importaciones y exportaciones entre los territorios sujetos a un monopolio distinto, especialmente cuando la patente pertenezca a distintas personas en los distintos Estado miembros.

Hablamos, pues, de la patente comunitaria (inicialmente canalizada a través del Convenio de Luxemburgo de 1975, reformado en 1985), la cual en estos momentos está en trámites bastante avanzados y probablemente estará en vigor en poco tiempo.



En definitiva, existe una patente única (“patente internacional”), común a todos los estados de la patente comunitaria (no vale el hecho de que se conceda, transfiera, anule o caduque sólo en uno de los territorios, sino que se hará para el conjunto de dichos estados).

### 3.1.10. El documento de patente y las “reivindicaciones”

El hecho de que la novedad de las invenciones se valore a nivel mundial, ha llevado a la homologación internacional de los trámites, junto con los impresos que se utilizan para la solicitud de la patente, el contenido de los documentos administrativos de patente y la forma de redactarlos.

#### 1) El documento de patente

La primera página del documento de patente suele ser común a todos los países y en ella debe constar la información siguiente:

- Los datos del inventor y del titular
- El título de la invención
- La fecha de “prioridad internacional”
- La nacionalidad
- Los datos administrativos de la propia oficina de patentes
- El resumen de la invención



La **prioridad internacional** es el mecanismo esencial de cooperación entre varios países en materia de propiedad industrial y consiste en que en una solicitud de patente, cuando hay varios que la pretenden, es preferido y seleccionado entre todos el primero que cumple las formalidades administrativas (no basta, pues, ser el primero que la ha ideado).

Según la “prioridad unionista” del Convenio de París, el plazo para poder ejercitar la prioridad intencional, si ésta existe, es de 12 meses; sin embargo, según el Tratado de Cooperación de Patentes o Patent Cooperation Treaty, la “prioridad unionista” se amplía hasta un periodo de 20 meses.

El resumen de la invención no debe superar las 200 palabras, si bien no tiene ningún valor frente a los tribunales, aunque sí de cara a las

bases de datos técnicos a nivel mundial que recogen todas las invenciones que se registran en las oficinas de patentes más importantes –la de Estados Unidos, la europea, japonesa, británica, alemana, etc., pues a efectos jurídicos, lo que vale son las “reivindicaciones” que seguidamente veremos.

Para facilitar el trabajo cuando se trabaja en otros idiomas, esta información incluida en la primera hoja del documento de patente va precedida de una clave internacional denominada INID (*International Numbers for the Identification of Data*).

A la primera página se añadirán tantas como sean necesarias para detallar en qué consiste la invención. La forma en que se debe hacer también está homologada a nivel mundial. Estas hojas incluyen:

- a) La **descripción técnica** de la invención o **memoria**. El monopolio solicitado debe estar perfectamente descrito en la descripción técnica de la invención o memoria. La descripción debe ser clara y completa, ya que lo que justifica el monopolio sobre la idea es su difusión al público en general.

En la descripción técnica, o memoria, el inventor comunica su hallazgo a la sociedad y lo describe de un modo claro y leal. La descripción suele hacerse por escrito, si bien se admite acompañarla con dibujos cuando facilitan la comprensión de la invención (algunas oficinas de patentes exigen que se adjunten siempre dibujos para ser publicados en los correspondientes boletines oficiales de patentes).

La descripción técnica ayuda a interpretar el significado de las reivindicaciones (que son las que reivindican el objeto de la invención y vinculantes para los tribunales). La descripción debe contener las instrucciones suficientes para que una persona experta en el campo específico de la invención pueda reproducirla sin necesidad de ninguna actividad inventiva adicional. En el caso de los programas de ordenador, ello implicaría dar la suficiente información en la descripción en relación con las especificaciones del programa, pero sin necesidad de aportar el código fuente (lo cual, como veremos, es criticable).



- b) Los **dibujos**. Los dibujos son complementarios de la descripción de la invención y, al igual que el título y el resumen, no vinculan a los tribunales de justicia, pues sólo tienen una función auxiliar de información en el procesamiento de datos por las oficinas de patentes y las bases de datos privadas. Si bien los dibujos no siempre son muy útiles para describir las patentes de software por sí mismo.
- c) Las **reivindicaciones**. Aunque la descripción es la parte central del documento de patente, desde el punto de vista técnico no es ésta la que determina el contenido del monopolio industrial, precisamente porque las descripciones son largas y complejas. Si bien son utilizadas por los tribunales para determinar, bajo la "teoría de los equivalentes", por ejemplo, si puede haber equivalentes entre una invención patentada y otra cuya infracción de la patente se pretende.

En consecuencia, para condensar el contenido legalmente vinculante de la patente en unas pocas líneas, de forma breve y concisa, se idearon las reivindicaciones. Se entiende por *reivindicaciones* las declaraciones en las que el inventor determina cuáles son los elementos nuevos que integran el invento. Es decir, en cada reivindicación, el inventor formula un acto de apropiación pública de lo inventado, que debe contener sólo los aspectos y elementos nuevos que constituyen la invención.

A continuación incluimos un apartado específico para las reivindicaciones.

## 2) Las reivindicaciones



Las **reivindicaciones** (*claims*) de las patentes deben numerarse correlativamente y su redacción debe ser extremadamente cuidada, pues es lo único que vincula a los tribunales de justicia, sin perjuicio de que la descripción pueda servir para aclarar algún punto en concreto del texto de la reivindicación.

### Nota

Algunas bases de datos también incluyen artículos de revistas y otras referencias bibliográficas, pero no es lo más corriente.



Debido a que lo único vinculante de una patente a efectos de lo protegido son las reivindicaciones, a menudo el título o el resumen de la patente dan una idea equivocada del espectro de lo protegido por la patente, mucho más amplia de lo que la reivindicación específica en realidad. De hecho, en la práctica, muchos comentarios periodísticos se basan en los títulos y en el resumen, pero no en las reivindicaciones, que son, en realidad, lo único vinculante a efectos legales.

Por otro lado, existe una corriente jurisprudencial, como la “teoría de los equivalentes”, por la cual se extiende la protección detallada en las reivindicaciones.



En la práctica, los desarrolladores de software sólo necesitan el texto de las reivindicaciones para poder determinar si el producto que desarrollan puede infringir o no la patente según lo detallado en la reivindicación.

En definitiva, las **reivindicaciones**:

- limitan el ámbito de lo cubierto por la patente,
- deben ser examinadas a la luz de la invención publicada,
- estarán limitadas en función de lo que está públicamente al alcance de todos con anterioridad a la fecha de la solicitud de la patente.

Por lo que se refiere a las invenciones implementadas por ordenador, las reivindicaciones pueden ser de dos tipos:

- Producto:** es decir, como ordenador programado, red informática programada u otro aparato programado.
- Procedimiento:** es decir, como procedimiento llevado a cabo por un ordenador, red informática o aparato mediante la ejecución de un programa.

Algunos países imponen limitaciones al número de reivindicaciones que pueden hacerse en un mismo documento de patente, pues las reivindicaciones deben tener una unidad de concepto inventivo y exigen una tasa complementaria a partir de un número determinado de reivindicaciones, como si se tratase de dos o más patentes diferentes.

Dado que frecuentemente también se añaden tasas por hoja adicional, muchas veces se recurre al sistema de **reivindicaciones dependientes** (referencias de una reivindicación a otra). Por ejemplo, una muy general, seguida de otras que digan “lo mismo que la primera, caracterizado por...”. Sin embargo, si una se declara nula, la nulidad arrastrará a todas las que dependan de éstas. Así, una estructura de **reivindicaciones independientes** requiere un texto más largo (y ello normalmente implica tasas más altas), pero nos permitirá una mejor defensa procesal ante un proceso judicial, al seguir en vigor las restantes reivindicaciones.

Por otro lado, para no destruir la novedad, el inventor ha de buscar un equilibrio entre lo que es realmente nuevo en su idea (y, por tanto, reivindica) y la intención de ampliar al máximo su monopolio sobre la idea.

Finalmente, es importante tener en cuenta que no es necesario que la innovación que se pretende patentar exista realmente, sino que basta con que simplemente se haya descrito. Esto no es un problema en el caso de patentes industriales que se refieren a procesos físicos, puesto que la descripción es lo suficientemente detallada para implementar el invento. Y, de lo contrario, la patente no tiene valor porque el proceso no es aplicable industrialmente.

Pero en la industria del software, donde para una patente tampoco se requiere una implementación funcional, el texto de la patente puede ser lo suficientemente vago como para ser aplicado a otras ideas o implementaciones que las propuestas por los inventores. En tal caso, el legítimo inventor va a quedar excluido de utilizar y recibir los beneficios económicos que le corresponderían por su invento, ya que, con los actuales sistemas de patentes de software, la patente ha sido entregada a quien no ha implementado la idea.

### 3.2. Titulares, duración y contenido de las patentes



En principio, están legitimados para solicitar y obtener una patente las personas físicas y las personas jurídicas.

Al igual que con los derechos de autor, las distintas legislaciones nacionales establecen una serie de reglas, como por ejemplo, que las patentes gocen de los beneficios de la Unión de París en aquellos casos que dichas personas no tengan un vínculo con el país en cuestión (por ejemplo, cuando no tengan residencia o establecimiento industrial o comercial en ese país). En otros casos, también se acude a criterios de reciprocidad.



Quedan legitimados para solicitar y obtener la patente: el "inventor" y sus causahabientes.

Además, y como veremos, la patente es transmisible por todos los medios que el derecho reconoce. En ese caso, hablaremos del **titular de la patente**:

- a) **Cotitularidad**: la invención puede haber sido realizada por varias personas. En dicho caso, les corresponde a todas ellas obtener la patente en común.
- b) **Individualidad**: si, por el contrario, la invención ha sido realizada por personas distintas, pero de forma independiente, la patente corresponderá a aquel que presente la solicitud en fecha anterior, siempre que acabe siendo concedida.

#### Nota

Nótese que en España existe una presunción del solicitante para ejercer el derecho de patente.

- c) **Inventiones laborales**: el régimen que hallamos en las patentes es normalmente análogo al que hemos visto en derechos de au-

tor. Existen, sin embargo, algunas matizaciones en función del país:

- **Inventores-trabajadores:** si el objeto de su contrato de trabajo (de investigación sistemática y de dedicación profesional) es precisamente el hallazgo, la patente pertenecerá al empresario. El autor de la invención no tendrá derecho a una remuneración suplementaria, excepto si ésta excede del contenido del contrato o relación de trabajo. En caso contrario, la invención pertenece al trabajador.
- **Trabajadores simples:** si el trabajador realiza la invención en relación con su actividad profesional en la empresa y en su obtención hubieran influido predominantemente conocimientos adquiridos dentro de la empresa o a la utilización de medios proporcionados por ésta, el empresario tendrá derecho a asumir la titularidad de la invención o a reservarse un derecho de utilización de la misma. En ese caso, el trabajador tendrá derecho a una compensación económica justa.

También podrán ser reclamadas por el empresario las invenciones que, durante un año desde la extinción del contrato laboral, el trabajador realice, sin perjuicio de los derechos que la ley le reconoce.

### 3.2.1. Duración de la patente

Las patentes suelen tener una duración legal de 20 años, si bien algunos países contemplan la posibilidad de su prórroga en aquellos casos en que se tenga que asumir una cuantía mayor de costes y riesgos. Además, hay que tener en cuenta los perjuicios de retraso que supone todo el procedimiento administrativo hasta la concesión final de la patente.

#### Ejemplo

Así ocurre en Francia con los productos farmacéuticos en la medida en que será necesario compensar dichos costes y riesgos adicionales con un mayor tiempo de comercialización.

Por lo que a software se refiere, podemos generalizar que la duración de la patente será de 20 años.

Normalmente, las patentes caducan cuando:

- expira el plazo de 20 años,
- su titular renuncia,
- por falta de pago de la anualidad o sobretasa correspondiente, y
- por falta de explotación.

### 3.2.2. Contenido de la patente

La patente suele conferir el derecho a impedir que terceras personas realicen sin el consentimiento del titular los actos siguientes:

- a) La fabricación, el ofrecimiento, la puesta en el comercio, la utilización y la importación o la posesión del objeto de la patente.
- b) La utilización de un procedimiento o el ofrecimiento de su utilización cuando el tercero sabe o cuando las circunstancias evidencien que la utilización del procedimiento está prohibida por falta de consentimiento del titular.
- c) El ofrecimiento, la puesta en el comercio, la utilización o la posesión a los fines mencionados, del producto obtenido directamente por el procedimiento patentado.
- d) La entrega u ofrecimiento a entregar medios para la puesta en práctica de la invención patentada relativos a un elemento esencial de la misma a personas no habilitadas para explotarla, cuando el tercero sabe o las circunstancias hacen evidente que tales medios son aptos para la puesta en práctica de la invención y están destinados a ella (a no ser que los medios de que se trate lo constituyan productos que se encuentran normalmente en el mercado, salvo si el tercero instiga a la persona a quien entrega tales medios a cometer los actos prohibidos de las tres letras anteriores).

#### Nota

Ya hemos mencionado que la Patente Comunitaria aún no está en vigor, si bien ya está muy adelantada.

Sin embargo, los derechos conferidos por la futura patente comunitaria y en varias patentes de los distintos países europeos, entre otros, no se extienden a los actos realizados en la esfera privada y para los fines no comerciales o experimentales.

Por otra parte, los derechos adquiridos por la patente no afectan en principio a los actos realizados por el propio titular relacionados con el producto amparado por la patente que se realicen en el territorio de los estados. Es lo que se conoce como **agotamiento del derecho**, figura jurídica que, como veremos, también es aplicable a las marcas y tiene mucha relación con las importaciones paralelas.



Se pueden dar tres tipos de agotamiento de derecho:

- a) Cuando el agotamiento de derecho se refiera a un solo estado, se denomina **agotamiento nacional**.
- b) La futura patente comunitaria regulada en el Convenio de Luxemburgo prevé que, una vez que el producto haya sido puesto en el comercio de uno de los estados miembros del convenio por el titular de la patente o con su consentimiento, el agotamiento de derecho se amplía a los territorios que son parte del mencionado convenio (es lo que se conoce como **agotamiento comunitario**).
- c) Cuando el agotamiento se refiere a todo el mundo (es decir, a todos los estados de los cinco continentes), hablamos de **agotamiento internacional**.

Las patentes confieren a su titular un verdadero título ofensivo, persecutorio de la infracción. Por tanto, las **patentes de cobertura** no deben valer.

#### Nota

Las **patentes de cobertura** son aquéllas con solicitud posterior, exclusivamente realizadas por el solicitante con fines para conseguir liberarse de las medidas judiciales cautelares ante una eventual demanda del inventor anterior.

Las **patentes agresivas** pueden tener un papel importante en el valor de los intangibles de una empresa a la hora de vender la empresa. Además, tener patentes importantes agresivas elimina a los compe-

tidores del mercado, si bien esto se debe hacer con las cautelas oportunas para no ser sancionado por las leyes anti-monopolio (Derecho de Defensa de la Competencia).

Sin embargo, luego veremos cómo, en el mundo del software, muchas de las empresas más importantes dentro de la industria informática acumulan un gran número de patentes de software a efectos “defensivos”. A este fin, conciertan la actual práctica de las licencias cruzadas en compensación de las patentes que incumplen. Al fin y al cabo, conseguir que una patente sea declarada nula supone tiempo, información y dinero que muchos prefieren no tener que costear. Especialmente si tienen unas buenas cartas que jugar.



Para las pymes es muy difícil crear una cartera de patentes con las cuales competir en una industria dominada por unos pocos protagonistas que poseen derechos sobre miles de patentes cada uno. Contrariamente a la intención de un sistema de patentes, que pretende proteger al pequeño inventor o innovador, se logra aumentar la barrera de entrada a la industria del software de forma significativa y artificial. Incluso el supuesto simple hecho de litigar, para evitar el pago por una demanda injustificada de licencias por patentes, puede hacer peligrar la existencia a empresas que no tienen los recursos necesarios para esto, por el alto coste que conlleva.

#### Nota

Ver las obras derivadas en la unidad 2.

De manera parecida a lo que hemos tratado con las obras derivadas en derecho de autor y el *copyright*, cuando la invención no pueda ser utilizada sin una patente anterior, la explotación conjunta necesitará el consentimiento de los respectivos titulares, a no ser que los mismos hayan obtenido una licencia forzosa u obligatoria.

### 3.2.3. Transmisión y licencias de patentes

Algunas legislaciones reconocen al inventor un derecho personalísimo que le vincula de forma indisoluble como autor de la invención (podríamos hablar de un “derecho moral” del inventor).



**Nota**

Así lo hace la Ley de Patentes española, en la medida en que establece el derecho del inventor a ser mencionado como tal en la patente, en los supuestos en que la titularidad de ésta resulte transmitida.

En realidad, y como así ocurre en los “derechos de autor”, sólo las prerrogativas del inventor/titular que tienen **naturaleza patrimonial** son las susceptibles de transmisión.

En principio, la transmisibilidad se acepta no sólo para la patente “ya concedida”, sino también para la patente en fase de “solicitud”.

Las patentes suelen poderse transmitir por cualquier medio jurídico.

**Ejemplo**

En particular, la patente española puede transmitirse por: venta, traspaso, cesión, o aportación a una industria. La patente española también puede constituirse en usufructo, darse en garantía, mediante hipoteca mobiliaria, etc.

Si bien en estos últimos casos no hablamos puramente de una “transmisión” de la patente, pues su titularidad (o de la solicitud de patente, en la medida en que ésta también es objeto de lo anterior) continúa perteneciendo al mismo sujeto. Ello se entiende, claro está, sin perjuicio de las limitaciones que implica la constitución de los derechos más arriba mencionados en cuanto al pleno disfrute de las prerrogativas que la patente ofrece a su titular.

Algunas legislaciones imponen una serie de requisitos para su transmisión.

**Ejemplo**

Así, por ejemplo, en España, todos los actos *inter vivos* deberán hacerse constar por escrito y ser notificados a la Oficina Española de Patentes y Marcas. Por su parte, está previsto que la futura patente comunitaria ha de ser cedida por escrito, salvo que tenga su origen en una sentencia judicial.



Las **patentes** son **indivisibles**, de forma que no son susceptibles de transmisión parcial. En este sentido, no sería posible transmitir tan sólo una reivindicación, o transmitir la patente o la solicitud para una parte del territorio nacional. Pero, la indivisibilidad de las patentes no excluye que puedan pertenecer a varias personas, pues las cuotas sí que pueden ser transmitidas.

Dado que se habla del “carácter negativo” de las facultades del titular de la patente frente a terceros, más que actos de “cesión” o “transmisión” de derechos en las patentes, se dice que las licencias relativas a patentes no ceden al licenciatarario las facultades de su titular, sino que lo excluyen de las prohibiciones de explotación sin conocimiento del titular, con lo que el licenciatarario posee la totalidad o alguna de las facultades que integran el derecho exclusivo de patente.

En este sentido hay autores que entienden que las licencias no son una mera tolerancia o una autorización (como hemos vistos en los derechos de autor), sino una renuncia a la persecución por usurpación de la patente.

### Tipos de licencias de patentes

Las licencias pueden ser:

- a) Para **todo el territorio** o para un **territorio parcial**. En España, a diferencia de los derechos de autor, en patentes se presume que la licencia se extiende a todo el territorio nacional.
- b) **Exclusivas** o **no exclusivas**. En España, salvo pacto en contrario, se presumen no exclusivas y, en el caso de que sean exclusivas, el licenciatarario está facultado para ejercitar él mismo las acciones derivadas de la patente y el licenciante no podrá conceder otras licencias ni siquiera explotar la invención por sí mismo.

- c) De **pleno derecho**. Por ejemplo, en España, el titular hace un ofrecimiento voluntario en la correspondiente oficina para que quien lo desee pueda hacer uso de la patente en calidad de licenciatario. Supone pagar menores tasas anuales que devengue la patente. El ofrecimiento puede ser retirado, y se presume que lo ha sido cuando la patente cambia de titular. Sin embargo, este tipo de licencia no puede ofrecerse si en la oficina hay inscrito un licenciatario exclusivo o una solicitud de licencia de esta clase, y lo mismo a la inversa.
- d) En su **totalidad** o sólo para **alguna de las facultades** que integran el derecho de exclusión. En España, a diferencia de lo que hemos visto con los derechos de autor, en patentes se presume que el licenciatario tendrá derecho a realizar todos los actos que integran la explotación en todas sus aplicaciones.
- e) **Temporales**. En España, a diferencia de los derechos de autor, se presume que la licencia durará todo el periodo de duración de la patente.
- f) **Forzosas u obligatorias**. Las patentes están limitadas al estado de que se trata. Más adelante las veremos con más detalle.

En principio, las licencias de patente no podrán ser cedidas ni podrán concederse sublicencias, a no ser que se hubiera convenido lo contrario con su titular.

#### **3.2.4. Obligación de explotar la patente: las licencias forzosas u obligatorias**



Uno de los principios básicos para justificar el derecho de patente es imponer la obligación a su titular a explotar el invento en el territorio para el que se concede la patente.

Esta explotación debe ser suficiente para satisfacer la demanda del mercado. Sólo en caso de una serie de “excusas legítimas”(es decir, las de carácter objetivo técnico legal, ajenas a la voluntad y a las cir-

cunstancias del titular) para no explotar la invención o que impidan una explotación mayor de la patente, se exime al titular de esta obligación.

La mayor parte de la legislación de patentes incluye disposiciones relativas a las licencias obligatorias. Si el titular de una patente se niega a conceder una licencia de forma voluntaria, en casos excepcionales se puede conseguir una autorización en los tribunales o a través de un organismo administrativo.

#### Ejemplo

Por ejemplo, es posible conseguir las licencias obligatorias si la patente no se usa en ningún producto o procedimiento (*non-usus*) o en el caso de que dependa de una patente previa.

Según el **Convenio de la Unión de París**, el no cumplimiento de la obligación de explotar la patente comporta una licencia forzosa u obligatoria.

Según el **Convenio de Luxemburgo**, lo anterior se atenúa en el sentido de que el hecho de explotar la patente en un solo estado de los países miembros supone llevar a cabo la explotación en todos los Estados miembros. El motivo de ello es no atentar contra la prohibición del artículo 30 del Tratado de Roma.

#### Nota

La prohibición del artículo 30 del Tratado de Roma es la relativa a no obstaculizar las importaciones dentro de la Unión Europea.



Las licencias forzosas u obligatorias se suelen conceder cuando hay una falta o insuficiencia de la explotación de la invención patentada, una necesidad de la exportación o una dependencia entre las patentes.

En España, el titular de la patente posterior podrá exigir una licencia obligatoria si su invención posterior sirve para fines distintos o representa un progreso notable en relación con la patente anterior de la que es dependiente (ésta es una diferencia destacable respecto a las obras derivadas en derechos de autor y *copyright*). Si se trata de los mismos fines industriales, sus titulares podrán exigir la concesión de

licencias recíprocas. Podemos decir que en estos casos queda justificado en la medida en que existen motivos de interés público para la concesión.

#### Ejemplo

Así viene ocurriendo en España cuando, por ejemplo, una patente tuviera por objeto un procedimiento para obtener una sustancia química o farmacéutica protegida por patente, si la misma representa un progreso técnico notable respecto a la patente anterior de producto, ambos titulares tendrán también derecho a licencias obligatorias recíprocas.

### 3.2.5. Las patentes de combinación, selección y aplicación

En definitiva, los competidores o terceros no pueden fabricar ni vender los objetos o procedimientos de los que la patente es objeto. Sin embargo, ¿qué ocurre si el titular o un tercero encuentra una mejora a dicha invención? Puede suceder lo siguiente:

- a) Que el propio titular de la patente originaria encuentra (por sí mismo o por investigadores que corren a su cargo) una mejora a su invención. En ese caso, puede solicitar un título de propiedad anexo que se denomina *patente de adición*. Dicha **patente de adición** no tiene por objeto alargar la patente en el tiempo (20 años) ni en el espacio (por ejemplo, España), sino alargarla en su contenido.
- b) Que un tercero encuentre una mejora sobre una patente anterior. En ese caso, el tercero puede solicitar una **patente independiente** sobre la patente originaria anterior. En algunos casos, la mejora simplemente consiste en una nueva aplicación del mismo objeto.

#### Ejemplo

Por ejemplo, se descubrió que la aspirina, además de sus efectos analgésicos, tenía efectos cardiovasculares.

A diferencia de la adición, en este caso las patentes son siempre distintas, independientes y superpuestas. Su explotación se realiza mediante las licencias cruzadas (*cross licensing*), que se dan automáticamente.

#### Ejemplo

No podrían comercializarse la aspirina analgésica y la aspirina cardiovascular por separado en el mercado, porque para el consumidor es lo mismo.

- c) Que un tercero desarrolle una nueva tercera invención en una combinación de dos productos ya patentados o en la aplicación de dos procedimientos industriales registrados. Ello también puede ocurrir a partir de dos ideas de dominio público pero que, combinadas, pueden originar un efecto industrial nuevo.
- d) Que un tercero descubra una idea nueva en una selección de la patente anterior. Éstas se admiten raramente. Deberán tener suficiente entidad por sí mismas para poder justificar una licencia cruzada.

#### Ejemplo

Un ejemplo sería un motor patentado para una oscilación térmica entre X e Y grados centígrados, si alguien descubre que a Z grados centígrados permite ahorrar un 75% de combustible.

De este modo, las grandes firmas sienten cada vez más la necesidad de abonarse a las bases de datos internacionales sobre patentes, para tener acceso a los datos siguientes (por orden de prioridad):

- comprobar si una invención está patentada,
- comprobar potenciales infracciones de patentes,
- conocer cuáles son las patentes solicitadas por la competencia (es decir, hacerles un seguimiento),
- obtener información del mercado,

- estar al día en los cambios tecnológicos,
- encontrar una solución a un problema tecnológico específico,
- otras razones.



Lo anterior es necesario para planificar la propia investigación y, en su caso, abandonar o reorientar nuestro proyecto en el caso de que un tercero acabe de solicitar una patente similar o idéntica a la invención que estamos investigando y desarrollando.

De hecho, las grandes empresas informáticas desarrolladoras de software propietario, acuden constantemente al *cross licensing* en la explotación de sus productos. Por ejemplo, IBM es la empresa que tiene un mayor número de patentes en el mundo. Muy por debajo está Microsoft, que tuvo que añadirse a la carrera, dado que en sus inicios se apoyó sólo en los derechos de autor y *copyright*. Por no haber podido “compensar” patentes en su inicio, Microsoft ha tenido que pagar importantes cantidades a los correspondientes titulares de patentes, y en la actualidad ha cambiado de política en pro de conseguir un gran número de patentes de software y entrar de forma agresiva (como también ha demostrado en las otras áreas jurídicas) en el juego del *cross licensing* y, en definitiva, combatir su competencia.

### **3.2.6. Diferencias entre los derechos de autor o copyright y las patentes**

Es precisamente en el objeto de protección donde reside la diferencia más importante entre el sistema de derechos de autor y el de patentes. Mientras el primero protege las ideas o creaciones novedosas respecto al estado de la técnica, el derecho de autor se refiere a creaciones literarias, artísticas o científicas, expresadas en cualquier medio o soporte.

**Nota**

Ver los mecanismos de protección para los derechos de autor o *copyright* en la unidad 2.

El requisito **invención** se contrapone, por tanto, con la **originalidad** (entendida como simple distinción respecto a otra creación protegida) que se exige a las creaciones protegidas como propiedad intelectual.

Por otro lado, los derechos de autor o *copyright* nacen en el mismo momento en que se crea la obra sin que se requiera un acto declarativo o título, como es el caso de las patentes. Si bien hemos visto que existen mecanismos de protección para los derechos de autor o *copyright* (por ejemplo, el depósito notarial o el que se efectúa ante uno de los registros de propiedad intelectual) únicamente para constituir un elemento de prueba de la autoría, sin que el mero depósito suponga, por lo general, que el depositante sea el autor. Esto deriva en que la protección en los derechos de autor o *copyright* es más débil, no sólo porque la autoría de la obra se puede impugnar con más facilidad que en el caso de las patentes, sino por los derechos y acciones que se conceden al titular.

Otra diferencia se halla en la duración. La duración de los derechos de autor o *copyright* es mucho mayor que la de las patentes. Si bien en ambos casos es excesiva, dada la poca duración en la práctica de los programas de ordenador.

Finalmente, otra diferencia importante es que la patente en los programas de ordenador protegería sus funcionalidades, resultados, operativa o secuencias, mientras los derechos de autor o *copyright* protegería la forma de expresión del programa (la secuencia de instrucciones que realizan determinada función). Es decir, el código fuente y el código objeto como viene admitiendo la mayoría de la doctrina, en la medida en que este último no es más que una representación del código fuente en un lenguaje que la máquina puede ejecutar.

Los **derechos de autor** y *copyright* protegen la “expresión” del código fuente. Es decir, protegen cómo se ha expresado la idea del programa. Con ello se consigue, entre otras cosas, que el código fuente no sea copiado por otros. Sin embargo, los derechos de autor y *copyright* no impiden que otro informático escriba otro código fuente similar o completamente distinto que conlleve la misma idea.



La **patente** protege la “idea” que contiene el código fuente. Ello permite al titular de la patente impedir que otro informático escriba otro código fuente (por muy diferente que sea en su expresión al suyo) que contuviera la misma idea.



Ninguna de las dos figuras es totalmente idónea para proteger los programas de ordenador. Pero en la medida en que las patentes suponen dar un monopolio mucho más fuerte (en realidad excesivo para los programas de ordenador) al informático que desarrolla un programa, la opinión mayoritaria es que es más práctico y conveniente proteger los programas de ordenador como si se tratara de “obras literarias” y, por tanto, por derechos de autor y *copyright*. De todos modos, históricamente este punto se ha venido cuestionando.



Precisamente, es por el monopolio que se crea sobre la idea por lo que las empresas importantes que desarrollan software propietario están interesadas en proteger el software por patente. Proteger el software por patente supone dar una protección adicional a la que ofrecen el derecho de autor o el *copyright*.

Con la patente se consigue proteger la idea que hay detrás del algoritmo y del programa, dado que el inventor se encuentra protegido frente a todo tipo de equivalentes verbales y técnicos. En definitiva, con la patente el titular consigue protegerse de cualquier otro algoritmo diferente que incorpore la misma idea, lo cual no se consigue con el derecho de autor y *copyright*.

### 3.3. Origen y marco normativo de la protección del software por las patentes

Hemos visto que las patentes de software son objeto de discusión a nivel mundial. Veamos ahora los orígenes de la protección del soft-

ware como patentes, para luego entrar en detalle sobre la actual polémica resurgida en relación con las patentes de software.

Los ordenadores nos rodean en nuestras tareas rutinarias. Pero de nada sirve un ordenador sin los correspondientes programas que ordenan a las máquinas ejecutar las tareas, pues el ordenador es un artefacto pasivo que por sí mismo no ejecuta ninguna función útil.

Precisamente, algunos de los programas propietarios de ordenador instalados en las máquinas pueden suponer un coste considerable y, por lo general, unas limitaciones que debemos observar (esto último también es aplicable al software libre, en la medida en que está condicionado a la correspondiente licencia).

Muchos de nosotros tenemos algún amigo que tiene varios programas originales de los que nos ofrece copias “ilegales”. Además, la piratería, ya sea en redes o física (por ejemplo, la del “top manta”) llega a niveles preocupantes para los creadores de los programas, lo cual se manifiesta en diversos foros.

#### Ejemplo

Ejemplo de una de las reacciones defensivas por mecanismos legales contra la piratería de software son las recientes demandas interpuestas a principios del año 2003 en Estados Unidos con ocasión del intercambio usuario a usuario P2P (*peer-to-peer*) y, el pasado julio del 2003, el intento de denuncia criminal masiva en España contra usuarios P2P que intercambian programas de ordenador sin respetar los derechos de sus titulares/creadores.

En ambos casos, los demandantes intentan conseguir, ya sea vía una *subpoena* (la cual no necesita autorización judicial, como es el caso en Estados Unidos) o vía judicial (como es el caso de España en que, para no vulnerar los derechos fundamentales del usuario relativos al “secreto a las comunicaciones” y a la “intimidad”, es necesario que un juez lo autorice) que los prestadores de acceso a Internet faciliten la identifica-

ción de la persona que hay detrás de una dirección IP determinada, que intercambia dichos programas de ordenador con terceros sin respeto a los derechos de sus titulares/creadores.

En los primeros tiempos de la comercialización en masa de los ordenadores, los programas se comercializaban de manera indisociada con el ordenador (lo que se conoce como *bundling*) y no existían tecnologías que hicieran posible la realización de copias o su utilización fuera del ordenador. Los programas de ordenador del hardware no tenían coste alguno. No fue hasta que comenzaron a aparecer las empresas de servicios informáticos dirigidas a la elaboración de programas para satisfacer las necesidades de los usuarios cuando, en 1969, IBM decidió el *unbundling* (es decir, la separación) de sus negocios de hardware y software.

En la medida en que los grandes fabricantes de equipos informáticos querían evitar que el público conociera el invento en su totalidad, se recurría al **secreto empresarial** y la imposición de **cláusulas de confidencialidad** en la utilización de los equipos informáticos, en especial en lo que al software se refiere.

#### Nota

Es cierto que, en la práctica, el titular de la patente se suele reservar algún *know-how* que es, precisamente, la guinda de la patente. Sin embargo, también es cierto que lo que en realidad pretende el legislador con la patente y lo que asegura el monopolio del inventor es la publicación completa de la invención.

En ese primer momento, dado que el programa de ordenador se convierte en su última etapa en código máquina (programa objeto), que no son más que señales electromagnéticas que actúan físicamente sobre el hardware (los equipos informáticos) y producen un resultado nuevo obtenido en un medio físico, se entendió que podía ser objeto de protección por la figura jurídica de las patentes. Así lo interpretó la jurisprudencia de Estados Unidos en varias ocasiones.

#### Nota

La aparición de empresas de servicios informáticos se vio favorecida por el desarrollo técnico que posibilitaba la utilización de los programas por distintos ordenadores.

### 3.3.1. Razones para excluir la patentabilidad del software

A finales de los sesenta y principios de los setenta, los distintos legisladores nacionales y de tratados internacionales empezaron a rechazar la protección del software como patente. Además, a partir de 1968, la legislación antimonopolio de Estados Unidos condujo a una separación progresiva entre hardware y software, por lo que, durante esas décadas, se descartó a nivel internacional que el software “en sí mismo” pudiera ser objeto de patente por los siguientes motivos: legales, burocráticos, económicos o políticos, que detallaremos a continuación. No quedaba, sin embargo, descartada su protección por patente si el software iba implementado en una invención mayor.

Veamos dichos motivos en detalle:

#### Nota

Decimos *equiparar* en la medida en que el programa de ordenador se ha adaptado y tiene una regulación especial con muchas excepciones a la teoría general de los derechos de autor y *copyright*.

- 1) **Legales:** para que una invención pueda ser protegida por una patente, es necesario que estemos ante una regla del obrar humano que implica la utilización de fuerzas de la naturaleza para llegar a un resultado material y concreto (“carácter industrial”). En cambio, un programa por sí mismo consiste en una pura actividad intelectual, sin aplicación directa al campo de la técnica industrial, dado que idea soluciones a un problema lógico y matemático que se concreta en un algoritmo, se manifiesta en enunciados de operaciones y se traduce en combinaciones de señales eléctricas que, en definitiva, hacen funcionar los circuitos del ordenador. El resultado no se produce por la operatividad física de las señales eléctricas, sino por la forma en que han sido combinadas por el programador. Es por ello que el programa de ordenador se equipará a una “obra literaria” (es decir, a los derechos de autor y *copyright*). No debemos olvidar que un programa de ordenador dirige instrucciones al ordenador por medio de lenguajes naturales y simbólicos (el **programa fuente**, o **código fuente**), y es el aspecto técnico lo que realmente origina el movimiento y las funciones que el ordenador cumple, un resultado de la transformación de esas instrucciones (dadas primero en esos lenguajes naturales o simbólicos) en esquemas de señales electromagnéticas (el **programa objeto**, **código objeto** o **lenguaje máquina**), que son las que actúan sobre el ordenador, y producen el resultado.

En este sentido, podría decirse que el programa “invisible” puede expresarse de forma “visible” para el hombre por el método

de la “numeración binaria” en las que se representan los impulsos electromagnéticos por las cifras 0 ó 1 (es decir, pasa o no pasa la corriente eléctrica).



La corriente mayoritaria entiende que los programas de ordenador no constituyen una solución que consiga un determinado resultado por el empleo de las fuerzas naturales, sino que poseen un contenido esencialmente intelectual. Es decir, consisten en idear soluciones que cabe dar a un problema lógico y matemático que se concreta en el algoritmo, el cual se manifiesta en enunciados de operaciones (instrucciones), las cuales se traducen en señales eléctricas que actúan sobre el ordenador.

La operatividad de las señales eléctricas no se debe a las señales en sí mismas (pues los ordenadores funcionan de igual modo), sino en la forma en que han sido combinadas según determina el programa, en definitiva, el programador. Por eso, gran parte de la doctrina entiende que el programa es obra de una concepción humana y no de fuerzas naturales.

Por otra parte, las ideas que contienen los programas no son siempre, ni necesariamente, nuevas. Y, como hemos visto, la “novedad” es una condición fundamental para obtener una patente. Además, muy pocos programas de ordenador harían prueba de una “actividad inventiva suficiente” (según las estimaciones de la OMPI, tan sólo un 1% lo son). Finalmente, recuérdese que el derecho de autor y el *copyright* tan sólo deben cumplir con el requisito de la “originalidad” (si bien el rigor de dicho criterio varía según las distintas legislaciones), condición mucho más fácil de cumplir que el de la “novedad” y “altura inventiva”.

En definitiva, existían múltiples razones legales que hacían poco aconsejable proteger los programas de ordenador en sí mismos considerados, por medio de las patentes.

- 2) **Burocráticos:** a diferencia de los derechos de autor y *copyright*, para que una invención pueda ser objeto de una patente, es ne-

#### Nota

Ver el derecho de autor y el *copyright* en la unidad 2.

#### Nota

Recuérdese que las figuras jurídicas de los derechos de autor y el *copyright* confieren, con ciertas matizaciones, protección inmediata al software por el mero hecho de su creación, sin necesidad del requisito de concesión administrativa y registro.

**Nota**

Ver el Convenio de Berna y el Convenio de Ginebra en la unidad 2.



La corta vida práctica del software (enseguida queda desfasado y necesita versiones sucesivas actualizadas y mejoradas) suponía una traba importante para los creadores y titulares del mismo, que sólo verían sus productos protegidos al cabo de un tiempo considerable desde el momento de la solicitud de la patente. Además, el hecho de que los programas pudieran ser protegidos por patente suponía dificultades de orden práctico para las correspondientes oficinas de patentes y un colapso mayor de éstas, dado que uno de los aspectos que requiere buena parte del trabajo de estas oficinas es la comprobación del examen de novedad y de la altura inventiva; y, dada la explosión de la industria informática, añadir una categoría de invención más para el software no era deseable en la medida en que las patentes serían concedidas aún mucho más tarde.

Por otra parte, la figura de los derechos de autor o *copyright* representa una solución mucho más rápida e internacionalmente más eficaz en la medida en que la protección se logra de una forma inmediata vía los convenios internacionales sobre derecho de autor, en especial el Convenio de Berna y el Convenio de Ginebra.

- 3) **Económicos:** el procedimiento de concesión de patente es francamente caro, no sólo por los gastos que acarrea el depósito de la patente, su renovación y el pago de un canon anual, sino también el coste que suponen los honorarios de los profesionales (ingenieros, abogados, etc.) que intervienen en el tema. La suma que debería pagar la industria informática que comercializa varios centenares de programas al año desaconsejaban también, en principio, la protección del software en sí mismo por la figura jurídica de la patente. Sin embargo, ello no supone obstáculo alguno para las empresas informáticas más importantes.

Además, los primeros programas de ordenador eran muy simples y el verdadero negocio estaba en el hardware, por lo que los titulares del software no veían en esos primeros momentos una necesidad ferviente para conseguir una protección fuerte al software.

- 4) **Políticos:** en Europa, existía en realidad una desconfianza para dejar abierta la vía de la protección del software por las patentes a los Estados Unidos, pues éstos ya dominaban la producción del hardware.

### **3.3.2. La exclusión de la patentabilidad del software en sí mismo: normativa y jurisprudencia**

La situación desembocó en una primera decisión a nivel internacional de excluir la protección del software por vía de la patente por considerarla poco adecuada o poco deseable. Pero sólo durante unas décadas.

A pesar de unos primeros atisbos de una jurisprudencia en Estados Unidos favorable a la patentabilidad de los programas de ordenador considerados aisladamente, a partir del caso “Gottschack frente a Benson” de 1971 y del caso “Parker frente a Flook” de 1978, los tribunales en Estados Unidos entendieron unánimemente que, por la vía de los *mental steps*, un método que puede desarrollarse por la mente humana, no puede patentarse, en la medida en que los programas de ordenador no estaban relacionados con la materia física y que podían ser ejecutados por la mente humana. Ello hizo que en Estados Unidos las patentes que tenían por objeto un programa de ordenador en sí mismo se rechazaran automáticamente.

Otros países (así lo hizo Francia en su Ley de Patentes de 1968) establecieron en su legislación una exclusión clara y rotunda respecto a la posibilidad de proteger un programa de ordenador por la figura jurídica de la patente. De forma paralela, la jurisprudencia de dichos países también reflejó la postura de la legislación (así, destacan el asunto francés Mobil Oil de 1973 y el alemán Disposition Program de 1976).

La regla de la no patentabilidad también desembocó en el Tratado de Cooperación en materia de patentes firmado en Washington el 19 de junio de 1970.

**Nota**

Como ya hemos anunciado, y como veremos con más detalle, la exclusión es sólo para el software considerado en sí mismo.

**Nota**

En el párrafo 1 se define lo que es objeto de patente.

**Nota**

Recuérdese que dicho tratado trata de evitar dispersión de esfuerzos, centralizando el procedimiento de registro mediante la creación de: (i) una demanda internacional única sometida a una búsqueda internacional de novedad; y (ii) la institución de un examen preliminar internacional también centralizado.

Igualmente, en el también visto Convenio de Munich sobre la patente europea de 5 de octubre de 1973, el cual excluye al software del beneficio de la patente en su artículo 52.2.



El texto de dicho artículo dice:

“No son consideradas como invenciones a los efectos del párrafo 1, en particular: [...] c) los planes, principios y métodos para el ejercicio de actividades intelectuales, para juegos o para actividades económicas, así como los programas de ordenadores.”

Lo anterior hizo que los países europeos adoptaran redacciones similares en sus legislaciones nacionales.



En concreto, **España** estableció en el artículo 4.4 c) de su **Ley de Patentes de 1986** el siguiente texto:

“No se considerarán invenciones en el sentido del apartado anterior (en el mismo se define lo que es objeto de patente), en particular los planes, reglas y métodos para el ejercicio de actividades intelectuales, para juegos o para actividades económico-comerciales así como los programas de ordenador.”

La postura de los textos legales anteriores también se ha visto reflejada en los distintos casos planteados ante los tribunales de los distintos países europeos acogidos al Convenio de Munich.





En definitiva, internacionalmente se entendió que, en principio, los programas de ordenador (como luego veremos, matizándolo a “en sí mismos”) no podían ser objeto de patente. A pesar de ello, en algunos casos el software sí que podía ser objeto de patente en la medida en que se trataba de una invención implementada por ordenador.

### 3.3.3. La protección del software por derechos de autor o por patente

El debate histórico sobre la conveniencia o no de proteger el software por las patentes ha surgido nuevamente por la presión a nivel mundial de las grandes empresas desarrolladoras de software de Estados Unidos y su intención de proteger al máximo el software con las distintas figuras jurídicas que, en su caso, sean aplicables.



Detrás de ello está el afán de proteger no sólo por el derecho de autor o *copyright*, y por tanto, conseguir la prohibición de la copia literal de la expresión formal del algoritmo (es decir, cualquier expresión del código fuente y código objeto) de los programas de ordenador, sino también por las patentes: el puro pensamiento o cualquier idea que está detrás del algoritmo y, por tanto, conseguir proteger cualquier expresión de esta idea en otras formas posibles en las que pueda ser manifestada o declarada la misma idea de ese algoritmo (aunque sea con otro algoritmo o código fuente y código objeto distintos).

No hay que olvidar que los países más industrializados, a los que van destinados los ingresos que genera la industria informática, buscan una protección fuerte del software, mientras que los países menos industrializados tienen como objetivo introducir las licencias obligatorias.

**Ejemplo**

Así era el caso de Brasil en 1984 que, sin embargo, tuvo que rectificar de postura al ser amenazado por el entonces presidente Reagan con el inicio de procedimientos para ser sancionado comercialmente por Estados Unidos.

Dado que a finales de los años sesenta y principios de los años setenta se entendió que, en principio, el software no podía ser protegido por la figura jurídica de las patentes, los países más industrializados buscaron vías alternativas en aras a conferirle una protección. En esos momentos ya existía gran presión de las grandes empresas informáticas ante las pérdidas sufridas, ya entonces, por la piratería de sus programas:

- 1) Por un lado, en esos momentos en los que se decidió excluir la protección del software por las patentes, los países buscaron una **protección específica** o *sui generis* para el software en sí mismo que nunca llegó a prosperar.

**Ejemplo**

Destacan los trabajos iniciados a principio de los setenta por la OMPI que desembocan en las disposiciones tipo sobre protección de software de 1978 y el Proyecto de Tratado Internacional sobre Protección de Programas de Ordenador de 1983, así como otros proyectos legislativos nacionales como el del MITI de 1982 en Japón o el del Gobierno brasileño de 1984.

- 2) Por otro lado, los países buscaron que la protección del software por sí mismo se hiciera por los **derechos de autor** y *copyright*, tal y como ya hemos comentado en la unidad 2. Ésta fue la solución que finalmente se adoptó a nivel internacional, la cual está fundamentalmente vigente hasta la actualidad, como decisión puramente pragmática.



En realidad, la consecución de una inmediata protección internacional que se ofrecía a través de las convenciones internacionales sobre el derecho de autor

**Nota**

Si bien la mayoría de los autores entienden que el derecho de autor y el *copyright* son las figuras jurídicas por excelencia para proteger al software, tampoco son las más ideales para ese fin.

(impedir la reproducción o copia del software) sin necesidad de procedimiento de registro alguno tuvo una importancia decisiva en la opción elegida en un primer momento. Sin embargo, dicha protección trata de ser actualmente ampliada por los intereses de las empresas informáticas más influyentes.

Sin perjuicio de un prudente depósito notarial en el que se garantiza la confidencialidad del contenido exhaustivo del programa de ordenador, los programas de ordenador no siempre se registran ante el correspondiente Registro de la Propiedad Intelectual porque el posible acceso al código fuente facilitaría incluso el trabajo de los imitadores o piratas, ya que se puede plagiar fácilmente un programa partiendo de su código fuente. Si bien en otros casos aparece aconsejable su depósito ante el Registro de la Propiedad Intelectual, ya sea por las presunciones *iuris tantum* (que admite prueba en contrario) de titularidad que goza el primer inscrito en el registro público (éste sería el caso de España) o por el requisito necesario de registro para poder perseguir judicialmente una infracción (éste sería el caso de Estados Unidos).

**Nota**

Las distintas legislaciones incluyen limitaciones respecto al acceso del público en general al código fuente depositado en el Registro de la Propiedad Intelectual.

Además, la imitación de las ideas sin copiar las palabras y, por tanto, de obtener un mismo resultado modificando o invirtiendo las sentencias o instrucciones lógicas del algoritmo es lo que se denomina *ingeniería inversa*. Y, a fin de no facilitar la ingeniería inversa, también existen disposiciones legales bajo las figuras jurídicas del derecho de autor y *copyright* que la limitan.

Aun así, ni el derecho de autor ni el *copyright* dan una solución ideal para impedir el uso no autorizado del software. Sin embargo, sí que suponían una rápida respuesta jurídica para los deseos de las grandes firmas informáticas con las que impedir la cada vez más creciente piratería del software. Además, el alineamiento claro

**Nota**

Ver el depósito notarial en la unidad 2.

de las legislaciones de los países más desarrollados con la vía del derecho de autor y *copyright* (impulsado por las grandes empresas informáticas en aquellos momentos en los que se excluyó la vía de protección por las patentes) fue decisivo en la adopción de esta postura en el resto del mundo.

Estados Unidos fue el primero y principal exponente de esta segunda solución; reformó su Copyright Act de 1976 para incluir que los programas de ordenador son objetos propios del *copyright*, con las adaptaciones que requería su inclusión.

Le siguieron, en 1985, los acuerdos adoptados a nivel internacional en el seno de la OMPI (el cual administra el Convenio de Berna de 1986), así como también Alemania, Japón, Gran Bretaña, Canadá y Francia.

España también se incorporó a este segundo movimiento legislativo en aras a asegurar la protección del software por la figura jurídica de los derechos de autor con la reforma de la Ley de Propiedad Intelectual de 1987.

**Nota**

En España se incluyeron los artículos 95 a 100 relativos a la regulación específica de los programas de ordenador por el derecho de autor). Si bien antes de dicha ley ya se podía entender que los principios de la Ley española de Propiedad Intelectual de 1879 podían haber sido utilizados para interpretar que los programas de ordenador podían ser objeto de protección bajo el derecho de autor, aunque no hubo ningún pronunciamiento judicial español que resolviera en este sentido.

Se entendió básicamente que la protección de los programas de ordenador debía salvaguardar las inversiones que se realizan en su creación, si bien incitando su divulgación. Además, debían ser protegidos frente a su fácil y poco costosa duplicación por aquellos que incluso no tienen conocimientos de tecnología dado que las medidas tecnológicas de protección no siempre son eficaces y, además, resultan costosas. Pues no hay que olvidar que un programa de ordenador es susceptible de ilimitada repetibilidad y de operar simul-

táneamente en un número indefinido de ordenadores o de estar almacenado en múltiples soportes.

Así pues, se partía de la filosofía de que el programa de ordenador representa intereses económicos e inversiones importantes porque implica un trabajo intelectual estimable y valioso que debe protegerse contra su posible copia. Y los derechos de autor y *copyright* aparecían, en principio, como figuras ideales para esta protección.



Por tanto, si bien en un primer momento se decía que los programas de ordenador podían asimilarse a obras científicas (y ser objeto de patente), después se entendió que los programas de ordenador tienen las mismas características que las obras literarias en la medida en que se conseguía perseguir las copias ilegales de los mismos.

Ahora bien, los programas de ordenador ¿son realmente obras literarias? ¿Qué problemas tiene proteger los programas de ordenador por los derechos de autor?

Los creadores y titulares de los programas de ordenador quieren proteger al máximo sus derechos, y la protección por derechos de autor no protege las “ideas” inertes en el programa de ordenador (en definitiva, lo más creativo y valioso en la creación de los programas: el algoritmo), pues los derechos de autor sólo limitan la protección a la “expresión”.

Además, tampoco debemos olvidar que los programas de ordenador pretenden un resultado técnico, es decir, son la tecnología para utilizar los ordenadores y no son creados para comunicar información o pensamientos a los seres humanos (como es la esencia última de las obras literarias), sino que están diseñados para comunicarse con las máquinas.

#### **3.3.4. La patentabilidad del software**

A pesar de todo, la presión de las grandes empresas desarrolladoras de software sobre las oficinas de patentes ha hecho que éstas les ha-

yan concedido muchas patentes sobre programas de ordenador en sí mismos considerados.

Hay quien sostiene que uno de los motivos por los que se conceden patentes de software rápidamente es que el hecho de aceptar las solicitudes para luego concederlas supone una buena fuente de ingresos en tasas para la correspondiente oficina de patentes.

De todos modos, no hay que olvidar que, en la práctica, las grandes empresas informáticas han llegado incluso a conseguir muchas concesiones de patentes sobre programas de ordenador que, en realidad, no debieran haber sido concedidas, estrictamente hablando. Así está ocurriendo no sólo en la oficinas de patentes de Estados Unidos, Australia y Japón, sino también más recientemente en las de Europa, en donde existen limitaciones legislativas relativas a la patentabilidad del software más claras.

En los siguientes apartados veremos en qué medida se acepta actualmente la patentabilidad del software en el mundo.

### Patentabilidad del software en Europa

En Europa, tanto por la vía legislativa como jurisprudencial, los programas de ordenador en sí mismos han sido excluidos del ámbito de protección de las patentes, pero se acepta, no obstante, la patentabilidad de un procedimiento completo en el que una parte del mismo sea desarrollada por un programa de ordenador.

En realidad, la exclusión europea de la protección del software en sí mismo por la figura de la patente no es tan sencilla. De hecho, se trata de una excepción y, como tal, no debe mantenerse de forma tan estricta. Precisamente, la cuestión problemática estriba en dónde y cómo definir la frontera de cuándo el software es patentable y cuándo no. Pues el acalorado debate doctrinal sobre la admisión o no de las patentes de software no tiene base en el concepto tan vago de lo que se entiende por *invención*, sino en cuáles deben ser los fundamentos y el objeto de protección de las patentes de software.

La anterior exclusión se limita al caso del programa considerado aisladamente, es decir, cuando el programa de ordenador sea el único elemento integrante de la solicitud de patente. Ahora bien, cuando el objeto reivindicado no es el mero programa, su presencia en la reivindicación no es por sí mismo causa suficiente para negar la patentabilidad si, como luego veremos, el objeto reivindicado produce efectos técnicos. Esto significa que, por ejemplo, máquinas, procesos de fabricación o de control dirigidos por un programa de ordenador deberán ser considerados objetos susceptibles de patentabilidad.



En consecuencia, la protección por patente en Europa no se otorga al programa de ordenador en sí mismo, sino a la combinación de éste con los otros elementos de la invención.

El hecho de que un programa de ordenador se encuentre cargado en el ordenador no aporta nada a la patentabilidad del programa. Sólo si la combinación de éste con el ordenador produce un efecto, un resultado técnico o determina una manera diferente de funcionamiento del ordenador, tal combinación será susceptible de patentabilidad en Europa, pero no el programa en sí.

Pero la diferencia puede no ser siempre clara. Lo cierto es que a menudo puede integrarse un programa (entendido como tal toda secuencia de instrucciones o indicaciones destinadas a ser utilizadas, directa o indirectamente, en un sistema informático para realizar una función o una tarea, o para obtener un resultado determinado, cualquiera que sea su forma de expresión o fijación) dentro del propio ordenador, aparato electrónico o dispositivo de modo que no pueda separarse de éste.

Podemos encontrar ejemplos de esto último en los primeros ordenadores, que incorporaban ya el sistema operativo, en algunos módems (los basados en dispositivos *Digital Signal Processing*) o, especialmente, en el hardware programable tipo FPGA (*File Programmable Gate Array*).

Así pues, en los casos en los que el programa interviene como un elemento más de la invención, se acepta en Europa, en principio, la patentabilidad del programa de ordenador. Si bien, ¿cuándo se entiende que el software está implementado en una invención? ¿Cuándo se está burlando la excepción para conseguir la patentabilidad de un software que, en realidad, no forma parte de ninguna invención patentable, sino que es independiente de dicha invención?



En la actualidad, la mayoría de legislaciones aceptan que el software es susceptible de patentabilidad cuando forma parte o es objeto de una innovación tecnológica mayor y, por tanto, más allá de esos límites, o mejor dicho, por debajo de los mismos, sólo es protegible a través de los derechos de autor o *copyright*.



En este sentido, el propio **Convenio de Munich** establece en su artículo 52.3:

“Lo dispuesto en el párrafo 2 excluye la patentabilidad de los elementos enumerados en el mismo solamente en la medida en que la solicitud de patente europea o la patente europea en sí no se refiera más que a uno de esos elementos considerados como tales.”

Por tanto, la exclusión de la patentabilidad del software se refiere únicamente a los supuestos en los que se reivindica la protección del programa de ordenador por sí sólo, pero no a los procedimientos industriales puestos en práctica con la ayuda de un programa de ordenador.

Quizás el motivo de la exclusión de los programas de ordenador en sí mismos obedezca a que, como los descubrimientos, las teorías científicas, los métodos matemáticos y las presentaciones de información no son de naturaleza técnica y la patentabilidad requiere una aplicación técnica y no ser un método abstracto.





Precisamente, la Ley española de Patentes de 1986 establece en su artículo 4.2 c) que:

“Lo dispuesto en el apartado anterior excluye la patentabilidad de las invenciones mencionadas en el mismo solamente en la medida en que el objeto para el que la patente se solicita comprenda una de ellas.”

Ello se plasma también en las **interpretaciones jurisprudenciales** que se hacen en los casos concretos que se plantea a los jueces.

Así, en el caso francés “Schlumberger” de 1981 se entendió que, dado que la mayoría de las invenciones importantes recientes necesitaban programas de ordenador, excluirlas del ámbito de la patentabilidad supondría resultados aberrantes sobre el plano práctico, llegándose a pensar que, en caso contrario, se reduciría el sistema de patentes a un archivo de la tecnología del siglo XIX. Y la misma tendencia se observa en el caso alemán “Seitenpuffer” de 1991.

En el mismo sentido, la Oficina Europea de Patentes (OEP), influenciada por las directivas de la misma oficina dictadas en 1985 y con el asunto Viacom de 1986 (en que se diferenciaba la protección del programa considerado aisladamente y la del programa integrado en el conjunto de la invención), ha entendido que una reivindicación relativa a un procedimiento técnico realizada bajo el control de un programa no puede ser considerado como concerniente a un programa de ordenador en el sentido del artículo 52.3 del Convenio de Munich, pues es por aplicación del programa que determina la sucesión de etapas del procedimiento como la protección es de hecho concedida y, por consiguiente, tal reivindicación es admisible en virtud del artículo 52.2 y 3 del Convenio de Munich.



En principio, la Oficina Europea de Patentes entiende que un programa de ordenador reivindicado por sí mismo no es susceptible de ser patentado, ni siquiera cuando el programa se carga en un ordenador concre-

to. Pero si el objeto reivindicado aporta una “contribución de carácter técnico” al “estado de la técnica”, la patentabilidad no deberá ser puesta en duda por la simple razón de que el programa de ordenador esté implicado en su puesta en práctica.

La posición anterior es confirmada por la Oficina Europea de Patentes en sus sucesivas decisiones, las cuales deben tomarse en cuenta en la interpretación de las legislaciones europeas.



A raíz de casos concretos (como VIACOM 1987 y el caso IBM 1997), la Oficina Europea de Patentes ha ido ampliando la salvedad del párrafo 3 del artículo 52 del Convenio de Munich, entendiendo que un programa de ordenador es patentable si supone una “contribución técnica” o produce un “efecto técnico”.

Aun así, por la presión de las grandes empresas estadounidenses, una decisión de la Cámara de Recursos de la Oficina Europea de Patentes reconoce que “puede concederse una patente no sólo en el caso de invenciones en las que un programa gestiona, por medio de un ordenador, un proceso industrial o el funcionamiento de parte de una máquina, sino también en los casos en que el programa de ordenador es el único medio, o uno de los medios necesarios, para obtener un efecto técnico dentro del significado indicado anteriormente, en el que, por ejemplo, un efecto técnico de ese tipo se consigue en virtud de una función interna de un ordenador bajo la influencia de dicho programa.

Sin embargo, el anterior criterio no es muy afortunado en cuanto que obliga a hacer una distinción no siempre fácil entre programas de ordenador con una contribución técnica (es decir, los que dirigen procesos y muestran datos de test/proceso, o controlan otros procesos con esos datos) de los programas de ordenador que no tienen una contribución técnica (es decir, los que no procesan datos, sino que controlan el ordenador o implementan algún proceso con el or-

denador). Lo anterior implica que los primeros se benefician de poder ser patentados, mientras que los segundos no.

Si bien en derecho norteamericano puede tener sentido exigir una “contribución técnica” en la medida en que no la establece la ley, no tiene tanto sentido exigirla por parte de la Oficina Europea de Patentes en la medida en que exigir por un lado la susceptibilidad de aplicación industrial y por otro lado la aportación de una “contribución técnica” es duplicar el mismo requisito.

Por otra parte, hemos de decir que la Oficina Europea de Patentes reconoce que tiene en cuenta en el ámbito de patentes (a pesar de no ser en principio aplicables) las disposiciones del Acuerdo sobre Aspectos de los Derechos de Propiedad Intelectual relacionados con el Comercio, adoptado por la Organización Mundial del Comercio en 1994, las cuales son muy amplias en cuanto a las posibilidades de patentabilidad.



Según el art. 27 del Acuerdo sobre Aspectos de los Derechos de Propiedad Intelectual relacionados con el Comercio, adoptado por la Organización Mundial del Comercio en 1994, son patentables:

“todas las invenciones, sean de productos o de procedimientos, en todos los campos de la tecnología, siempre que sean nuevas, entrañen una actividad inventiva y sean susceptibles de aplicación industrial.”

Y se añade que:

“[...] a los efectos del presente artículo, todo miembro podrá considerar que las expresiones ‘actividad inventiva’ y ‘susceptibles de aplicación industrial’ son sinónimos respectivamente de las expresiones ‘no evidentes’ y ‘útiles’.”



La práctica de la Oficina Europea de Patentes y la Comisión Europea vienen reconociendo que es posible patentar

el software presentando las solicitudes redactadas de una forma particular. A partir de ahí, la Oficina Europea de Patentes ha concedido más de 30.000 patentes de software. Esta situación no es muy satisfactoria porque produce inseguridad jurídica y distorsiona el mercado, ya que, además, la aplicación práctica de la normativa de patentes no es igual en toda Europa.

Paralelamente, las correspondientes oficinas de patentes de algunos estados miembros de la Unión Europea, como Gran Bretaña y Alemania, cuya legislación también contemplan la prohibición de patentabilidad para los programas de ordenador, también han seguido la jurisprudencia emanante de la Sala de Recursos de la Oficina Europea de Patentes en la medida en que las oficinas de patentes respectivas a esos estados miembros de la Unión Europea han concedido patentes sobre programas de ordenador que producen un “efecto técnico” o resuelven un “problema técnico”.

En resumen, en Alemania y Gran Bretaña también han entendido que no se puede excluir la patentabilidad de un proceso por el simple hecho de que esté basado en un algoritmo. Fruto de lo anterior, es interesante destacar la sentencia del Tribunal Supremo alemán de 17 de octubre del 2001, la cual exigió una “contribución técnica inventiva” para declarar la existencia de “actividad inventiva”.



En definitiva, las diferentes jurisprudencias nacionales europeas muestran el alcance limitado de la regla de la exclusión de la patentabilidad de los programas de ordenador.

En ellas se observa que dicha limitación debe ceder ante la solicitud de un procedimiento industrial en el que esté presente un programa de ordenador como elemento integrante del mismo. Es decir, cuando el objeto reivindicado no es el mero programa, sino que éste se combina con un objeto material al que controla o se inserta en un proceso de fabricación o de control y produce un resultado de “carácter técnico”,

el procedimiento en el cual está inserto no puede ser privado de patentabilidad. Pero ello no significa que el programa en sí mismo está protegido por patente, pues la patente concedida al procedimiento no permite su monopolio, de tal manera que su utilización en otros procesos quedará libre, pero no en el mismo procedimiento.

Podríamos decir que el criterio para admitir la patentabilidad conjunta de un programa de ordenador dentro de una invención es la petición de protección, no ya sobre el programa en sí mismo, sino en combinación con otros elementos técnicos (ya sean materiales o inmateriales) que implican o dan como resultado un avance tecnológico (es decir, la invención en conjunto). En este sentido, lo que se patenta es el conjunto de medios y funciones, donde los medios son los diferentes dispositivos técnicos que componen la invención (y de los que puede formar parte el software) y las funciones son el fin de los medios (es decir, aquello para lo que sirven los medios).



Pero la norma del Convenio de Munich no contempla directamente los límites. Ello ha propiciado que en Europa se haya ampliado el concepto de lo que es patentable y, bajo este nuevo criterio, se haya aceptado (en realidad, “colado” en la oficina de patentes) incluso la patentabilidad del software en sí mismo.

En conclusión, podemos decir que en Europa, si bien un programa de ordenador en sí mismo no puede ser objeto de patente, cuando se combina con un objeto material (al cual controla) o se inserta en un proceso de fabricación o de control para producir un resultado de carácter técnico, entonces dicho programa sí que goza de la protección que confiere una patente.

Es decir, a grandes trazos, únicamente cuando un programa de ordenador va incluido en un producto o en un proceso industrial de forma indisoluble (en combinación con otros elementos técnicos de los que forma parte y que pueden calificarse en conjunto como una invención), puede ser registrado en la oficina de patentes correspondiente.

Sin embargo, no hemos de olvidar que el software que se incluye en dicha invención sólo quedará protegido dentro de los límites intrín-

secos de la propia patente y, por tanto, no podrá ser objeto de otros fines industriales.

### Patentabilidad del software en Australia

El nuevo test que la oficina de patentes australiana adopta para determinar la patentabilidad de las invenciones relativas al software es el siguiente: verificar si la invención supone la producción de un “resultado económico o comercial útil”. Esta postura es incluso mucho más abierta que la de la Oficina Europea de Patentes, dado que es mucho más difícil determinar qué clase de invenciones deberían ser excluidas, pues la mayoría de este tipo de invenciones tendrán alguna “aplicación comercial”.

#### Ejemplo

En Australia han sido admitidos como patentables, por ejemplo, una imagen mejorada de cara a la distribución de programas de ordenador de gráficos y un algoritmo que supone una operación más eficiente en un ordenador.

### Patentabilidad del software en Estados Unidos

La Oficina de Patentes de Estados Unidos ha sido la primera a nivel mundial en ir aceptando las solicitudes de patentes de software (incluso en sí mismo considerado), marcándose así una diferencia durante los años noventa respecto a Europa. Además, la jurisprudencia de Estados Unidos ha entendido que las patentes de procedimiento no deben ser excluidas para toda aquella invención que, enteramente o en parte, pueda ser puesta en ejecución con la ayuda de un programa de ordenador.

#### Ejemplo

Así resulta del caso “Diamond frente a Diehr” de 1981 el cual, a diferencia del caso también citado “Parker frente a Flook” de 1978, no se buscaba solamente patentar el algoritmo o fórmula matemática, sino un procedimiento novedoso de vulcanización de caucho sintético para moldearlo, donde no se pretendía reservarse el uso de la ecuación matemática conocida que se usaba en la patente.

En Estados Unidos, a finales de los años sesenta, se entendía que los programas podían ser patentables en la medida en que “mejoraban un proceso de análisis sin la intervención humana” o cuando se conseguía una “mejora útil y nueva”.

Posteriormente, en los años setenta se entendió que los “algoritmos no constituyen materia patentable” al estar situados en el plano de las ideas y, en definitiva, de los procesos y conceptos mentales o intelectuales abstractos que no son patentables. Por contra, si el programa informático suponía una “aplicación inventiva” de un “fenómeno de la naturaleza” (y no su simple descubrimiento), entonces sí que era patentable si concurrían los requisitos de “novedad”, de “utilidad” y de no tratarse de un mero algoritmo matemático.

Pero desde el caso “Diamond frente a Diehr”, en 1981, en Estados Unidos se entendió que la patente era extensiva al procedimiento en el cual participan los programas de ordenador, siempre que el procedimiento en el que se basa la solicitud de patente no esté exclusivamente constituido por un algoritmo. En este último caso se viene a definir la patentabilidad del software cuando se trata de una solicitud de proceso que incluye una fórmula matemática como un paso más, siempre que no se busque la protección particular de esa fórmula matemática y, por tanto, cuando la fórmula matemática “forma parte del proceso completo” el cual es, en realidad, el objeto de la patente.

A diferencia de lo que hemos visto con la doctrina de la Sala de Recursos de la Oficina Europea de Patentes, en Estados Unidos no es necesario que la invención patentable aporte una “contribución técnica” al “estado de la técnica” para que sea patentable. En Estados Unidos basta con que la invención pertenezca al “ámbito de la técnica” para que sea patentable (no es necesaria una “contribución técnica”). Por tanto, el simple hecho de que una invención utilice un ordenador o un programa de ordenador es suficiente para que entre en el “ámbito de la técnica”, a condición de que produzca un “resultado tangible, útil y concreto”. Ello permite, además, la patentabilidad no sólo del software, sino también de los métodos de negocio, lo cual en Europa está claramente excluido.

Por tanto, la Ley de Patentes de los Estados Unidos y la jurisprudencia permiten en principio la doble protección del software como

propiedad intelectual (bajo el concepto anglosajón de *derechos de autor*, es decir: *copyright*) y como propiedad industrial (patentes). Y ello tanto si se trata de una patente de software en sí mismo considerado, como si se trata además de una invención implementada por ordenador.

Esto está ocasionando un terrible caos en la Oficina de Patentes, que no cuenta con personal suficientemente cualificado para analizar rápidamente la novedad y la altura inventiva sobre el programa, y crea, además, inseguridad en los mercados financieros.

Todo empezó con la práctica de la Oficina de Patentes de los Estados Unidos, que iba aceptando las solicitudes de patentes de software, marcando así una diferencia durante los años noventa respecto a Europa. La Oficina de Patentes de Estados Unidos (USPTO) preparó en 1995 y llegó a crear una guía para los examinadores que, si bien no tiene el rango de ley, sí que es tomada en cuenta por los tribunales. De hecho, el Tribunal Supremo admitió en 1999 la posibilidad de que un algoritmo matemático pudiera ser objeto de patente. Anualmente se conceden unas 20.000 patentes de software.

Si bien la ley de patentes de Estados Unidos requiere que las invenciones sean originales y no evidentes, en la práctica, la Oficina de Patentes no exige este requisito, en parte quizás porque con más de 20.000 patentes (sólo en 1999), dicha oficina no tiene capacidad para examinar cada petición críticamente.

Como resultado de ello, muchos han sostenido que la trivialidad de algunas de las patentes de software que se conceden en Estados Unidos se debe a la falta de recursos humanos, personal cualificado y sistemas de búsqueda en las correspondientes oficinas de patentes. De hecho, no fue hasta 1994 cuando la USPTO contrató a examinadores titulados en informática. La oficina suele dedicar un promedio de 17 horas por patente, lo cual no es suficiente para pensar detenidamente en la patentabilidad de un programa. La concesión de una patente de software por error no sorprende, pues, con este sistema.

Sin embargo, tampoco hay que olvidar que, si no es imposible, una búsqueda de anterioridades resulta francamente complicada



en relación con el software. Especialmente si tenemos en cuenta que la manera de realizar un programa de ordenador no es más que una mera forma de escritura (así debieran entenderse las instrucciones que da el programa a la máquina), tal como encontramos en la manera de escribir música, matemáticas, artículos científicos, etc., y es impensable que ningún humano pueda tener conocimiento absoluto de todas las obras literarias escritas en el mundo.

Además, en Estados Unidos se entiende que si el procedimiento de examen de la patente solicitada no es demasiado profundo o de calidad (en Estados Unidos no hay fase de observaciones, oposiciones, etc.), se consigue que el coste de la solicitud de la patente sea ágil y barato, y permite su acceso a la pequeña y mediana empresa (a diferencia de Europa, donde el procedimiento de solicitud de patente es mucho más costoso).

Si bien a veces se defiende la facilidad con que una patente de software puede ser obtenida en Estados Unidos con el argumento de que posteriormente puede ser recurrida en los tribunales y revisada o anulada subsecuentemente, el solo hecho de que alguien encuentre la misma idea de la patente en el estado de la técnica anterior, invalida la patente.

Sin embargo, creemos que no parece conveniente tener un sistema legal donde las empresas o las personas tengan que demandar para rechazar o invalidar una patente concedida incorrectamente, aunque el coste para ellos sea barato (como ocurre en Dinamarca, en donde es posible que una patente se reevalúe por el organismo competente por unas pocas coronas cuando no es original). En este caso, nos encontraríamos con un sistema de patentes débiles, lo cual potencia una litigiosidad asegurada, entendemos que poco conveniente para aquellas empresas que no se la pueden costear.

#### Ejemplo

Así ocurrió en España cuando las patentes se concedían sin examen de novedad o sin examen del estado de la técnica; y así ocurre con las marcas comunitarias, que se conceden de una manera mucho más fácil que, por ejemplo, una marca española.

### Patentabilidad del software en otros países: Japón y Corea

En **Japón**, y antes que en Europa, ha pasado lo mismo que en Estados Unidos, donde cada año se registran unas 35.000 patentes de software (incluso aisladamente). En Japón existe una doctrina que se ha interpretado tradicionalmente de forma similar a la de la “contribución técnica” de la Oficina Europea de Patentes. En este sentido, en Japón se considera que la invención debe consistir en una creación muy avanzada de conceptos técnicos mediante la cual se aplica una ley natural.

En **Corea** también conceden patentes de software.



En definitiva, varios son los países que actualmente aceptan la patentabilidad del software y de los algoritmos matemáticos, con la consecuencia de una larga lista de patentes triviales concedidas.

#### 3.3.5. La Propuesta de Directiva del Parlamento Europeo y del Consejo

Ya hemos visto antes que los programas informáticos en la Unión Europea están excluidos, como tales, de la protección de la propiedad industrial mediante patentes, aunque lo cierto es que la interpretación que se da al Convenio de Munich sí que permite patentar un producto que contenga un programa informático determinado, siempre que dicho producto sea de “carácter técnico” y cumpla los requisitos para ser calificado como “nuevo” e “inventivo”. Incluso existe jurisprudencia a nivel europeo que permite patentar los programas de ordenador en determinadas circunstancias.

Sin embargo, en Europa se ha abierto un debate muy polémico en torno a la patentabilidad del software cuando éste está implementado en una invención, fundamentalmente por la presión de las grandes empresas informáticas de Estados Unidos que han “colado” muchas patentes de software como programa de ordenador aislado y pretenden continuar haciéndolo, si bien bajo las garantías del am-

paro legal. En el fondo, lo que se debate es qué debe entenderse por *invención* o, dicho de otro modo, a qué debe quedar circunscrita la limitación de patentabilidad sobre el software. Realmente, la labor de delimitar una frontera clara no es una tarea fácil.

La Oficina de Patentes Europea, especialmente desde 1997 y muy a pesar de lo dictaminado por el artículo 52 del Convenio de Munich, siguió a Estados Unidos y a Japón en la práctica de aceptar las patentes de software. De las patentes de software que ha concedido la Oficina Europea de Patentes, el 75% pertenecen a empresas de Estados Unidos, precisamente por el interés y presión de éstas en Europa, amén del desconocimiento de las empresas europeas para utilizar las patentes como medio para proteger sus productos (o, en su caso, porque las empresas europeas que lo conocen consideran las patentes complejas, caras y difíciles de aplicar para las pequeñas empresas).

Dada la extrema rapidez del desarrollo tecnológico en el campo de la tecnología de la información, parece muy dudoso que la Oficina de Patentes Europea sea capaz de hacerlo mejor que su compañera en Estados Unidos. De todos modos, sea o no posible establecer un sistema más crítico en Europa, uno debe temer que las consecuencias de la armonización con Estados Unidos se traducirán en que la jungla de patentes software triviales inundará Europa.

La jurisprudencia nacional europea existente hasta la fecha en el ámbito de las invenciones implementadas en ordenador se ha desarrollado principalmente en los tribunales de dos estados miembros: Alemania y Gran Bretaña. Aunque, curiosamente, han llegado a adoptar decisiones divergentes sobre cuestiones importantes relacionadas con los requisitos de obtención de una patente (es decir, en relación con la definición de “materia patentable”).

Lo anterior indica que, en ausencia de medidas de armonización, los tribunales de los demás estados miembros podrían llegar a adoptar soluciones totalmente discrepantes cuando deban pronunciarse sobre estas cuestiones. El problema reside, pues, en la confusión que genera la variación con que los tribunales nacionales europeos aplican las reglas de la Oficina Europea de Patentes.

#### Nota

El artículo 52 del Convenio de Munich ha sido prácticamente ignorado por la Oficina Europea de Patentes, en la medida en que ha hecho una interpretación muy distinta a la que se desprende de su tenor literal.



Los licenciatarios y el público en general, usuarios potenciales de objetos patentables, no tienen en la actualidad la certeza de que, en caso de litigio, vayan a confirmarse las patentes concedidas en este ámbito.

En su comunicación de febrero de 1999 sobre la protección mediante patente, la Comisión ya identificó la necesidad de una acción legislativa en este ámbito. Con objeto de elaborar una directiva sobre el tema, entre el 19 de octubre y el 15 de diciembre del 2000, la Comisión abrió una amplia consulta pública, cuyos resultados han puesto de manifiesto la falta de consenso respecto a las normas que deberían regir las patentes de software: ¿deberían ser restrictivas, como hasta ahora?, ¿o deberían seguir la línea permisiva de la legislación estadounidense?

La Comisión trabaja desde entonces en el análisis de estas posiciones. En febrero del 2002 la Comisión presentó una **Propuesta de Directiva sobre la Patentabilidad de las Invenciones Implementadas en Ordenador** [COM (2002) 92].

La Comisión justifica la Propuesta de Directiva en buscar una acción concertada en este ámbito, para reforzar la posición competitiva de la industria del software europeo frente a Estados Unidos y Japón, y para que éstos no dominen el mercado (especialmente Internet). Pues, en caso contrario, las empresas europeas se encontrarían en desventaja, fundamentalmente frente a las empresas de Estados Unidos, que ya tienen asumido que los programas de ordenador son patentables.

En principio, la Propuesta de Directiva pretende poner freno a las tendencias europeas, especialmente de la Oficina de Patentes Europea, de los últimos años, a admitir patentes de software y acabar con la incertidumbre jurídica que supone resoluciones jurisprudenciales dispares dentro de la Unión Europea, especialmente en contra o no de la prohibición de la patentabilidad de software en sus legislaciones internas.

Sin embargo, en las primeras redacciones y enmiendas al texto legal, parece que en realidad se pretende encubrir una protección vía pa-

tente, fundamentalmente por parte de las empresas informáticas más importantes de Estados Unidos, respecto al programa de ordenador en sí mismo considerado, y no como si formara parte de un conjunto, a fin de ceder a los intereses de las grandes empresas informáticas estadounidenses. Y ello es bastante preocupante, no ya sólo porque la mayoría de patentes hasta ahora concedidas en la Oficina Europea de Patentes está en manos de empresas de Estados Unidos, sino, entre otras cosas, por lo peligroso que supone admitir las patentes de software frente a los principios en los que se basa el software libre.

**Nota**

La razón por la que la mayoría de patentes concedidas en la Oficina Europea de Patentes está en manos de empresas de Estados Unidos es el desconocimiento de las empresas europeas para utilizar las patentes como medio de protección de sus productos o, en su caso, porque las empresas europeas que lo conocen consideran las patentes complejas, caras y difíciles de aplicar para las pequeñas empresas.

La Propuesta de Directiva parte de una protección mixta en la que los programas de ordenador tanto pueden protegerse por los derechos de autor, como por las patentes. La propuesta entiende que ambas protecciones son complementarias y, por tanto, no excluyentes. Trata de armonizar las distintas legislaciones europeas en este tema y, en consecuencia, las resoluciones que puedan dictar los distintos órdenes judiciales.

Fundamentalmente, dicha propuesta trata de luchar contra el gran número de patentes triviales que se han “colado” en la Oficina Europea de Patentes y en algunas “oficinas de patentes” correspondientes a los países miembros de la Unión Europea, pues la Oficina Europea de Patentes ha evolucionado, a través de una serie de casos resueltos, en una dirección que, para algunos, representa una liberalización de los criterios de patentabilidad, lo cual ha dado lugar a quejas en relación con las invenciones nimias o que hacen una aportación insuficiente en relación con el estado de la técnica, y por el hecho de que el examen de estas cuestiones tiende a quedar relegado

por la cuestión bastante estéril y filosófica de si la presunta invención aporta o no un “efecto técnico”.

Muchas de las 30.000 patentes relacionadas con programas informáticos ya concedidas por la Oficina Europea de Patentes están en manos no europeas. Consentir esto supondría dar ventaja competitiva a dichas empresas, las cuales ya están acostumbradas a solicitar dicha protección por patente en sus respectivos países.

En la actualidad continúa habiendo una gran polémica dados los intereses en juego de los sectores confrontados. Parece que el objeto de protección que finalmente se propondrá seguirá teniendo un alcance limitado, en la medida en que la “invención” tenga una contribución técnica que sea evidente para un experto en la materia. De todos modos, la cuestión aún puede dar algún giro inesperado.

La propuesta insiste en que la protección por patente debe incluir un programa de ordenador que esté aplicado a un ordenador, pero no un programa de ordenador creado independientemente en la medida en que éste estaría protegido por el derecho de autor. En este sentido, quedaría incluido el software integrado en electrodomésticos inteligentes, teléfonos móviles, dispositivos de control de motores, máquinas herramienta, aparatos de todo tipo y sistemas informáticos complejos.

Las distintas enmiendas a la propuesta han tratado de aclarar las ambigüedades al intentar definir qué son las “invenciones implementadas en ordenador” y qué es una “invención técnica”. Asimismo, tratan de defender la “interoperabilidad” para asegurar los “estándares” y “protocolos” y el acceso a la sociedad de la información, defender la “libertad de publicación” y evitar que se utilice el sistema de patentes contra su propia naturaleza para impedir la difusión de conocimientos, monopolizar medios de expresión y mantener todos los requisitos legales de patentabilidad.

En definitiva, se trata de evitar que las pequeñas empresas de software tengan que hacer frente a la concesión inadecuada de patentes en el caso de patentes poco claras y obvias.

Finalmente, la propuesta descarta claramente que los modelos de negocio o métodos comerciales puedan protegerse como patentes, tal y como ocurre en Estados Unidos.

**Ejemplo**

Un ejemplo de ello es el *one-click* de Amazon, patentado en la Oficina de Patentes de Estados Unidos (USPTO). De todos modos, no hay que olvidar que dicha patente tiene una “prima” en Europa, pues las reivindicaciones de la patente de la Oficina Europea de Patentes son diferentes a las de la USPTO.



En realidad, las verdaderas cuestiones sobre la patentabilidad del software no son jurídicas, sino económicas y de oportunidad.

### 3.3.6. Ventajas y desventajas para proteger el software por las patentes

En este apartado veremos las ventajas y los inconvenientes que supone proteger el software como patente desde el punto de vista de las **grandes empresas informáticas**.

En el bloque siguiente de esta unidad veremos las ventajas y desventajas que supone proteger el software como patente desde el punto de vista del **usuario final** y, especialmente, del **programador de software libre**.

**Nota**

Ver el apartado 3.4 y siguientes.

#### **Ventajas**

Las grandes empresas informáticas están interesadas en que el software sea protegido por patente por los siguientes motivos:

- a) Las patentes protegen las “ideas” que hay detrás del programa de ordenador. En cambio, los derechos de autor y *copyright* sólo protegen la “expresión”. En la medida en que la patente protege la “idea”, se puede impedir que otra persona pudiera crear un programa similar (aunque lo hiciera con código fuente o algoritmos distintos), si el programa de ordenador fuera fruto de la misma idea. En este sentido, el monopolio que ofrece la

patente es mucho mayor que el que ofrece el derecho de autor y *copyright*.

- b) El empresario consigue ser titular de una patente de una forma más directa. Normalmente, en el seno de patentes, las invenciones de los trabajadores pertenecen al empresario y los derechos a favor del trabajador se restringen considerablemente.

#### Ejemplo

Por ejemplo, el trabajador no tiene derecho a remuneración complementaria alguna, excepto cuando la aportación personal exceda de forma evidente del contenido explícito del contrato de trabajo o de la prestación de servicios.

Por otra parte, la información adquirida por el trabajador en el seno de la empresa y gracias a la cual se ha llegado a la invención, también será propiedad de la empresa, la cual podrá asumir la titularidad de la invención o reservarse un derecho de utilización sobre la misma.

- c) En derecho de patentes no existen los derechos morales propiamente dichos. Sin perjuicio de que en algunas legislaciones de patentes se reconozca al inventor un derecho personalísimo (“derecho moral” del inventor) por el cual se le vincula de forma indisoluble como autor de la invención, podemos decir de forma general que en las patentes se evita la posibilidad de que el autor del programa (detrás del cual siempre habrá una persona física) pueda ejercitar las prerrogativas específicas de los derechos morales propios de los sistemas de derecho de autor (entre ellas, el de divulgación, lo cual impediría la divulgación del programa una vez creado).



Dado que en el derecho de autor (no es así de claro bajo el *copyright*), los derechos morales son irrenunciables e intransmisibles, la empresa que comercializa el programa está siempre expuesta a que el programa no pueda ser comercializado o sea retirado del comercio.



Si bien las empresas suelen hacer firmar cláusulas a sus trabajadores conforme el trabajador no ejercitará sus derechos morales bajo apercibimiento de una cantidad económica, dada la posible nulidad de este tipo de cláusulas en los países bajo el sistema de “derecho de autor” en la medida en que los derechos morales son irrenunciables, la empresa conseguirá en su caso una indemnización, pero no conseguirá asegurar la comercialización del software, que puede ser impedida en cualquier momento por el autor.

- d) La mayor facilidad para transmitir los derechos de explotación sobre una patente o para conceder una licencia. Así ocurre en relación con la exclusividad y el derecho a otorgar sublicencias.

### Desventajas

Pero no todo son ventajas; las grandes empresas informáticas también ven inconvenientes en que el software sea protegido como patente:

- a) El periodo de protección es más corto. Las patentes dan una protección por 20 años, mientras que el derecho de autor y *copyright* dan una protección durante muchos más años; en particular, durante 70 años en Europa, y ampliables hasta 120 años en Estados Unidos. Pero esta desventaja no es tal en la medida en que en la práctica comercial los programas de ordenador tienen pocos años de vida, pues enseguida quedan obsoletos, y las duraciones que estipulan tanto las patentes como los derechos de autor resultan extremadamente largas.
- b) Coste para obtener una patente. Las patentes suponen unos costes de honorarios de abogados, ingenieros y tasas en las correspondientes oficinas de patentes bastante elevados. Sin embargo, esta desventaja puede ser incluso una ventaja para las grandes firmas respecto a los pequeños desarrolladores de software que, por no contar con los recursos necesarios, no pueden afrontar los costes que supone obtener una patente.

- c) Tiempo para obtener una patente. Como hemos visto, una patente sólo confiere los derechos tras un largo procedimiento (puede llegar fácilmente a los tres años, según el país en que se intente obtener la protección por patente), si bien también es cierto que se dan una serie de protecciones desde el momento de la solicitud y los efectos son retroactivos al momento de la solicitud de la patente. Sin embargo, los efectos y protecciones están siempre pendientes de la concesión final de la patente. En cambio, los derechos de autor protegen al programa de ordenador desde el mismo momento de su creación, sin necesidad de registro alguno.
- d) El software objeto de patente sólo quedará protegido dentro de los límites intrínsecos de la propia patente y no podrá ser objeto de otros fines industriales. En cambio, bajo el derecho de autor y *copyright*, el software quedaría protegido para cualesquiera finalidades.

### 3.4. Patentes de software y software libre

El modelo de código abierto resulta muy interesante incluso desde un punto de vista del mercado, pues es una forma clave para luchar contra los monopolios establecidos (Linux frente a Windows, Mozilla frente a MSIE, OpenOffice frente a MS Office, Java frente a los entornos de desarrollo Microsoft), abaratar los costes a las instituciones públicas (cuya misión es dar un servicio público a las varias administraciones públicas de distintos países que apuestan por el software libre) que, de otra manera, serían totalmente dependientes del arbitrario sistema de licencias actual de Microsoft.

Sin embargo, si se aprueba en Europa un sistema de patentes, o si se consolida en el resto del mundo y “se aplica” en el sentido de tener patentes de software (software que, en realidad, no forman parte de una invención mayor, sino que son independientes), este mundo de código abierto podría llegar a desaparecer en la medida en que la protección conferida por la figura de las patentes se utilice mal o de forma abusiva. Pero lo cierto es que la posibilidad de poder patentar el software por sí mismo considerado ha ido creciendo progresivamente a lo largo de los últimos años y es una realidad fáctica. De ahí la gran polémica, no sólo en Estados Unidos, sino también a escala internacional.

#### Nota

Se entiende aquí que, en realidad, el software objeto de la patente no forma parte de una invención mayor, sino que es independiente.

### 3.4.1. Las patentes de software y el secreto del código fuente

Hemos visto que una patente es un monopolio limitado en el tiempo que un gobierno otorga a un inventor por una invención que sea básicamente nueva, no obvia y susceptible de aplicación industrial. Durante 20 años nadie puede usar comercialmente la invención sin el permiso del poseedor de la patente.

Tradicionalmente, las patentes se otorgan a la persona que ha creado una invención novedosa, con altura inventiva y de la cual puede sacarse provecho en el ámbito industrial. Pero en este modelo no deberían entrar los descubrimientos, las teorías científicas o fórmulas matemáticas, en la medida en que se consideran de la naturaleza o propiedad de todos y cuyas creaciones derivadas ya son protegidas mediante las diversas legislaciones relativas al derecho de autor o *copyright*.

Aquí entra en juego el software, que en realidad no es más que un conjunto de instrucciones abstractas ejecutadas por el dispositivo lógico universal: el ordenador. Las patentes aplicadas al software podrían llegar, al menos de modo indirecto, a monopolizar formas y métodos abstractos para conseguir determinadas tareas o resultados.

Sin embargo, en el mundo del software hay algo que protege la innovación y que hace que no sea comparable con otros tipos de industrias: el **secreto del código fuente**.

#### Ejemplo

Cuando una empresa química o farmacéutica desarrolla una molécula o sustancia nueva para un fin, la innovación no suele estar en el proceso de fabricación, sino en la molécula en sí. Una vez descubierta, la molécula puede ser fácilmente reproducible por otros, de modo que lo que se busca proteger con la patente es la inversión efectuada para su descubrimiento. Si la molécula fuera secreta (no es el caso, porque basta un microscopio electrónico o un análisis químico para que deje de serlo) no habría tanta necesidad de patente porque los competidores tendrían que recorrer un camino igual de complejo para llegar a ella.

En cambio, en el mundo del software la molécula “es secreta” y sólo se conoce su efecto (su funcionamiento), gracias a la conversión del código fuente en un código de más bajo nivel (el código objeto, y finalmente, el código máquina).

De hecho, los derechos de autor y *copyright* podrían resultar una protección jurídica suficiente para las invenciones que puedan ocultarse en el software interno. Si el código fuente del programa se mantiene en secreto, será difícil desvelar dichas invenciones a partir del código objeto. Sin embargo, podría objetarse que la protección de los derechos de autor del nivel del código objeto puede constituir un impedimento para lograr una prueba de violación de patentes en algunos casos, pues las excepciones a la descompilación sólo se aplican si la reproducción del programa tiene por objeto la interoperabilidad.

Por otro lado, los **secretos industriales** también gozan de protección jurídica a través de varios regímenes como el derecho criminal. La opción del secreto industrial presenta técnicamente múltiples ventajas, pero tienen el inconveniente de que se pierde cuando se quiebra. El riesgo de pérdida de un secreto industrial es particularmente elevado si se concede una licencia.

En cambio, las **patentes** permiten conservar en secreto el “código fuente”, con la excepción de alguna sentencia en sentido contrario pronunciada en Estados Unidos.



En la práctica, los redactores de los documentos de las patentes de software no suelen incluir una descripción del algoritmo, ni tan siquiera una línea del código fuente, sino algún subterfugio que les permita “colar” su patente por los “agujeros” de la oficina de patentes. Y, además, de este modo consiguen proteger con la patente la idea que hay detrás del algoritmo en cuestión. Este proceder asegura la existencia de las malas patentes.

#### Nota

Ver el apartado 3.6, dedicado a los secretos industriales.

**Ejemplo**

Véase, por ejemplo, cómo está reivindicado el programa de ordenador en las patentes estadounidenses USPO 5132992 y 6144702 y la muy similar patente europea EP 0566662 (incluye España), de Acacia Media Technologies, relativas a la transmisión y recepción de ficheros de audio y vídeo comprimido sobre Internet:

<http://swpat.ffii.org/patents/txt/ep/0566/662/>

<http://l2.espacenet.com/espacenet/bnsviewer?CY=ep&LG=en&DB=EPD&PN=EP0566662&ID=EP+++0566662B1+I+>

Ya hemos visto que en la práctica en el documento de patente no se suele incluir el código fuente ni el *know-how* necesario. Sin embargo, en España nos encontramos con que el artículo 76 de la Ley de Patentes dispone, salvo pacto en contrario, que quien transmita una “solicitud de patente” o una “patente ya concedida” o conceda una “licencia sobre las mismas” estará obligado a poner a disposición del adquirente o del licenciataria los conocimientos técnicos que posea y que resulten necesarios para proceder a una adecuada explotación de la invención. Sin embargo, también dispone que el adquirente o licenciataria a quien se comuniquen conocimientos secretos estará obligado a adoptar las medidas necesarias para evitar su divulgación.

Este artículo refleja la pura realidad, ya que los titulares de tecnología patentada no dan a conocer en la descripción de la patente todos los datos y conocimientos que permiten explotar la invención o la explotación más rentable de dicha invención. Y ello es perfectamente lícito, puesto que para que la invención sea patentable no se requiere que sea “explotable”, sino tan sólo “ejecutable”.

El artículo 76 de la Ley de Patentes española pretende subsanar la posición de inferioridad del adquirente o licenciataria en la medida en que obliga al titular de la patente a poner a disposición del primero los conocimientos técnicos que posea el titular y que resulten necesarios para poder proceder a una adecuada explotación de la invención.

En consecuencia, el titular/licenciante no debe poner a disposición todos los conocimientos que están en su poder, ni tampoco los que hagan la explotación más rentable, sino sólo los que hagan que la explotación sea “adecuada”.

Los conocimientos técnicos pueden tener carácter secreto o no. En el caso de que no lo tengan, la obligación del titular/licenciante consiste en la prestación de asistencia técnica (retribuida o no). En el caso de que los conocimientos técnicos sean secretos, nos hallaremos ante la obligación de comunicar el *know-how* (el carácter reservado del *know-how* que supone una ventaja competitiva justifican su retribución).

Evidentemente, la obligación se limita a la comunicación de conocimientos técnicos que posea el titular/licenciante en el momento de celebrar el contrato, y no a los puedan obtenerse después. Éstos deben formar parte, en su caso, de una obligación de comunicar mejoras. Tampoco parece que esta obligación comprenda los conocimientos que corresponden a terceros, dado que en caso contrario supondría infringir los pactos de confidencialidad con ese tercero.

El problema de las “malas patentes” (o patentes obvias) quedaría resuelto si al patentar el software se obligara a registrar los algoritmos y las líneas de código fuente del programa de ordenador en la oficina de patentes. Aunque es cierto que, de este modo, muchas empresas utilizarían la vía alternativa de los secretos comerciales (como hizo Coca-Cola con su famosa fórmula) para evitar patentar su software y mantener su código fuente en secreto. Y otra vez iríamos contra la verdadera pretensión de las patentes, es decir, promover al público el conocimiento de las invenciones.

En realidad, no debiera enviarse sólo un CD con el código objeto del programa de ordenador, sino también los documentos de diseño, los diagramas de flujos, las especificaciones de interfaces, y especialmente, el código fuente. Al fin y al cabo, el código fuente es la descripción más concisa posible de cómo construir el producto final. Pero ¿estarían dispuestas a ello las grandes empresas desarrolladoras de software?

### 3.4.2. Elementos patentables del software

Paradójicamente, es la versión binaria del programa (el código máquina, compuesto de 1 y 0) la que se considera “técnica” según los abogados que trabajan en patentes y, por tanto, susceptible de patentabilidad.



El **código máquina** es lo patentable, mientras que el **código fuente** de ese mismo programa de ordenador no lo es. Ello es así aun cuando la versión binaria se produce por mediación de un proceso trivial automatizado que siempre genera el mismo resultado.

Es igualmente paradójico que los programas que se ejecutan directamente en base a su código fuente no sean patentables. Las anteriores dicotomías se traducen precisamente en las dos posturas siguientes, contrapuestas en la práctica:

- 1) Algunos desarrolladores de paquetes de código abierto establecen mecanismos de conmutación (*switches*) o parches (*patches*) que eliminan o desactivan aquellos fragmentos de código que puedan infringir una patente de software según el país donde se utilice el programa de ordenador.
- 2) Otros desarrolladores de paquetes de código abierto entienden que el código fuente no es una versión ejecutable, sino una mera descripción de cómo debe actuar el ordenador, de modo que no está cubierto por la patente, ni siquiera en los países en donde el software es objeto de patente y, por tanto, no eliminan ni desactivan aquellos fragmentos de código susceptibles de infringir una patente.

Esta última tesis aún está pendiente de ser probada en los tribunales de justicia y, por otra parte, gran parte de la doctrina está en desacuerdo. Sin embargo, sigue dejando abierta la problemática para el usuario final que necesita el programa ejecutable, ya no sólo para la entera industria de software, sino especialmente para los desarrolladores del software libre.

### 3.4.3. Las patentes de software y el software libre: contradicciones y peligros

Las patentes de software no son sólo una amenaza para la industria de software en general por la incertidumbre y necesidad de entrar en el juego de las licencias cruzadas, sino especialmente para los desarrolladores de software libre, por los principios en los que se basa el software libre. Consideremos con detalle los siguientes puntos:

- 1) La publicación y disponibilidad pública del código fuente del software libre permite a cualquier empresa que ostente una patente de software analizar fácilmente si el código fuente utilizado en el software libre vulnera o no su patente.

Es mucho más fácil hacer un estudio de patentabilidad y determinar a partir del código fuente las ideas que hay detrás del programa y ver si alguna de las técnicas que se usan en los programas infringe o no una patente. En la medida en que el software libre contendrá seguramente miles de procesos elementales, muchos de los cuales pueden estar patentados, es casi seguro que el software libre infringirá alguna patente de terceros sin tener conocimiento de ello.

#### Ejemplo

Precisamente la excepción relativa a la descompilación para los casos de interoperabilidad contemplada en los derechos de autor y *copyright* se establece para poder determinar las “ideas” que subyacen al programa de ordenador en ciertos casos limitados legalmente.

La disponibilidad pública del código fuente en que se basa el software libre facilita la búsqueda de violaciones de patentes y lo vuelve más vulnerable a posibles demandas judiciales e incluso a querellas criminales. Esta vulnerabilidad (por la exposición pública del código fuente) puede determinar que las empresas que utilizan código libre prefieran no publicarlo, de modo que la patentabilidad del software desvirtúa el principio de publicidad y apertura en el que se basa el software libre.



Para evitar esta contradicción, hay quien aboga por dar un privilegio de inmunidad por infracción de patente al software que conlleve la publicación de su código fuente, si bien la explotación debería quedar condicionada a las limitaciones del titular de la patente y, por tanto, sujeta a la correspondiente licencia.

- 2) El hecho de que en el software libre no haya una única empresa titular de todos los derechos dificulta los mecanismos para defenderse en caso de litigios por infracción de otras patentes, licencias cruzadas o pago de *royalties*.

Asimismo, la construcción tan descentralizada del software libre (por múltiples aportaciones) complica el establecimiento de las responsabilidades de cada desarrollador en la cadena y la evaluación de los riesgos legales de las diversas contribuciones.

- 3) Otorgar una patente al software puede ser peligroso en la medida en que se patentan “ideas”. Algunas, imposibles de considerar como inventos, ya que se basan en conceptos descubiertos con los años en distintos ámbitos del conocimiento, y que ahora se aplican al mundo informático.
- 4) La innovación informática no se produce a grandes saltos a partir de inventos geniales, sino que es fundamentalmente incremental.

Cualquier aplicación contiene muchísimas pequeñas técnicas y prácticas que, si se patentasen, impondrían un lastre insoportable al desarrollo y a la comercialización del software por el coste de comprobar la patente de cada detalle. La cantidad de patentes necesarias para producir un solo producto puede ser del orden de miles. En cambio, en otras áreas de desarrollo tecnológico, en que la práctica de la patentabilidad está justificada, la cantidad de patentes por producto comercializable suele ser bastante inferior (éste es, por ejemplo, el caso de la industria farmacéutica, en que para una droga o proceso se aplica generalmente una sola patente).

En el caso de las patentes de software, detectar si un producto contiene código que está sujeto a alguna patente suele tener un

coste muchas veces superior al de la creación del código afectado, por dos razones:

- a) La gran cantidad de patentes que existen en los países en que se aceptan las patentes de software.
- b) El lenguaje en el que están descritos los procesos protegidos, que se alejan mucho del lenguaje tradicional usado por los programadores y requiere largas interacciones entre abogados y programadores para que ambas partes comprendan qué protege realmente la patente en cuestión.

Incluso si se pudiera hacer un estudio acabado de la existencia de potenciales problemas con patentes de software para un producto, sería imposible garantizar que no sea necesario licenciar alguna patente.

Por un lado, es difícil efectuar búsquedas extensas y exhaustivas por la cantidad de potenciales patentes involucradas, y por otro lado, es posible que después de terminado un producto, otra persona o empresa que haya desarrollado ese método antes, solicite su patente. Este hecho causa una incertidumbre que no existe con otras formas de protección de bienes inmateriales como los derechos de autor.

#### Nota

En la literatura sobre el tema se habla de campos minados en los cuales se mueve el desarrollo del software.

#### Ejemplo

Un ejemplo de esto es lo ocurrido con Unix Compress. El programa fue creado en 1984, y en 1985 se otorgó una patente sobre el algoritmo LZW. Con esto, el Unix Compress se volvió ilegal de un día para otro hasta que existió una licencia que permitía su uso.

La práctica de usar una base de datos de búsqueda de anterioridades de patentes como medio de información dentro del proceso de desarrollo de un producto relativo a la informática no sólo no existe, sino que no tiene sentido instaurarla por su ineficiencia inherente.

Todo ello afecta al desarrollo del software libre en la medida en que perjudica la capacidad de innovación y desarrollo de los profesionales y las empresas del sector del software libre. Patentar las modificaciones anteriores y las mejoras es abortar la innovación.

- 5) Las patentes de software amenazan la importancia creciente que se da a los estándares de software (formatos, protocolos, interfaces de las aplicaciones informáticas, etc.), en la medida en que pueden obstaculizarlos.

La necesidad de hallar lenguajes comunes para los estándares es cada día más acuciante. Otorgar un monopolio sobre uno de estos lenguajes concedería a su creador un absoluto control sobre los productos de la competencia e impediría la interoperabilidad de los programas de ordenador en la medida en que ésta pudiera infringir la patente, amén de atentar contra la libertad de difusión del conocimiento o *know-how* que subyace en cualquier invención (al quedar monopolizados los medios de expresión).

Tampoco hay que olvidar que en la actualidad se trata de potenciar el acceso a la sociedad de la información y que la posibilidad de conectar equipos de modo que sean interoperables es una manera de garantizar las redes abiertas y de evitar el abuso de posiciones dominantes.

En especial, un algoritmo, en la medida en que es una secuencia de instrucciones destinadas a efectuar una tarea específica, puede englobar tanto procesos “técnicos” como “no técnicos”. Pero, analizado de forma abstracta y en términos de lógica pura, un algoritmo no tiene puntos de referencias físicos, pues puede aplicarse a muchas funciones diferentes. En este sentido, un algoritmo no debiera ser objeto de monopolio. El software libre puede verse limitado en el uso de ideas, estándares y/o algoritmos que, en realidad, no debieran haber sido patentados por ir en contra de los principios en que se deben sustentar las patentes.

#### Ejemplo

Por ejemplo, el estándar GIF puede infringir la patente Núm. 4.464.650, relativa al algoritmo de compresión LZW, concedida en 1981 por la Oficina de Patentes de Estados Unidos (USPTO). Imaginemos, a la contra, que un navegador no pudiera leer archivos en formato GIF, tan utilizados por los usuarios.

Estas limitaciones restarían competitividad al software libre respecto a los avances “patentados” del software propietario y le impedirían la compatibilidad con este último en la medida en que tampoco podría utilizar las ideas, los estándares y los algoritmos patentados.

No es sorprendente, pues, que empresas como Microsoft obtengan muchas patentes de software con la intención de eliminar la competencia que supone el software libre.

#### Ejemplo

Microsoft incluye en sus licencias de patentes que éstas no puedan ser convertidas a las licencias GPL, ni siquiera a la LGPL. Pero esto también puede ser atacado bajo las normativas antimonopolio (derecho de defensa de la competencia) en la medida en que la patente no se utiliza correctamente.

En definitiva, lo anterior está haciendo que, en la práctica:

- a) Se solicitan patentes a los solos efectos defensivos (es decir, para evitar que otros titulares les demanden), especialmente para entrar en el juego de los acuerdos de licencias cruzadas (y lograr compensar sus patentes con las de otros titulares).
- b) Los titulares se niegan a conceder patentes, en la medida en que prefieren excluir a sus competidores (en rigor, con ciertas matizaciones, no hay una obligación legal de otorgar una patente).
- c) Los titulares sólo conceden licencias con la única condición de que el licenciado:
  - Licencie todo lo patentado por el titular o, por lo menos, licencie otras patentes del titular.
  - No desarrolle productos competitivos con los del titular de la patente.

#### Nota

Ello supone un gran problema para el software libre, en la medida en que el software libre es competidor en varios nichos de mercado, como sistemas operativos, bases de datos, entornos de ofimática, etc.

- Pague regalías en función de las ventas no sólo de los productos patentados, sino también de los no patentados.

**Nota**

Ello supone un gran problema para el software libre porque sería difícil conseguir una licencia de la patente, ya que las ventas de software libre son cero y porque el software libre no controla el número de copias distribuidas (obsérvese que el desarrollador de software libre no puede saber cuántas copias hay en todo el mundo ni obtiene beneficios económicos suficientes como para pagar licencias de uso ilimitado).

- Etcétera.
- d) Las grandes empresas se compensan mutuamente con su larga cartera de patentes (lo que se llama *cross-licensing agreements* o *acuerdos sobre licencias cruzadas* sobre el compartimiento de sus grandes colecciones de patentes). Ello significa que las pocas empresas que pueden entrar en este juego se reservan el mercado del software y con su enorme portafolio o fondo de patentes restringen la entrada al mercado de nuevas empresas. Su filosofía es patentar tanto como puedan y exigir a los demás tanto como puedan para rentabilizar al máximo sus largas carteras de patentes, de modo que los precios de las licencias suben arbitrariamente.

Como consecuencia, los grandes oligopolistas acumulan patentes y se las licencian entre ellos para evitar las posibles denuncias por la acumulación defensiva de patentes. En cambio, las pequeñas empresas o informáticos individuales que desarrollen software libre apenas podrán pagarse una solicitud de patente y, muy probablemente, infringirán muchas de las patentes de las grandes corporaciones.

Las grandes empresas con patentes de software inválidas saben que, aunque el demandante amenace con que ha descubierto que la idea sobre la invención ya existía en el estado de la técnica anterior a la solicitud de la patente y que, por tanto, la patente es nula, el demandado no tendrá dinero para plantear la nulidad de la patente y acabará condenado en el procedimiento por infracción de patente.

**Ejemplo**

Tratar de invalidar la patente de Acacia Media Technologies, relativa a la transmisión y recepción de ficheros de audio y vídeo comprimido por Internet (que puede llegar *de facto* a imponer la censura privada en las redes) ante un tribunal costaría más de un millón de euros. Dichas patentes son las patentes estadounidenses USPO 5132992 y 6144702 y la muy similar patente europea EP 0566662 (incluye España).

Vemos, pues, que la actual práctica de conseguir una larga cartera de patentes se justifica, en realidad, en la obtención de patentes defensivas y acuerdos de licencias cruzadas, pero no como compensación a los costes invertidos para asegurar la innovación. Y la guerra no sólo incluye a los grandes desarrolladores de software (como el gigante Microsoft), sino también a los grandes productores de hardware (como es el caso del gigante IBM, empresa que curiosamente, y con diferencia, es la que tiene el mayor número de patentes concedidas a nivel mundial).

En definitiva, las patentes de software se utilizan en la práctica de forma agresiva para luchar con armas legales contra los competidores, pero no para luchar contra productos superiores.

Finalmente, no debemos olvidar que las patentes de software impiden que el software libre pueda probar el buen funcionamiento del software propietario y, en definitiva, corregir sus errores y perfeccionar sus ideas. Y yendo más lejos, pueden impedir la utilización de un algoritmo en software libre basado en la misma idea (si bien expresado de forma diferente) que otro patentado, bajo la amenaza de considerables y cuantiosos riesgos de demanda.

- 6) Las patentes de software tienden a promover el secreto industrial y a fomentar el monopolio abusivo por el interés que hay en no publicar el código fuente en la descripción de la patente (y condicionarla al correspondiente *know-how* o, en su caso, a no solicitar una patente si las normas de patentes obligan a la publicación del código fuente), lo cual es inconsistente con la finalidad histórica de las patentes de promover la compartición de conocimiento, y

contribuye a eliminar la competencia innovadora, lo que también es inconsistente con su supuesto efecto económico.

La publicación de la patente debería incluir una descripción que permitiera a un entendido en el área reproducir el proceso o invención, a efectos de justificar la recompensa al monopolio del inventor. Sin embargo, al no incluirse el código fuente en los documentos de patente, para reproducir un procedimiento es necesario volver a desarrollar el código fuente, con lo cual el aporte de una patente es, en el mejor de los casos, dudoso.

#### Ejemplo

Por ejemplo, para el caso de invenciones es necesario proveer planos y diagramas que explican claramente cómo funciona un elemento dentro del sistema que se pretende patentar. En el caso de software, en la práctica se agregan solamente descripciones vagas de lo que se pretende patentar, y es necesario desarrollar el producto para tener un programa concreto y usable.

En este aspecto, las patentes de software no fomentan la investigación y el desarrollo en el campo de los programas de ordenador, ni la producción de programas que satisfagan mejor (cualitativa y cuantitativamente) las necesidades de los ciudadanos y las empresas.

Lo anterior es completamente contrario a la filosofía del software libre, para el cual, dada la complejidad de los actuales programas de ordenador y la mayor dificultad en la descompilación y reimplementación de un programa desde su código objeto, es imprescindible tener acceso al código fuente para poder mejorar la calidad del software que se desarrolla en el mundo.

Fueron necesarios unos dos años hasta que los desarrolladores de procesadores de texto pudieron descompilar los formatos de los archivos del Microsoft Office 97 para que sus productos fueran compatibles con los productos Microsoft. Para no infringir una patente, las licencias de patentes deberían ser gratuitas u obligatorias y otorgadas sin demora a efectos de permitir la interoperabilidad entre distintos programas de ordenador.

#### Nota

El *copyleft* permite el acceso al código fuente; ver el *copyleft* en la unidad 2 y las licencias GPL en las unidades siguientes.

En la medida en que el código fuente no esté publicado, el programa de ordenador es potencialmente susceptible de ser protegido por el sistema de patentes sin peligro real a que se revele el “secreto” de lo patentado.

Debería ser obligatorio que el documento de patente incluyera una divulgación de la invención tal que facilite la tarea de las personas que deseen adaptar un programa a otro ya existente que incorpore características patentadas, o la posibilidad, en caso de que los derechos de patente se ejerzan de forma abusiva, de recurrirse a las licencias obligatorias, así como a la legislación en materia de competencia para conseguir que el titular de la patente no ostente el monopolio abusivo.

Lo ideal, bajo los principios que rigen el sistema de patentes, sería que se obligase a que en la descripción (que es pública) se incluya el código fuente y, por supuesto, debería permitirse plenamente la investigación sobre el objeto de las patentes y, por tanto, su descompilación, según se establece con carácter general para las invenciones patentables.

Si bien el sistema de patentes tampoco obliga a publicarlo absolutamente todo, como así serían las técnicas importantes o *know-how* que, de otra manera, no podrían permanecer como secretos industriales o comerciales.

Lo cierto es que debería evitarse lo que actualmente ocurre en muchos documentos de patente: que incluyen mucha información, pero de poca utilidad (por ejemplo, suelen describir mal las especificaciones del programa de ordenador y excluir el código fuente), en la medida en que buscan la mayor protección posible y, por tanto, una cobertura que excluya a terceros de poder desarrollar soluciones tecnológicas alternativas.

Finalmente, mencionaremos que no tiene sentido mantener en las patentes la limitación a la descompilación establecida en la Directiva sobre Protección Jurídica de Programas de Ordenador, pero sí que habrá que permitir el uso de las interfaces, las cuales no deberían ser patentadas. Obsérvese que la descompilación en Europa y Australia es ilegal, a no ser que se haga a los efectos de



interoperabilidad o de reparación de errores. En cambio, en Estados Unidos, la descompilación es siempre ilegal y en Japón sólo es ilegal respecto al software desarrollado en Estados Unidos.

#### Ejemplo

La no armonización en el tema de la descompilación tiene su importancia. Por ejemplo, en 1994 la empresa Stac ganó contra Microsoft porque la última infringió una patente de la primera. Sin embargo, Microsoft ganó contra Stac por descubrimiento de secretos industriales porque la segunda tuvo que descompilar el programa de ordenador de Microsoft para procurarse una prueba.

Y, por supuesto, debería mantenerse la exigencia de que el titular de la patente tenga adecuadamente abastecido al mercado con el resultado de la invención patentada.

- 7) Las patentes de software son muy costosas y el procedimiento para su concesión puede ser muy largo y complicado. De hecho, requiere no sólo pagar cuantiosas tasas en el procedimiento de solicitud y concesión, sino también los honorarios de ingenieros informáticos y abogados (y demás profesionales implicados, según sea conveniente o necesario en los distintos sistemas legislativos de cada país) para desarrollar los documentos de patente (en particular, detallar de forma precisa las reivindicaciones) y hacer el seguimiento procedimental, no sólo de oficina (vía administrativa), sino también en los juzgados (vía contenciosa), que pueden llevar años de gastos y dolores de cabeza inasumibles para los más pequeños.

Además, el hecho de que deba justificarse que la patente de software vaya implementada en una invención mayor requiere unos conocimientos adicionales a los de la mera programación, por lo que supone de un ensamblaje complicado de productos y de una tecnología compleja; es decir, requerirá profesionales cualificados.

En cambio, este planteamiento no tiene sentido para el software libre, dado que su desarrollo no requiere inversiones importantes

en I+D (a diferencia de las cuantiosas inversiones que se requieren en la industria química –incluidos los productos farmacéuticos–, la industria médica o los sectores de maquinaria muy especializada), a menudo desarrollado por pequeñas empresas o informáticos independientes con escasos recursos económicos y con menos acceso a la información necesaria para tramitar y defender una patente. No hay que olvidar que el software en general, si bien es una tecnología compleja, está abierta también a los pequeños desarrolladores.

No deja de ser un absurdo que los informáticos desarrolladores de software tengan que dedicar una gran parte de sus recursos (energías, dinero, etc.) a hacer búsquedas de anterioridades en las bases de datos de patentes o en otras fuentes para asegurarse, antes de escribir cualquier código, de que no infringen patentes de software de terceros o a contratar a profesionales que les puedan proponer una estrategia adecuada para minimizar sus posibles riesgos. Especialmente, teniendo en cuenta que estos recursos serán cada vez mayores en la medida en que la base de datos mundial de patentes de software crece cada vez más, de manera exponencial, y los litigios por patentes de software van a ser más frecuentes. Cada vez va a ser más difícil para los examinadores y desarrolladores de software determinar la novedad y la actividad inventiva. ¿Hasta dónde vamos a llegar?

Por otro lado, tampoco hay que olvidar que algunas de las solicitudes de patentes pendientes de concesión son al principio secretas y se publican posteriormente (en Estados Unidos suele tardar unos 18 meses). Sin embargo, durante ese tiempo, un programador de software libre puede haber desarrollado la misma idea de forma totalmente independiente y ser demandado por infringir una patente que no conocía, por los efectos retroactivos que tiene la patente (a la fecha de solicitud).

¿Acaso las patentes de software no están complicando demasiado las cosas a los desarrolladores de programas de ordenador? Especialmente, al informático independiente (como profesional autónomo) o a las pequeñas consultoras compuestas de unos pocos profesionales, que no van a tener manera de leerse todas las patentes publicadas y entender qué cubren exactamente (espe-

cialmente, por lo difíciles que son de comprender, dado el lenguaje legal utilizado). ¿No se induce a los programadores de software libre a que no comprueben las patentes de software existentes (incluso a sabiendas, por la imposibilidad que implica con sus recursos) y, en definitiva, a que su producto competitivo infrinja alguna patente para poder ser coaccionado bajo amenaza de demanda en un juicio con indemnizaciones millonarias?

En la medida en que la patente puede limitar la libertad y entrada en el mercado de los desarrolladores de software libre, se limita la innovación y el desarrollo del software libre. Por ello, especialmente el sector de los desarrolladores de software libre con menos recursos y los usuarios finales de software libre, defienden en los foros públicos europeos los puntos siguientes:

- a) Consideran que la legalización de las patentes de programación en Europa causaría grandes perjuicios a la sociedad de la información europea, en general, y a la capacidad de innovación y desarrollo de los profesionales y las empresas del sector, en particular.
- b) Creen que la legalización de las patentes no fomenta de ninguna manera la investigación y el desarrollo en el campo de los programas de ordenador ni la producción de programas que satisfagan mejor (cualitativa y cuantitativamente) las necesidades de los ciudadanos y las empresas europeas.
- c) Solicitan a las instituciones con competencia en este asunto que tomen las medidas necesarias, que se aseguren de que la legislación sobre patentabilidad no afecte a los programas de ordenador y, por tanto, que los programas de ordenador queden de forma exclusiva bajo la legislación sobre derechos de autor y *copyright*, que hasta el momento se ha mostrado como la más adecuada para ellos.
- d) Instan a las instituciones europeas a clarificar la situación legal en este sentido en los aspectos y términos que sea necesario.
- e) Entienden muy perjudicial que se impida la comercialización de las tecnologías patentadas por terceras partes cuando han sido descubiertas de forma totalmente independiente.

- f) Consideran que la protección por patente no es adecuada para los programas de ordenador, que su impacto económico y sobre la innovación es negativo, y que el sector informático puede desarrollarse perfectamente sin ella, como ha sucedido hasta muy recientemente, incluso en los países donde se ha introducido últimamente.
- g) En particular, consideran que la patentabilidad del software en la Unión Europea tendría, entre otros, los siguientes efectos negativos:
- Reduciría la innovación y reforzaría los monopolios en un recurso tan básico como el software, con el perjuicio tanto para las posibilidades de elección de los consumidores como para la relación calidad-precio de los productos y servicios en este sector, y se privaría a los ciudadanos de una sociedad de la información sana.
  - Perjudicaría el comercio electrónico al permitir la extorsión legal por parte de poseedores de patentes de software.
  - Comprometería las libertades de creación, publicación y comercialización de los resultados del trabajo propio (el poseedor de una patente podría impedir la publicación o comercialización de un programa original por su propio autor).
  - Causaría incertidumbre legal a los poseedores de derechos de autor por causa de la inflación de patentes (no sabrían si infringen una patente de un tercero hasta que les llegara la amenaza o la citación judicial correspondiente).
  - Pondría en peligro a los profesionales y pymes que no tienen los recursos para acumular patentes o ir a juicio y representan la mayoría de la ocupación y la innovación en tecnologías de la información en la Unión Europea.
  - Además, introducirían una contradicción legal fundamental al usar patentes para monopolizar información (los programas son simplemente información), en lugar de su propósito original de difundir información sobre invenciones.

### 3.5. Marcas

En este apartado vamos a tratar las marcas de software, si bien no de manera tan amplia como hemos hecho con las patentes. En realidad, las marcas de software no suponen ningún debate especial en relación con la protección del resto de productos y servicios que también se protegen bajo la figura jurídica de las marcas. Pero no por ello debemos dejar de prestar atención a sus aspectos legales. De hecho, es importante señalar que hay licencias de software libre que obligan (por ejemplo, la licencia de Apache) o prohíben (por ejemplo, la licencia de Zope) su uso.

#### 3.5.1. Concepto, función y clases de marca

En general podemos decir que las distintas legislaciones de marcas entienden que las marcas sirven para identificar productos (software, como OSI Certified Mark, Unix; o hardware como Machintosh, Hewlett Packard, etc.) o servicios, es decir, prestaciones (por ejemplo, los servicios de desarrollo de un programa de ordenador, el servicio Red Hat, el servicio Suse, etc.).



El Convenio de Niza para la clasificación internacional de productos y servicios en materia de marcas (también conocido como *nomenclator internacional*) tiene por objeto facilitar la información sobre marcas a nivel nacional e internacional, de forma que los países firmantes utilicen una clasificación única a efectos administrativos en las marcas que se inscriben ante sus registros. El monopolio de una marca sólo es válido para los productos y servicios que fueron señalados en los documentos de solicitud. Por ejemplo, las clases más habituales para el software son la 9 (de "producto") y la 42 (de "servicio").

La verdadera función de las marcas es valorar los objetos industriales y las organizaciones que los producen y distinguirlos de otros similares (capacidad "distintiva" o diferenciadora de la marca). Precisamente, los consumidores distinguen los objetos industriales función,

#### Nota

En la ley española, por *marca* se entiende todo signo susceptible de representación gráfica que sirva para distinguir en el mercado los productos o servicios de una empresa de los de otras.

de la función identificadora (*goodwill*) asociada a las organizaciones. Se prohíbe, por tanto, el aprovecharse del prestigio ajeno. La marca no sólo tutela a su titular, sino también a los consumidores y al correcto funcionamiento del mercado en general.

También deberán observarse la normativa relativa a la publicidad comparativa, en la medida en que se utilice una marca para indicar que un programa de ordenador es compatible con la ejecución de otro (incluida la compilación, la llamada a otras librerías, la interoperatividad, etc.), o con un ordenador.



En definitiva, las marcas tratan de evitar el “riesgo de confusión” de los productos y servicios en el mercado, incluido el “riesgo de asociación” entre empresas.

Al igual que las patentes, y, a diferencia de los derechos de autor, las legislaciones suelen establecer que la marca debe registrarse en la oficina de marcas correspondiente a un territorio determinado para poder reivindicar su propiedad, y deben pagarse una serie de tasas administrativas para que pueda tener efectos en el tiempo.

Las marcas y las patentes se diferencian en los aspectos siguientes:

- a) Las **patentes** confieren al titular un derecho exclusivo de explotación industrial y comercial de producto o procedimiento patentado por un tiempo limitado (20 años), y lo facultan para prohibir a terceros que fabriquen y comercialicen el producto patentado y obtenido por el procedimiento patentado sin su consentimiento.
- b) Las **marcas**, en cambio, confieren al titular el derecho exclusivo de su uso para distinguir sus productos o servicios en el mercado de productos o servicios similares de otras empresas por un tiempo ilimitado (la marca puede ser renovada indefinidamente, siempre que se paguen las tasas y sea efectivamente usada en el mercado), y lo facultan para prohibir a terceros que comercialicen productos o servicios similares con la misma marca, o una marca similar que pueda inducir a confusión al público.

#### Nota

La marca no prohíbe que terceros comercialicen productos idénticos o similares, sino únicamente que los comercialicen con una marca confundible.

Vemos que el monopolio de la marca es más limitado, al no tener por objeto el producto en sí, sino la marca que lo distingue, de modo que no puede prohibir la fabricación ni la comercialización del producto en sí, sino sólo que se comercialice con una marca confundible. En definitiva, marca y patentes tienen objetos, contenido, duración y funciones diferentes.

Las marcas pueden ser denominativas (una o varias palabras), gráficas (un dibujo), letras, cifras o sus combinaciones, mixtas (mezcla de palabras y dibujos), tridimensionales (la forma de un objeto: por ejemplo, la botella de Coca-Cola o el muñeco Michelin), sonoras (el sonido "Nokia" de un teléfono móvil) o cualquier combinación de las anteriores. Normalmente, las marcas asociadas a un programa de ordenador serán denominativas, gráficas o mixtas, en la medida en que se visualizan en pantalla.

Las leyes de todo el mundo suelen prohibir el registro de marcas genéricas (por ejemplo, la marca Software). Esta regla lógica tan natural se ignora muchas veces y los resultados para la empresa pueden ser muy perjudiciales, especialmente cuando se ha gastado mucho dinero en publicidad. Muchas legislaciones de marcas también prohíben el registro de nombres geográficos (a excepción de las denominaciones de origen). Existen también en las legislaciones de marcas otra serie de prohibiciones que no mencionaremos en esta unidad, por su extensión y porque no aportan nada especial en relación con el software libre.

Por lo que se refiere a los programas de ordenador, las marcas tienen una eficacia limitada. Se emplean sobre todo para los programas de aplicación destinados a un amplio número de usuarios o para sus servicios de programación. El titular fija la marca sobre el programa a fin de que el usuario (consumidor) lo identifique como proveniente de un empresario determinado y, consiguientemente, con una determinada calidad.

Aunque la función primordial de la marca será indicar el origen empresarial del programa de ordenador, el derecho de marcas puede ofrecer una base para una protección indirecta del producto que identifica. Éste sería el caso, por ejemplo, de las situaciones siguientes:

- La reproducción de un programa (o líneas de código) y su posterior comercialización con el mismo signo distintivo.

- La comercialización de ejemplares de programas de ordenador (o líneas de código) producidos por un titular eliminando la marca del titular que figuraba en los programas.



En definitiva, una **marca** es un título que concede el derecho exclusivo, territorial y temporal, *erga omnes* (absoluto) de utilización de un signo para la identificación de un producto o un servicio en el mercado.

### 3.5.2. Duración y objeto

En principio, una marca es para siempre, con tal de que se pague la correspondiente tasa administrativa de renovación.



La **duración de la marca** suele ser de 10 años, sin perjuicio de su renovación por sucesivos bloques de 10 años que puede hacerse con carácter indefinido siempre que se cumpla con el pago de las tasas y el cumplimiento de los requisitos legales. Sin embargo, las marcas deberán ser usadas en la práctica (normalmente, bajo pena de “caducidad” en caso de no uso durante 5 años).

Las marcas confieren un monopolio sobre los productos y servicios designados. Una vez que dichos productos han sido marcados por el titular y puestos a la venta, el derecho se agota. Cuando se ha vendido el producto, el titular no podrá exigir que se le devuelva en base a su monopolio de la marca y ser vendido una segunda y tercera vez sin que el producto deje de estar asociado a la marca.

Vemos que, igual que con las patentes y los derechos de autor y *copyright*, en sede de marcas también es aplicable el concepto de **agotamiento de derecho** (*detournement de droit*) para los productos comercializados en el territorio donde la marca extiende sus efectos.





Las marcas también suponen un monopolio legal sobre una “idea” cuyos límites no pueden ser violados por terceras personas. El propietario de la marca es el único que puede hacer uso de ella personalmente o autorizar su uso por parte de terceras personas.

La percepción por parte de los consumidores de la marca es lo que da lugar a un juicio sobre las cualidades y notas de los productos o servicios (función identificadora), y permite comparar entre los de otros competidores y facilitar al consumidor tomar un juicio positivo o negativo. En consecuencia, el valor de la marca es el que adquiere en el tráfico, es decir, el que recibe de los consumidores y reconocen sus competidores.

El derecho fundamental que confiere la marca es otorgar a su titular el derecho exclusivo de utilizarla en el tráfico económico. Y, por tanto, el titular de la marca podrá prohibir, entre otras cosas:

- Poner el signo en los productos o en su presentación.
- Ofrecer los productos, comercializarlos o almacenarlos con esos fines u ofrecer o prestar servicios con el signo.
- Importar o exportar los productos con el signo.
- Utilizar el signo en los documentos mercantiles y la publicidad.
- Usar el signo en redes de comunicación telemáticas y como nombre de dominio.
- Poner el signo en envoltorios, embalajes, etiquetas u otros medios de identificación u ornamentación del producto o servicio, elaborarlos o prestarlos, o fabricar, confeccionar, ofrecer, comercializar, importar, exportar o almacenar cualquiera de esos medios incorporando el signo, si existe la posibilidad de que dichos medios puedan ser utilizados para realizar algún acto que conforme a las letras anteriores estaría prohibido.

- Impedir que los comerciantes o distribuidores supriman dicha marca sin su expreso consentimiento, si bien no podrá impedir que añadan por separado marcas o signos distintivos propios, siempre que ello no menoscabe la distintividad de la marca principal.

Sin embargo, siempre que se haga de “buena fe” y no constituya un “uso a título de marca”, los terceros podrán utilizar en el mercado sin el consentimiento del titular:

- su nombre completo y domicilio.
- indicaciones relativas a la especie, calidad, cantidad, destino, valor, procedencia geográfica, época de producción de producto o prestación del servicio u otras características.
- la marca, si se trata de indicar el destino de un producto o servicio, en particular en el caso de accesorios o piezas sueltas, siempre que ese uso se realice conforme a las prácticas leales en materia industrial o comercial.

#### Ejemplo

Una empresa como Red Hat puede ayudarse de su marca para ejercer un monopolio sobre el software libre que comercializa. De este modo, dará una garantía de calidad de los paquetes de distribución (por ejemplo, que no hay virus o errores en la versión Red Hat, etc.) y diferenciará las herramientas de instalación e integración de componentes.

### 3.5.3. Adquisición de las marcas

En algunos países, los derechos que confiere la marca se adquieren mediante su registro, con algunas excepciones en relación con las marcas notorias (marcas conocidas por el sector pertinente del público al que se destinan los productos o servicios) y las marcas renombradas (marcas conocidas por todo el público en general).

En otros países, en cambio (por ejemplo, en los países anglosajones), las marcas se adquieren con el uso, y su registro es una mera formalidad con efectos probatorios y documentales.

Dado que las marcas son territoriales, podemos adquirir distintos títulos de marca: marca nacional, solicitud internacional o marca comunitaria.

### 1) Marca nacional

Cada país tiene su propia oficina de patentes y marcas (excepto los países del Benelux, que tienen una oficina común para los tres; o la marca comunitaria, que tiene su oficina en Alicante, con efectos en toda la Unión Europea) y es imprescindible tramitar la solicitud de marca separadamente para la protección de la marca en ese país. Ello requiere el cumplimiento de unos sencillos requisitos formales y el pago de una tasa administrativa (tasa de solicitud, tasa de concesión, tasa de renovación, etc.).

El procedimiento de solicitud varía en cada oficina de patentes y marcas de cada país. Básicamente, trata de filtrar las solicitudes que no cumplan con los requisitos de la ley. Comienza con el depósito de documentos, sigue por la publicación de la solicitud en el correspondiente boletín de marcas; posteriormente, se abre el plazo de alegaciones; y por último, se concede o deniega.

Las legislaciones de marcas suelen permitir la existencia simultánea de dos signos iguales registrados para productos o servicios distintos, con el fin de que se agoten las palabras utilizables (a estos efectos tiene importancia la clasificación de Niza: *nomenclator* internacional, a la que ya nos hemos referido). Así, por ejemplo, no es lo mismo la marca IBM para servicios de programación que para alimentos lácteos. Precisamente, para evitar que nuestro signo distintivo quede limitado a una sola serie de productos o servicios, es aconsejable, en caso de dudas, incluir el máximo de productos, si bien ello supondrá mayores costes en tasas y tramitación.

Al igual que hemos visto en materia de patentes, el solicitante de una marca también goza de prioridad internacional (*prioridad unionista*) para solicitar la misma marca en los demás países firmantes del Convenio de París. La duración del plazo de esta prioridad internacional es de 6 meses, y es improrrogable.

La prioridad internacional permite registrar la marca en múltiples países, pero el proceso es bastante caro y complicado (supone seguir un procedimiento separado para cada país). Por esta razón, en el año 1891 se firmó el Convenio de Madrid, relativo al registro internacional de marcas que veremos en el apartado siguiente.

## 2) Solicitud internacional

La solicitud internacional es un sistema de cooperación administrativa internacional dirigido a simplificar los trámites necesarios para registrar una marca en diversos países de forma simultánea. Dicha tramitación resulta más barata que la tramitación separada e individualizada país por país.

El procedimiento se gestiona en Suiza por la Organización Mundial de la Propiedad Intelectual (OMPI), que es una agencia especial de las Naciones Unidas. El procedimiento parte de una marca ya registrada (nacional, comunitaria, etc.) y extiende sus efectos a otros países (permite reivindicar la protección a varios países a la vez) por mediación de un solo documento: la solicitud internacional.



El resultado es una “colección de títulos de propiedad” sobre una marca separados e independientes.

Las tasas dependen de cada país. Se sumarán tantas tasas como países designados. Dicha suma es siempre menor que la correspondiente a las solicitudes separadas.

El inconveniente de este sistema es que, durante un período de cinco años a partir de la fecha de su registro, un registro internacional dependerá de la marca que se ha registrado o solicitado en el país de origen. Y si el registro de base deja de tener efecto, ya sea por cancelación como resultado de una decisión de la oficina de origen o un tribunal, por cancelación voluntaria o por no renovación, el registro internacional dejará de estar protegido en este período de cinco años y podrá ser cancelado.

Asimismo, si el registro internacional se basó en una solicitud presentada en el país de origen, éste quedará cancelado si se deniega o retira la solicitud en el período de cinco años o si el registro resultante de esa solicitud deja de tener efecto durante ese período, lo cual ha sido muy criticado.

Sin embargo, tras la expiración de ese período de cinco años, el registro internacional pierde toda dependencia con el registro de base o la solicitud de base.

### 3) Marca comunitaria

El procedimiento de cooperación administrativa internacional no es suficiente para el buen funcionamiento de la Unión Europea.

En consecuencia, el **Reglamento (CE) 40/94 del Consejo, de 20 de diciembre de 1993**, sobre la marca comunitaria, establece la posibilidad de solicitar una marca comunitaria, bien directamente ante la Oficina de Armonización del Mercado Interior (OAMI), o bien por mediación de las respectivas Oficinas de Patentes y Marcas de los estados miembros de la Unión Europea para su posterior remisión a la OAMI.



El resultado es un título único que tiene validez simultánea en todo el territorio de la Unión Europea.

Las tasas son mayores que las de una marca nacional, pero menores que la de la suma de varias de ellas.

El inconveniente de este sistema es que, si se deniega la marca en un país sobre la base de una marca nacional anterior, las tasas de solicitud se pierden y hay que volver a empezar de nuevo para solicitar otra marca distinta, o bien solicitar las correspondientes marcas nacionales restantes en las que no haya anterioridades iguales o similares. Otro inconveniente es que por su sistema de fácil concesión, se trata de marcas débiles.

El procedimiento de examen de una solicitud de registro de marca comunitaria puede resumirse de forma esquemática en las siete fases siguientes:

- 1) Presentación de la solicitud.
- 2) Examen preliminar y acuerdo sobre una fecha de presentación.
- 3) Examen de las demás formalidades (tasas, clasificación, prioridad, antigüedad, etc.).
- 4) Búsqueda de anterioridad.
- 5) Examen de los motivos absolutos de denegación.
- 6) Anuncio público de la solicitud.
- 7) Expedición del título e inscripción en el registro de las marcas comunitarias.

#### 3.5.4. Cesión y licencias de las marcas

El tema de la cesión y de las licencias en el software libre tiene su importancia. Pensemos en el uso de la marca Linux en distribuciones de intermediarios como Red Hat, o el uso de OSI Certificated para una licencia, o el uso de la marca Apache en servidores, etc. Este tipo de usos requieren licencia.

#### Nota

En las siguientes unidades profundizaremos más en este asunto.



La marca, como las patentes y los derechos de autor y *copyright*, constituye un título de propiedad inmaterial sobre el cual, normalmente, en todas las legislaciones es posible efectuar cualquier tipo de compraventa, hipoteca, garantía, alquiler, *leasing*, etc. Al igual que las patentes, es transmisible, no sólo la solicitud de marca, sino también el título de marca, una vez ha sido concedida por la correspondiente oficina de patentes y marcas.

**Nota**

En particular, la inscripción en la correspondiente licencia ante la Oficina Española de Patentes y Marcas para que tenga efectos frente a terceros debe hacerse por escrito para poder acceder a dicho registro público. También ahí será posible anotar los embargos judiciales sobre una marca, las sucesiones, herencias, etc.

La transmisión de las marcas puede hacerse de forma separada a la de una empresa (éste es el caso de la marca nacional española o de la marca comunitaria). Sin embargo, en algunos países la regla general es la contraria (sólo se admite la transmisión de las marcas junto con la estructura empresarial o lo que se conoce como *assignment in gross*).

Al igual que las patentes y los derechos de autor, las marcas pueden ser objeto de licencia. Las licencias podrán ser:

- a) Para la totalidad o una parte de los productos o servicios para los cuales esté registrada la marca.
- b) Para la totalidad o una parte del territorio de la marca.
- c) Exclusiva o no exclusiva.

### 3.6. Secretos industriales

Además de la protección por derechos de autor, patentes y marcas, el titular de un programa de ordenador puede protegerlo por el secreto industrial. Para acabar esta unidad haremos una breve reseña al secreto industrial, sin desarrollar el tema con tanta amplitud como hemos hecho con las patentes o las marcas.



Por **secreto industrial** podríamos entender toda aquella información, incluidas las fórmulas, modelos, compilaciones, bibliotecas, programas, mecanismos, métodos, técnicas o procesos que:

- 1) tengan un valor económico independiente, actual o potencial, que no esté en el dominio público ni a disposición de terceros que puedan obtener un beneficio de su conocimiento o divulgación, o
- 2) sean resultado de unos esfuerzos que razonablemente se deben mantener en secreto.

En definitiva, los secretos empresariales contendrán aquella información que den al empresario una ventaja competitiva respecto a sus competidores.

Para que la obtención de un programa de ordenador que un empresario mantiene en secreto sea contraria a las normas de corrección y buenos usos mercantiles dependerá lógicamente del medio por el cual tal conocimiento se obtiene. No debe olvidarse que el ordenamiento jurídico no otorga al titular del secreto un derecho de exclusiva sobre el mismo.

Es posible que la programación o adquisición de las líneas de código por parte de un empresario se lleve a cabo por medios legítimos, como la investigación o por actos de divulgación efectuados por un tercero sin reserva alguna (por ejemplo, publicación en revista especializada).

Por lo que al software libre se refiere, contrario a los buenos usos mercantiles será la adquisición por parte del programador de software libre del conocimiento sobre el secreto por medios ilegítimos o la ilegítima comunicación o utilización de tal conocimiento, aunque su adquisición fuere irreprochable. Los supuestos pueden ser variados: espionaje industrial, aprovechamiento de tratos precontractuales, adquisición del secreto por medio de empleados del titular, etc.

El mismo hecho de que no exista disponibilidad del código fuente en los documentos públicos de patente permite que la figura jurídica de los secretos industriales confiera suficiente protección a las ideas de implementación de software. Así, si una idea es lo suficientemente tri-



vial para ser implementada sin requerir el código fuente ni la publicación de las ideas, probablemente nunca debió haber recibido una patente y, en el caso del secreto industrial, puede ser usada libremente si alguien logra desarrollar la misma idea de forma independiente.

Normalmente, los programas de ordenador que se mantienen en secreto suelen contener huellas digitales que no son más que instrucciones redundantes o en círculo en el módulo de programación a efectos de tener, más adelante, una prueba del acceso al secreto en un posible juicio.

En ese caso, será el juez quien, a la vista de las circunstancias concurrentes, habrá de valorar si ha habido o no, infracción de las normas de corrección y buenos usos mercantiles. Valoración que no debe limitarse a la constatación de la eventual vulneración de concretos preceptos legales, sino que se han de considerar los objetivos perseguidos por la disciplina contra la competencia desleal.

En este sentido, no cabe duda de que los programas de ordenador mantenidos en secreto (y, por tanto, calificables de secreto empresarial) pueden ser protegidos por las normas de competencia desleal que reprimen la violación de secretos. Pues la competencia desleal protege muchos otros supuestos también aplicables al software.

Una interpretación amplia del objeto de protección de la competencia desleal protegerá al empresario titular del programa de ordenador, no sólo frente al que vulneró el secreto y es competidor de su titular, sino también contra todo aquel que violó el secreto para utilizar el programa de ordenador en su actividad económica, aunque no sea competidor del lesionado.

Por otra parte, el software mantenido en secreto puede gozar de tutela penal, bajo el delito de descubrimiento y revelación de secretos. Si bien el tipo de este delito normalmente está dirigido contra el empleado, encargado u obrero de una fábrica y otro establecimiento industrial que, en perjuicio de su dueño, descubriere los secretos de industrial. Ello supone que el eventual competidor que se beneficia de la revelación del empleado no recibirá sanción penal, salvo que fuese partícipe en el delito, por lo que su conducta sólo podrá ser

perseguida por la vía de la competencia desleal. Distinto será el supuesto en que el competidor adquiriese personalmente el secreto mediante actuaciones penadas por este delito.

Sin embargo, el que adquiere el secreto con su propio esfuerzo actúa correctamente. Y ello porque el titular del secreto no tiene legalmente la exclusiva y, por tanto, no tiene una protección *erga omnes*.

### 3.7. Conclusiones

Si bien las licencias de software normalmente se centran en los derechos de autor o *copyright*, en la medida en que éstos son los derechos mayormente afectados en el software libre, también deberán tenerse en cuenta los derechos que confieren las patentes, las marcas y, en su caso, las obligaciones de confidencialidad sobre la base de un secreto comercial.

En la medida en que sea necesario conferir protección a los distintos aspectos que cubren estas distintas figuras jurídicas, será necesario que la licencia incluya las cláusulas necesarias que estudiaremos con detalle en las unidades siguientes. En estas unidades se estudiarán las cláusulas apropiadas que deben incluir las licencias de software libre para asegurar –minimizando cualesquiera riesgos jurídicos– quién debe responder por las garantías y responsabilidades inherentes a la programación del software libre.

#### 4. Introducción al contrato de licencia de software

En las unidades anteriores hemos conocido cómo las legislaciones de los distintos países conceden derechos en exclusiva sobre el software a quienes lo desarrollan. Y hemos visto cómo estos derechos en exclusiva se protegen fundamentalmente por las normas sobre propiedad intelectual (*copyrights* en su acepción anglosajona); es decir, los derechos de titularidad sobre el software consisten preferentemente en derechos de autor.

De este modo, el titular de los derechos sobre el software dispone de derechos en exclusiva de orden moral y de orden económico: es el único que puede usar, modificar o distribuir el software, **y es el único que puede autorizar a terceros** el uso, modificación o distribución del software. Es más, podemos decir que quien desarrolla un software, no lo hace para su único uso y aprovechamiento: normalmente, pretenderá su máxima difusión a terceros usuarios, sea por haberlo creado con fines de explotación económica o no.

En este sentido, vamos a ver en esta unidad cómo la “**licencia de uso**” es el instrumento legal que más emplean los desarrolladores de software para distribuirlo a terceros, que pasarán a ser los usuarios-licenciatarios del mismo. Veremos que se trata de un contrato, al consistir en un acuerdo de voluntades entre proveedor de software y usuario (si bien en países anglosajones la licencia puede concederse también de forma unilateral).

La licencia de uso se emplea para distribuir software propietario, porque el proveedor de software pretende conceder a terceros el derecho a usar su programa, pero reservándose él la exclusividad de modificar o distribuir el software. Y la licencia de uso se emplea también como el instrumento legal para distribuir software libre, en este caso para asegurar a los usuarios las libertades de uso, modificación y redistribución.

En esta misma línea, explicaremos las formas que pueden revestir las licencias de software, y su dependencia del soporte físico en que se

distribuya el software: veremos que los derechos que adquiere un usuario sobre el soporte físico (CD-ROM, por ejemplo) pueden ser distintos respecto de los que adquiere sobre el software propiamente dicho. También veremos cómo la licencia de software, y su contenido integrado por condiciones generales, es propio de un software que se distribuye en masa en el mercado.

Para entender la importancia de la licencia de uso en la distribución de software, analizaremos su “función jurídica y económica”. Abordaremos el tema de la “naturaleza jurídica”, para situar el tipo de contrato que es la licencia de uso y qué marco normativo le es de aplicación.

Por último, conoceremos algunas modalidades especiales de licencia de uso, y contratos accesorios a las licencias de uso que, como el de mantenimiento, pueden ser de mucha importancia para el uso y explotación de determinado software.

Con el estudio de esta unidad de aprendizaje, se alcanzarán los objetivos siguientes:

1. Conocer el concepto de licencia de software, así como su carácter de contrato entre dos partes: proveedor del software-licenciante y usuario-licenciario.
2. Analizar la relación entre software y su soporte físico, los distintos derechos que adquiere el usuario sobre cada uno de estos elementos; así como las formas que puede adoptar la licencia de software, en función del soporte físico.
3. Reflexionar sobre el concepto de *software estándar*, distribuido en masa, como una de las características que explican el empleo de la licencia de uso, y que sus cláusulas estén compuestas por condiciones generales.
4. Comprender las razones por las que la licencia de software es el instrumento legal que se emplea habitualmente para distribuir programas informáticos, sean de tipo propietario o libre. Por qué y para qué se emplean: su función económica y jurídica. Asimismo, analizaremos las diferentes funciones que cumplen las licencias de software libre respecto de las de software propietario.

5. Analizar la “naturaleza jurídica” de la licencia de uso de software: conocer que se trata de un contrato *sui generis*, sin una ley que lo regule por completo; y al que se le pueden aplicar, por analogía, normas que regulan otros contratos más tradicionales.
6. Conocer las principales normas jurídicas (leyes, reglamentos, etc.) que son de aplicación a las licencias de software: su marco normativo.
7. Conocer algunas modalidades especiales de licencias de uso de software.
8. Asimismo, saber de ciertos contratos accesorios a las licencias de software y su importancia especial en aquellas licencias (tanto de software propietario como libre) que versan sobre ciertos programas informáticos especializados, de aplicaciones profesionales: contratos de mantenimiento, consultoría y escrow.

#### 4.1. Consideraciones generales



Por **licencia de software** entendemos el contrato que establece los términos y condiciones en virtud de los cuales el autor o titular de los derechos de explotación de un programa informático (al que llamaremos *licenciante*, *titular del software* o *proveedor del software*), permite la utilización de dicho programa a otra persona (a la que llamaremos *licenciataria* o *usuario*).

Se trata, en definitiva, del instrumento legal habitual por el cual quien ha desarrollado un software permite que otros lo utilicen; generalmente, pero no siempre, a cambio de una remuneración económica.

Decimos que la licencia de software es un contrato porque consiste en un acuerdo entre dos partes: el proveedor del software y el usuario. Sin embargo, casi nunca se trata de un contrato que se negocie de tú a tú entre el proveedor del software y el usuario. Es el proveedor del software quien previamente estipula de forma unilateral los términos y condiciones de la licencia. El usuario no puede negociar con

#### Nota

La licencia es el instrumento legal por el que el desarrollador de un software permite su uso a terceros.

el licenciente las condiciones de la licencia, sino que sólo puede limitarse a aceptarlas o a rechazarlas.

De hecho, al emplear el término *licencia*, se hace referencia también al “permiso” o “autorización” que el proveedor del software concede a los usuarios para utilizar un software de su titularidad, sobre el cual posee los derechos en exclusiva gracias a la legislación sobre propiedad intelectual (derechos de autor o *copyrights*). El titular de los derechos sobre el software es el que decide a quién, dónde, cómo, cuándo, a cambio de qué, permite que se utilice “su” software.

Por eso, en la práctica solemos emplear el término *licencia* en dos sentidos: tanto para referirnos al contrato de licencia de software (así, por ejemplo, hablamos de “aceptar los términos y condiciones de la licencia”), como al permiso, autorización o derecho que se concede al usuario para utilizar un software a resultas de haber firmado/ aceptado el contrato de licencia (así, por ejemplo, hablamos también de “tener licencia para utilizar un software”).

De hecho, este doble significado del término *licencia* tiene consecuencias prácticas, y da lugar a ciertas controversias. Ello, porque en el **derecho anglosajón** (EE.UU., Reino Unido) cabe la posibilidad de que las licencias de software sean tanto:

- a) Un **contrato**, en el que además de los términos y condiciones de uso del software, puedan las dos partes (licenciente y licenciario) acordar pactos accesorios al derecho de uso sobre el software, tales como obligaciones de confidencialidad, responsabilidad, tribunales competentes para resolver los conflictos que surjan de la licencia, etc.
- b) Una **declaración unilateral del licenciente** por la que autoriza el uso del software a los que cumplan y respeten ciertas condiciones y límites, de conformidad con la ley de derechos de autor que sea aplicable. En este caso, no se necesita la aceptación expresa del licenciario; por ello, la licencia sólo puede referirse exclusivamente al derecho de uso del software (no debe regular pactos accesorios).

#### Nota

En España y en el resto de países de sistema de derecho “continental”, la licencia de software sólo puede entenderse como un contrato.

En EE.UU. o el Reino Unido, también caben las licencias de software “unilaterales”.

Sin embargo, en el **derecho de la Europa continental**, entre el que se encuentra el derecho español, no parece que sea posible conceder licencias de uso de forma unilateral. Sólo cabe entender la licen-

cia como un contrato, cuyo contenido es aceptado por ambas partes (licenciante y licenciario), porque es precisamente mediante la aceptación por lo que el usuario-licenciario quedará obligado a usar el software respetando los límites y restricciones, tanto los previstos por la ley, como los que exija el proveedor.

Muchas licencias de software son adaptaciones o traducciones de contratos de licencia de software originales en inglés, que incorporan cláusulas propias del derecho anglosajón (propio de EE.UU. o del Reino Unido). Ello da lugar a algunas dificultades de aplicación o interpretación de dichas cláusulas en otros países, como España.



Las licencias de software tienen como objetivo principal estipular los derechos que el proveedor concede al usuario sobre el software (qué puede hacer con el software), así como las limitaciones y prohibiciones que el usuario debe respetar (qué no puede hacer con el software).

Además, las licencias regulan otros aspectos como:

- El número de copias y/o licencias que se otorgan.
- Las características técnicas del ordenador en que se va a ejecutar el programa.
- El modo de entrega e instalación del software.
- El posible período de instalación y pruebas de aceptación por el usuario.
- La duración de la licencia (limitada, prorrogable o indefinida).
- El precio de la licencia (pago de una sola vez o cuotas periódicas).
- El período de garantía.
- Las responsabilidades y limitaciones de responsabilidad del licenciante.
- La entrega de manuales y documentación.

- La ley aplicable y tribunales competentes en caso de litigio derivado de la licencia.

**Nota**

A los efectos de las unidades 4 a 8 nos centraremos en las licencias de uso de software estándar, esté parametrizado o no, a las necesidades del usuario; de este modo, dejaremos de lado los “contratos de desarrollo”, por los que un programador recibe el encargo de un cliente de desarrollar un software a su medida. En tales contratos, si el programador retiene su titularidad sobre el software creado, se incorporará accesoriamamente una licencia para que el usuario pueda utilizar el software. En cualquier caso, a esta licencia le será de aplicación lo que expliquemos en las unidades siguientes.

A continuación analizaremos los distintos modos como se presentan las licencias de software, frecuentemente en función del soporte físico en que se halla incorporado el mismo. También, sabremos por qué las licencias de software suelen estar integradas por cláusulas genéricas (condiciones generales), al distribuirse el software en masa. Pero, ante todo, efectuaremos una primera aproximación a las diferencias entre licencias de software propietario y licencias de software libre.

#### **4.1.1. Licencias de software propietario y licencias de software libre**

Las diferencias entre el software propietario y el software libre se plasman en unas licencias de uso y contenido totalmente diferentes. Ya sea software propietario o software libre, se acostumbra a distribuir por medio de un mismo instrumento legal: la **licencia de uso**.

Algo distinto es el contenido final de estas licencias, que varía radicalmente, en particular los derechos que el proveedor concede al usuario sobre el software:

- a) Las **licencias de software propietario** suelen restringir al máximo los derechos del usuario, los cuales se reducen a una facultad limitada de usar el software y a poder hacer una copia de seguridad del mismo. Se prohíbe al usuario la copia, modificación o



redistribución del software, y se le entrega sólo una copia del mismo en código objeto.

- b) Las **licencias de software libre** contemplan una amplia gama de libertades para el usuario, como el libre uso, la copia, la modificación y la redistribución. Además, el proveedor proporciona o pone a disposición de los usuarios el código fuente.

#### 4.1.2. Soporte físico y formas de las licencias de software

La licencia de software puede adoptar varias formas. La forma de una licencia de software suele depender del modo en que se distribuye el programa y, en concreto, del soporte en que el proveedor proporciona la copia del software al usuario.

Sabemos que el software como tal es un bien inmaterial o intangible. Quien pretenda utilizarlo debe instalarlo y ejecutarlo previamente en un elemento *hardware* (generalmente, un ordenador). Habitualmente, el proveedor del software lo distribuye al usuario en un soporte físico (preinstalado en el disco duro, en disquetes o CD-ROM) que permite su instalación y/o ejecución en el equipo *hardware*.

No obstante, con la aparición de Internet, el soporte físico ya no es imprescindible para distribuir un software, dada la posibilidad de descargar programas *on-line*.

#### Derechos sobre el soporte físico

El software (bien inmaterial, obra intelectual) se distingue del soporte en que se contiene (bien material). Las licencias de software suelen especificar qué derechos tiene el usuario sobre el soporte físico en el que se halla grabado el software:

- a) El usuario puede tener la obligación de devolver el soporte (el CD-ROM, por ejemplo) al proveedor, una vez expire la licencia. En tal caso, entendemos que mientras la licencia se encuentra en vigor, el proveedor proporciona el soporte a título de préstamo, para que pueda instalar y ejecutar el software.

##### Nota

Las formas que puede revestir una licencia de uso suelen supeditarse al soporte en el que se proporciona el software.

##### Nota

Las licencias de uso de software contemplan distintos derechos del usuario sobre el soporte físico en el que se distribuye.



Pensemos en el software de aplicaciones destinado a empresas, sujeto a actualizaciones periódicas y cuya licencia se concede por un plazo determinado (por ejemplo, X años): muchas de las licencias sobre este tipo de software imponen la devolución del soporte al finalizar la licencia o bien cuando el proveedor proporciona al usuario la nueva versión del software.

- b) El usuario no tiene la obligación de devolver el software al proveedor una vez finalice la licencia. Ello es más típico del software comercializado en masa, en particular a consumidores, cuya licencia se suele conceder por tiempo indefinido. Algunas licencias exigen al usuario incluso que, de extinguirse la licencia, destruyan el soporte si no lo devuelven al proveedor. Entendemos que se trata de una exigencia tan deseable para el proveedor (que no desea que el usuario siga utilizando su software o distribuya a un tercero su copia), como prácticamente imposible de hacer cumplir.

#### Ejemplo

Tenemos el caso de Microsoft®, en cuyos “Contratos de Licencia para Usuario Final de Microsoft (CLUF)”, contempla el derecho de Microsoft a terminar el contrato si el usuario no cumple sus términos y condiciones. En tal caso, el usuario debe “destruir todas las copias del PRODUCTO SOFTWARE y todas las partes componentes”.

Otras licencias no imponen obligación alguna al usuario una vez finalice la licencia respecto al soporte del software (ni que lo destruya ni que lo devuelva, como el software preinstalado en el disco duro): únicamente exigen al usuario que deje de utilizar el software. Por tanto, permiten que éste se quede con el soporte, por lo que entendemos en estos casos que el usuario es licenciatario del software y, a su vez, pleno propietario del soporte.



Evidentemente, al usuario no le proporciona mucha utilidad poderse quedar con el CD-ROM o con el disquete una vez finalice la licencia. Lo que de verdad tiene valor

para el usuario cuando adquiere una copia de un software licenciado es el derecho a utilizar el programa. La principal ventaja para el usuario, al poderse quedar con el soporte tras finalizar la licencia, es más bien la comodidad de no tener que devolverlo al proveedor o verse comprometido a destruirlo, y no tanto el exiguo valor en sí del CD-ROM o del disquete.



Un dato importante que debemos tener bien claro es que si el usuario adquiere una copia de un programa de ordenador sujeto a una licencia que le permite quedarse con el soporte, sólo “compra” la propiedad del soporte. Sin embargo, respecto al software, únicamente adquiere el derecho a utilizarlo una vez acepta los términos y condiciones de la licencia.

De este modo, durante la vigencia de la licencia de uso o al finalizar ésta, el usuario podrá dar o vender el soporte físico, pero siempre que haya borrado el software del soporte.

Cuando un usuario acude a una tienda y adquiere la copia de un programa, sólo compra el soporte (CD). El soporte incorpora una oferta de licencia de uso que el usuario deberá aceptar para tener derecho a utilizar el software. Por tanto, ser dueño del soporte no significa automáticamente tener el derecho a utilizar el programa.



Sobre este aspecto, la propia Ley de la Propiedad Intelectual prevé una regla, aplicable a obras literarias o audiovisuales, pero también al software:

“El adquirente de la propiedad del soporte a que se haya incorporado la obra no tendrá, por este solo título, ningún derecho de explotación sobre esta última” (art. 56.1).

**Nota**

Las licencias de uso de software deben constar por escrito.

**Formas de la licencia**

La **forma** de la licencia de software viene tradicionalmente pareja con el medio por el cual se distribuye el software. La Ley de la Propiedad Intelectual exige un solo requisito de forma para las licencias de software, y es que deben formalizarse **por escrito**.

Por tanto, no son válidas las licencias de software que se otorguen verbalmente. La exigencia de que conste por escrito se debe, en principio, a proporcionar al licenciatarario cierta protección y seguridad, al quedar plasmados por escrito sus derechos y limitaciones sobre el software licenciado.

No obstante, el requisito de que la licencia conste por escrito también proporciona mayor seguridad para el propio proveedor del software: aquel que utilice un software sin haber aceptado la licencia de uso no podrá alegar que el proveedor del software le había autorizado verbalmente a usarlo.



Decimos que la Ley de la Propiedad Intelectual exige que las licencias se formalicen por escrito, porque su artículo 45 establece que toda cesión de los derechos de explotación de una obra literaria, artística o científica (también el software), incluida la mera cesión del derecho de uso “deberá formalizarse **por escrito**”.

Por *forma escrita* no hemos de entender solo la licencia que se recoja en formato papel. También es válida la licencia que consta por escrito en **soporte electrónico**, como un archivo de texto.



El artículo 23 de la Ley 34/2002, de 11 de julio, de servicios de la sociedad de la información y de comercio electrónico (la tan conocida como controvertida LSSI) ha dado plena carta de validez a los contratos –entre ellos la licencia de software– celebrados por vía electrónica:

“Los contratos celebrados por vía electrónica producirán todos los efectos previstos por el ordenamiento ju-

rídico, cuando concurran el consentimiento y los demás requisitos necesarios para su validez [...]"

"Siempre que la ley exija que el contrato o cualquier información relacionada con el mismo conste por escrito, este requisito se entenderá satisfecho si el contrato o la información se contiene en un soporte electrónico."

Como **formas típicas** de una licencia de software tenemos:

#### a) Documento en papel

Distinguimos dos situaciones:

- Ambas partes (proveedor y usuario) firman un contrato de licencia. El contrato de licencia prevé en qué soporte se hace entrega del software al usuario, y si la instalación del software queda a cargo del proveedor o si lo instalará directamente el usuario. Esta forma es propia de licencias de programas con destino a empresas o profesionales, sobre todo cuando el programa requiere de ciertos ajustes para la "personalización" o "parametrización" a las necesidades del cliente (lo que se conoce como modalidad de licencia "llave en mano").
- Las cláusulas de la licencia se incorporan dentro del envoltorio que contiene el software. Esta forma es propia de licencias de programas comercializados en masa y para uso doméstico, que se distribuyen en soporte físico de disquetes o CD-ROM. Como veremos, esto da lugar a las licencias denominadas *shrink-wrap* según las cuales, la rotura del precinto del envoltorio que contiene el software implica la aceptación de las cláusulas de la licencia por el usuario.

#### b) Archivo electrónico

Puede tratarse de un archivo electrónico de **tipo texto** (.doc, .txt, o similar), o parte de un **ejecutable**. Al principio o hacia el final de la

#### Nota

Veremos las licencias *shrink-wrap* en la unidad 5.

**Nota**

En la unidad 5, en el apartado relativo a la “aceptación” de la licencia, analizaremos el debate sobre la validez o no de las licencias *shrink-wrap* y *click-wrap*.

**Nota**

Decíamos que el único requisito de forma que la ley exige a la licencia de software es que figure por escrito. No obstante, como todo contrato necesitará que conste una expresión formal del consentimiento de las partes. Cuando la licencia tenga el formato de un documento tradicional en papel, el consentimiento de las partes figurará mediante sus firmas respectivas. Más adelante veremos que en el resto de casos (firma electrónica, licencias *shrink-wrap*, *click-wrap*), se discute si cabe entender que el usuario-licenciatarario ha prestado válidamente o no su consentimiento.

**4.1.3. Software estándar**

La licencia de software es propia de aquel software “estándar”; es decir, no de un software que el proveedor crea “a medida” y por previo encargo de un cliente, sino del software cuya utilidad o aplicaciones han de servir, según su creador, para una pluralidad indeterminada de usuarios.

**Nota**

Las licencias de uso son propias del software que se desarrolla para distribuirlo en masa.

Sabemos que la mayor parte del software (al menos el que tiene mayor difusión) no lo crea un programador independiente para su exclusivo uso o el de unas pocas personas, lo crean empresas programadoras que tienen por actividad económica, precisamente, dedicarse a producir software, cuya utilidad y aplicaciones confían que han de satisfacer las expectativas y necesidades de multitud de usuarios.

Cuando un programador o empresa programadora crea un software por encargo de un cliente, en especial cuando lo crea adaptado y personalizado para satisfacer las instrucciones y necesidades exclusivas de ese cliente, la relación contractual entre ambas partes se regula por un “contrato de desarrollo de software”. En el derecho español, este contrato se encuadra en la categoría tradicional de los “contratos de arrendamiento de obra” (regulados por los artículos 1.588 a 1.600 del Código civil).



En todo caso, una de las cláusulas más importantes en un contrato de desarrollo de software es la que determina a quién le corresponde la propiedad del software, creado por el programador por encargo del cliente:

- Si se atribuye al **cliente**: en tal caso, el programador (autor del software) cede los derechos de explotación al cliente.
- Si se atribuye al propio **programador**: en tal caso, para que el cliente pueda utilizar el software, el contrato de desarrollo prevé que el programador conceda al cliente una licencia indefinida de uso. Se prevén asimismo cláusulas de mantenimiento a prestar por el programador y/o acceso al código fuente por el cliente.

También puede ser que el programador conserve en su poder aquella parte del software que emplea genéricamente en sus desarrollos (licenciándola al cliente); mientras que ceda los derechos de explotación sobre la parte verdaderamente “parametrizada” para el cliente.

Por tanto, las empresas creadoras de software producen un **software “estándar”** o **en serie**. No obstante, en muchos casos (software con aplicaciones especializadas, destinado a profesionales) pueden hacer algunas adaptaciones a las necesidades particulares de cada cliente: el proveedor dispone de un software “núcleo”, que proporciona al cliente tras efectuar ciertas adaptaciones a las necesidades de éste, bien en el código fuente (**customizaciones**, también conoci-

das como *modificaciones* en sentido estricto) o cambios de parámetros (**parametrizaciones**, también conocidas como *extensiones*). El software adaptado a las necesidades del cliente es, evidentemente, un software más complejo y de precio elevado.

**Nota**

El software estándar es distinto del que se crea por encargo del cliente. Sin embargo, un software estándar puede incorporar también adaptaciones a las necesidades particulares del usuario (*customización* o *parametrización*).

El objetivo de las empresas creadoras de software estándar es distribuir el software al mayor número de usuarios posible, para obtener la máxima divulgación del software y, por qué no, los mayores beneficios económicos posibles. Se habla, por tanto, de que este software se comercializa o distribuye **“en masa”**. En caso del “tradicional” software propietario, los ingresos son mayores cuantas más copias del programa consiguen comercializar, a cambio de una remuneración económica.

**Nota**

En este sentido, el software con destino a ser comercializado en masa, en especial entre los consumidores, no deja de ser como otro tipo de producto. Está claro que (al igual que, por ejemplo, la ropa, el calzado, etc.) es mucho más costoso un software “a medida” que uno producido en serie.

El negocio para la empresa creadora del software presupone que su software estándar es más que suficiente para cubrir las necesidades de los usuarios que lo adquieren. A los usuarios, por su parte, les es mucho más económico adquirir una licencia de software estándar que encargar a un programador que les haga, por ejemplo, un procesador de textos “a su medida”.

**Nota**

El software libre es también estándar.

Por lo que respecta al **software libre**, también se trata de un software “estándar”, cuyo autor original también pretende que se divulgue entre el mayor número de usuarios posible. Además, está dispuesto a que los usuarios modifiquen el software, lo adapten, lo mejoren e incluso lo redistribuyan con sus propias modificaciones.



Otra cosa distinta es que, en el software libre, o bien el autor no persigue obtener un beneficio económico (sino que lo pone a disposición de los usuarios por motivos altruistas, de conseguir reputación en la comunidad, etc.); o bien, si pretende obtener beneficios económicos, es a través de un modelo de negocio alternativo. A grandes rasgos, podemos decir que, de la obtención de ingresos por la venta de copias de software (software propietario), el negocio del software libre se desplaza, en su caso, a la prestación de servicios sobre el mismo, tales como de actualización o mantenimiento.



Por tanto, el software que se distribuye mediante licencias de uso (tanto el propietario como el libre) se ha creado para su distribución en masa a potenciales usuarios. Ello explica que las licencias de software acostumbren a estar compuestas por las denominadas **condiciones generales**.

#### 4.1.4. Contratación “en masa” y condiciones generales

Como hemos visto, la licencia de software suele ser un contrato (en España y en otros países de derecho continental, siempre será un contrato), en el que son partes del mismo tanto el proveedor del software como el usuario. Sin embargo, casi siempre es el licenciente o proveedor del software quien estipula unilateralmente los términos y condiciones de la licencia. En tal caso, el usuario no puede negociar con el licenciente las condiciones de la licencia, sólo puede limitarse a aceptarlas o rechazarlas.

Que los términos y condiciones de la licencia de uso se estipulen unilateralmente por el licenciente y éste los imponga a los usuarios es una consecuencia lógica al tratarse de un software estándar destinado a distribuirse “en masa”, sea o no para obtener un beneficio económico.

El proveedor del software, evidentemente, ni puede ni quiere negociar los términos de la licencia con cada uno de los centenares o miles de usuarios, sino al contrario: el proveedor del software quiere que todos los usuarios utilicen su software conforme a unas mismas condiciones impuestas por él. Así pues, todos los usuarios tendrán unos mismos derechos y limitaciones sobre el software. Si el usuario

**Nota**

Las licencias de uso de software se componen fundamentalmente de condiciones generales.

no acepta las condiciones, no adquiere la licencia y, por tanto, no podrá utilizar el software.



Cuando el proveedor del software impone a todos los usuarios unos mismos términos y condiciones de la licencia de software, que sólo puede aceptar (si desean utilizar el software) o rechazar, se dice que emplea **condiciones generales**.



El empleo de “condiciones generales” es propio de una sociedad moderna, industrial, en la que no sólo los programadores del software, sino que las empresas en general ofrecen sus productos o servicios “en masa” al mercado.

Si una empresa programadora tuviera que negociar sus licencias con cada potencial cliente, y entrar a discutir con él si incluye o no una cláusula, o bien no le saldría rentable producir software alguno, o bien debería cobrar unos precios astronómicos por la licencia.

Como hemos explicado, y como podemos ver al leer cualquier licencia, casi todas las cláusulas de las licencias de uso de software consisten en condiciones generales que se aplican por igual a todas las licencias que el proveedor concede sobre el mismo software.

En algunas ocasiones, las licencias cuentan también con **condiciones particulares**, que se aplican solamente a una relación contractual en particular: por ejemplo, si la licencia contiene alguna cláusula sobre adaptación (*customisation*) o parametrización a las necesidades específicas de un usuario concreto.

Es importante tener en cuenta que, al emplearse en la licencia de software condiciones generales, éstas han de ser respetuosas y cumplir ciertos requisitos legales. En España, como en el resto de Estados de la Unión Europea, rigen leyes sobre las condiciones generales en los contratos, que intentan proteger la posición del contratante que

se “adhiera” a las condiciones, frente a posibles abusos del empresario o profesional que impone dichas condiciones generales.



Existen leyes que regulan las condiciones generales en los contratos para impedir que el empresario que las impone las aproveche ilegítima o abusivamente en perjuicio de la parte más débil.

Como **requisitos generales**, se exige a las condiciones generales:

- Que se redacten con concreción, claridad y sencillez, debiendo constar en el mismo texto del contrato o en un documento aparte al que haga referencia el contrato y que se haya facilitado al consumidor previa o simultáneamente a la conclusión del contrato.
- Que la parte adherente tenga la oportunidad de conocerlas antes de aceptar el contrato.
- Que la parte que imponga las condiciones generales no puede beneficiarse de una redacción no clara o ambigua: en caso de duda sobre su interpretación, la cláusula deberá interpretarse en el sentido favorable al adherente.
- Que si una condición general tiene un contenido incompatible con una condición particular, prevalece la condición particular.
- Que si además el adherente es consumidor, como veremos, algunas cláusulas se consideran abusivas y no se pueden imponer, por entenderse que son injustas y desproporcionadamente desfavorables para el consumidor.

## 4.2. Función jurídica y económica

Una vez definida la licencia de uso de software, nos podemos preguntar:

- ¿Por qué se emplean las licencias de uso de software?

- ¿Para qué se usan las licencias de uso?

En relación con estas dos preguntas, nos podemos cuestionar también: ¿por qué la licencia de uso es el medio o instrumento legal habitual por el cual los proveedores distribuyen el software a los usuarios?



Precisamente, por **función jurídica y económica** nos referimos a aquellos derechos e intereses de las partes que, junto con las características técnico-jurídicas de la licencia de uso, explican que sea ésta el instrumento legal que los proveedores de software (estándar) emplean para distribuir los programas informáticos a terceros.

A la pregunta de por qué se emplean las licencias de uso como instrumento legal para distribuir software, hemos comenzado a responderla en la unidad 2 y en el apartado 1 de esta unidad: el software (sea propietario o libre) presenta unas características singulares, respecto a otros bienes y servicios. Características tanto técnicas como jurídicas, en particular, el ser objeto de protección legal por medio de los derechos de autor (propiedad intelectual). Analizaremos de nuevo todas estas características singulares del software, de forma conjunta, las cuales explican que el instrumento legal para su distribución sea la “licencia de uso”, en lugar de otro tipo de contratos o figuras jurídicas.

Una vez analizado “por qué”, las causas que fundamentan el empleo de las licencias de uso en lugar de otros contratos, la función jurídica y económica de las licencias de software también explica “para qué”, la finalidad de las licencias: se emplean para distribuir legalmente el software, conceder derechos a los usuarios e imponer limitaciones al uso del software. Resulta un buen instrumento legal para gestionar los derechos del propietario del software, a la vez que favorecen su difusión (con ánimo de lucro o no) a los usuarios.



Ante todo, podemos encontrar una explicación simple al hecho de que la licencia de uso sea el contrato habi-

tual elegido por los proveedores del software: es el contrato que más protege sus derechos y que más favorece sus intereses, en particular los económicos.

Todo lo cual referido, sobre todo, al mundo tradicional del software propietario. No obstante, como veremos, las licencias de software también se muestran como el instrumento legal adecuado para distribuir software libre. En este caso, no tanto para proteger los intereses de los proveedores de software, sino para asegurar las libertades de los usuarios.

#### 4.2.1. Consideraciones generales: la singularidad del software

En cualquier caso, sea software propietario o libre, los programas informáticos presentan unas **características singulares, tanto técnicas como jurídicas**, que explican que su distribución se efectúe mediante licencias de uso y no mediante otro tipo de contratos.

Entre las **características técnicas** que hacen singular al software en relación con otros bienes o productos, podemos citar las siguientes:

- a) Sabemos que el software es un bien **inmaterial** o **intangibile**: se trata de un bien con valor económico, pero consistente en una creación del intelecto humano, sin plasmarse en un elemento material.
- b) A partir de una copia del software se pueden hacer múltiples copias de igual calidad. Es más, si no fuera por la obligación de respetar los derechos de autor, para el que adquiere una copia de un software no existe ningún problema en hacer nuevas copias y cederlas (cobrando o no) a terceros, pues su ejemplar tampoco se ve perjudicado con la realización de copias.
- c) Asimismo, el software puede modificarse, dando lugar a un programa derivado (con mejoras, evoluciones, nuevas aplicaciones, errores corregidos, etc.) siempre que se disponga del código fuente.

**Nota**

Podéis ver el apartado 4.1.3 de esta unidad.

- d) El software evoluciona continuamente; o, dicho desde otra perspectiva, queda obsoleto con cierta rapidez.
- e) Para el usuario puede ser beneficioso poder modificar el software pero, ciertamente, ante todo el software le es útil por la aplicación o resultado que conlleva su uso. De este modo, el usuario puede decidir adquirir un software sabiendo que únicamente se le va a conceder un derecho limitado de uso.
- f) Como hemos visto, el software se crea en gran parte de forma estándar. A diferencia del software que se desarrolla por encargo de un cliente y personalizado a sus necesidades, el software estándar se distribuye “en masa”.

En cuanto a las **características de orden legal**, tenemos las siguientes:

- a) El software (en Europa, y hasta la fecha) se ha protegido fundamentalmente por derechos de autor, equiparado a las obras literarias. No obstante, el software difiere de estas obras, ya que los usuarios lo utilizan para obtener alguna utilidad práctica.
- b) Los proveedores del software hallan protección en la normativa sobre propiedad intelectual, de aplicación a obras artísticas o literarias: la Ley de la Propiedad Intelectual les concede **derechos exclusivos** para autorizar su uso, copia modificación y distribución.
- c) Además de los derechos de autor, los proveedores de software no están obligados por ley a divulgar el código fuente del programa; al contrario, tienen derecho a preservarlo en secreto.

Teniendo en cuenta estas características del software, veremos que la función jurídica y económica de las licencias de software difiere según si se trata de una licencia de software propietario “tradicional” o de una licencia de software libre.

#### **4.2.2. Función jurídica y económica en las licencias de software propietario**

Precisamente las empresas que desarrollan software propietario se benefician al máximo de los derechos exclusivos de explotación que les concede la legislación sobre derechos de autor (en España, la Ley de la Propiedad Intelectual). Cuando se persigue obtener el máximo

rendimiento económico de un software propietario, las empresas programadoras basan su modelo de negocio en la comercialización de copias, cuantas más, mejor.

Por ello, las licencias de software propietario tienen tradicionalmente un contenido y alcance muy restrictivos de las facultades que sobre el software se conceden al usuario, así como muy protectores de los derechos en exclusiva del autor. La propia Ley de la Propiedad Intelectual es también muy protectora del titular del software y de sus derechos exclusivos.



De este modo, el proveedor permite el uso del software al usuario, pero con muchas limitaciones; además, le prohíbe modificar y redistribuir el software, derechos que el proveedor se reserva para sí. Estas restricciones le sirven al proveedor para poder maximizar sus ingresos por la comercialización de copias.

En este sentido, los proveedores de software propietario pretenden mantener a toda costa la exclusiva y el control sobre la comercialización de las copias del software que desarrollan. No quieren que un usuario sea “propietario” de una copia del software, sino un simple facultado para usarlo, porque si fuera propietario, éste podría revender libremente su copia. Para ello, los proveedores de software no venden una copia de software al usuario, solamente le permiten utilizar el software mediante una licencia.

Hemos de tener presente que, aunque la Ley de la Propiedad Intelectual conceda al proveedor del software el derecho exclusivo a autorizar su copia y distribución, también es cierto que limita el derecho de distribución a la primera venta de una copia.



Así, en el último párrafo del artículo 99 de la Ley de la Propiedad Intelectual establece que:

“La primera venta en la Unión Europea de una copia de un programa por el titular de los derechos o con su con-

sentimiento, agotará el derecho de distribución de dicha copia, salvo el derecho de controlar el subsiguiente alquiler del programa o de una copia del mismo.”

Por tanto, si un proveedor del software “vendiera” copias, el comprador podría distribuirla libremente a terceros, por lo que el proveedor perdería los posibles ingresos por nuevas copias. Y el software, técnicamente, es fácilmente reproducible en múltiples copias, todas ellas de igual calidad.

¿Cómo preservan los proveedores del software la exclusiva en la distribución de copias? Para conseguir que todos y cada uno de los usuarios de un programa informático “pasen por caja”, lo que se hace es no vender software, no conceder derecho de propiedad alguno al usuario sobre el software: solamente conceden el derecho a usarlo, pero el único propietario de la copia del software sigue siendo el proveedor.



Esto se traduce en que las licencias de software propietario suelen prohibir o restringir al usuario el derecho de ceder su copia a otro usuario (salvo las licencias de software *freeware*, *shareware* y otras excepciones): ni venderla, ni regalarla, ni alquilarla o prestarla a un tercero.

Por otra parte, las restricciones de modificación del software al usuario, junto con la negativa a proporcionarle el código fuente, son también importantes para los intereses del proveedor de software propietario. Impide o dificulta, por un lado, que otros puedan plagiarle el programa y crear otro con prestaciones idénticas. Por otro lado, también se reserva la exclusividad en la prestación de servicios de actualización o mantenimiento, por los que también cobra a los usuarios.

Es más, cuando un proveedor comercializa nuevas versiones de un software, en muchas ocasiones deja de prestar el servicio de mante-



nimiento sobre las versiones antiguas. Así, fuerza a los usuarios a adquirir una nueva licencia.

#### 4.2.3. Función jurídica y económica en las licencias de software libre

El software libre también se divulga y distribuye a los usuarios con licencias de uso. Ello, a pesar de que el proveedor del software, cuando decide distribuirlo como libre, o bien no persigue un beneficio económico (lo distribuye por razones altruistas, por conseguir notoriedad o reputación, etc.); o, si lo persigue, no lo va a obtener por la comercialización de copias, sino a partir de un modelo de negocio distinto: la prestación de servicios sobre el software, tales como actualización, consultoría o mantenimiento.

De una forma un tanto paradójica, las licencias de software, que tradicionalmente se han empleado para restringir al máximo los derechos del usuario, resultan también un medio adecuado para garantizar a los usuarios de software libre sus derechos sobre el mismo.



En el caso del software libre, las licencias cumplen una función totalmente distinta. No se emplean para proteger derechos exclusivos del proveedor, o para lograr que se comercialice el mayor número de copias posible. En el software libre, las **licencias** conceden y aseguran a los usuarios **las libertades de uso, modificación (adaptación y mejoras) y redistribución**.

Así pues, las licencias de software libre conceden muchos más derechos al usuario y, por el contrario, las restricciones y prohibiciones son mucho menores que las recogidas en las licencias de software propietario. Es más, dichas restricciones consisten más bien en “condiciones” para el ejercicio de los derechos, que no pretenden proteger derechos de explotación exclusivos del proveedor, sino preservar su reputación y garantizar que todos los usuarios del software podrán beneficiarse de dichas libertades, impidiendo posibles intentos de apropiación del software.

**Ejemplo**

Como ejemplos de restricciones en las licencias de software libre, tenemos la obligación impuesta a los usuarios de distribuir en todo momento el software con el código fuente; o, en el caso de la GNU-GPL, tenemos también la cláusula de *copyleft*, que permite a los usuarios modificar y distribuir el software (con o sin modificaciones) siempre que éstos lo distribuyan también como software libre.

Nada que ver, por tanto, con las estrictas prohibiciones de modificar, copiar o ceder la copia que contemplan las licencias de software propietario, las cuales tienen por objetivo preservar exclusivamente los intereses del proveedor.

En particular, al distribuir el software libre mediante licencias de uso basadas en el reconocimiento del autor:

- Se impide que alguien intente apropiarse del software libre.
- A su vez, los usuarios tienen la garantía de que han obtenido el software con la autorización de quien tenía los derechos de explotación originales; es decir, que usan, modifican, copian y/o redistribuyen el software legalmente, sin que nadie pueda acusarles de infringir derechos exclusivos.
- Dado que los derechos morales son irrenunciables según la Ley de la Propiedad Intelectual, el empleo de licencias de uso permite reconocer en todo momento quién es el autor del software y preservar su reputación.

### 4.3. Naturaleza jurídica

En esta unidad hemos visto que la licencia de uso de software es un contrato. Ahora bien, ¿qué tipo de contrato?

La naturaleza jurídica del contrato de licencia de software consiste en determinar si dicha licencia se ha de equiparar o no a alguno de los

contratos “clásicos”. Es decir, determinar si, cuando un proveedor de software concede una licencia a un usuario, debemos entender que nos encontramos ante un contrato equivalente, por ejemplo, a una compraventa, un arrendamiento o una prestación de servicio. O bien si se trata de un contrato completamente independiente, *sui generis*, que no tiene nada que ver con otro tipo de contratos.



El **contrato de compraventa** consiste en que el vendedor entrega una cosa (y la propiedad de la misma) al comprador a cambio de que éste le abone una cantidad de dinero (precio) determinada.

Por su parte, el **arrendamiento** o **alquiler** consiste en la cesión temporal por el arrendador del derecho a usar una cosa de su propiedad a favor del arrendatario; éste abona a cambio cierta cantidad de dinero (normalmente en forma de renta o pagos periódicos) y al finalizar el contrato debe devolver la cosa al arrendador.

En el supuesto de aquellas licencias que se conceden de forma gratuita (en algunos casos de software propietario, como el *freeware*, *shareware*, etc.; y en gran parte del software libre), la cuestión se centra en determinar si la licencia de uso de software se puede equiparar a un **préstamo** o a una donación.

Cuando el software se adapta o parametriza a las necesidades del usuario, también podemos asimilarlo a la realización de una obra o prestación de servicio.

La cuestión de la naturaleza jurídica se plantea porque no hay ninguna norma jurídica (al menos en España ni en el conjunto de la Unión Europea) que se encargue de regular de manera específica y completa las licencias de uso de software, a diferencia de los contratos “clásicos”, que sí que se regulan por ley. De este modo, aunque en principio determinar la naturaleza jurídica de las licencias de software se trataría de un debate propio de juristas, el hecho de que no exista una ley o norma que las regule por completo conlleva

que la resolución de dicha cuestión tenga una consecuencia práctica importante:

- Si entendemos que el contrato de licencia de software equivale a un **contrato “clásico”** de compraventa, de arrendamiento (en su caso, a un préstamo o a una donación), o bien a una realización de obra o prestación de servicio, la licencia se regirá por las normas jurídicas que regulen dichos contratos: en España, fundamentalmente el Código civil y el Código de Comercio.
- Por el contrario, si se entiende que es un **contrato *sui generis***, no le serán aplicables las normas propias de otros contratos. La licencia de software se regirá ante todo por lo pactado entre las partes y por las normas generales sobre obligaciones y contratos, pero no por las normas que regulan contratos específicos, tales como la compraventa, el arrendamiento, el préstamo, la prestación de servicios, etc.

Según lo dicho, saber cuál es el régimen legal que se ha de aplicar al contrato de licencia de uso es importante. Aunque la licencia de uso se rija, en principio, por sus propios términos y condiciones, también se regirá por ciertas leyes y demás normas legales. Las normas legales que se aplican a las licencias de uso de software serán distintas en función de si se consideran equivalentes a otro contrato (se aplicarán las normas de ese contrato), o bien si se califican como un contrato distinto.

En cualquier caso, conviene distinguir entre aquellas **normas legales**:

- a) **Imperativas**: aquellas normas que rigen obligatoriamente entre licenciante y licenciataria en cualquier caso. Si en el contrato de licencia han pactado una cláusula contraria a una norma imperativa que le sea aplicable, esa cláusula será nula.

#### Ejemplo

Si en la licencia figura una cláusula por la que el proveedor del software se exime de prestar cualquier garantía o responsabilidad, será una cláusula nula porque existen leyes que prohíben este tipo de cláusulas.

- b) **Dispositivas:** aquellas normas que rigen entre licenciante y licenciario para el caso de que no hayan pactado otra cosa distinta en el contrato de licencia.

**Ejemplo**

Si no se pacta expresamente en el contrato una cláusula de exclusividad, el artículo 99 de la Ley de la Propiedad Intelectual entiende que la licencia se concede al licenciario sin derecho de exclusividad.



En este sentido, la opinión mayoritaria actualmente considera que la licencia de software se trata de un contrato *sui generis*, atípico, distinto a cualquier otro contrato (de compraventa, arrendamiento de cosas, préstamo, donación, prestación de servicios, etc.).

En relación con el **contrato de compraventa**, sabemos precisamente que el conceder una licencia de software tiene como uno de los motivos principales evitar que pueda hablarse de “venta” del software (hablamos aquí de software “propietario”). Con ello, el licenciante conserva en todo momento los derechos de propiedad sobre el software, y el control sobre su copia y distribución. Es decir, deja bien claro al usuario que no le “vende” el software, sino que únicamente le permite ciertos usos limitados del mismo.

La licencia de software tampoco puede asimilarse por completo a un **contrato de arrendamiento** (alquiler): en muchos casos, la licencia se concede por tiempo indefinido, mientras que en los arrendamientos se establece necesariamente un plazo de tiempo determinado para usar la cosa alquilada. Y, aun en los supuestos en que la licencia de software se concede por un plazo determinado, lo que el licenciante permite al usuario-licenciario son derechos limitados sobre un bien inmaterial como es el software, mucho menores que los que se suelen conceder en un arrendamiento. Además, cuando un contrato de arrendamiento finaliza, el arrendatario ha de devolver la cosa arrendada al arrendador. En el caso de las licencias de software, si bien a veces se exige que el usuario devuelva la copia al licenciante al finalizar el contrato, en muchas ocasiones el usuario nada ha de devolver, sino destruir, borrar y/o desinstalar definitivamente el programa.

En el caso de programas *shareware* o *demos*, la licencia podría asimilarse a un préstamo (que consiste en la cesión temporal de una cosa de forma gratuita), si bien la equiparación tampoco puede ser absoluta: no se cede una cosa, sino un derecho, y posiblemente no se deba devolver físicamente nada al licenciante al finalizar el período de vigencia (sino borrar el programa, e incluso éste queda a veces automáticamente deshabilitado).

En los supuestos de licencias de programa *freeware*, o de software libre que se conceden gratuitamente, podrían asimilarse a una donación: se permite el uso (un derecho, no una cosa) de un software de forma gratuita e indefinida). En todo caso, podrían tratarse de donaciones modales (cuando se regala algo a alguien, pero a cambio el beneficiado con la donación debe cumplir con una condición o gravamen, si bien dicha condición o gravamen le supone menos perjuicio que el valor de lo que recibe), porque tales licencias siempre imponen ciertas condiciones al licenciario, como por ejemplo la cláusula de *copyleft*.

No obstante, las antiguas normas del Código civil sobre donaciones, que datan del siglo XIX, difícilmente podrán ser adecuadas y suficientes a las situaciones que plantea una licencia de uso de software.



Si bien la opinión mayoritaria considera las licencias de software como un contrato *sui generis*, que se regirá fundamentalmente por lo pactado entre las partes, por las normas genéricas aplicables a todos los contratos y por las normas sobre derechos de autor, también se afirma que, según cada caso concreto, podrán aplicarse (puntualmente) algunas de las normas que regulan la compraventa, el arrendamiento de cosas o de obra, el préstamo, la donación, la prestación de servicios, etc.

En todo caso, la aplicación —a las licencias de uso de software— de las normas que regulan otros contratos, no se produciría con carácter general, sino para supuestos puntuales y por **aplicación “analógica”** de dichas normas: para resolver aquellas controversias que deriven de aspectos no regulados en la propia licencia, o bien que se hayan regulado por unas cláusulas contractuales ambiguas, difíciles de interpretar; o bien porque una cláusula de la licencia se

considere nula por vulnerar una norma imperativa (por ejemplo, si cabe aplicar a las licencias de software la normativa sobre garantías en materia de compraventa).

La aplicación, por analogía, de las normas que regulan un contrato u otro dependerán de las circunstancias concretas de cada caso: algunas licencias de software se asemejan más a una compraventa (software comercializado en masa), otras a un arrendamiento o prestación de servicios (software parametrizado), otras a una donación o un préstamo (software distribuido gratuitamente).

Así, tanto en España como en otros países (Reino Unido, EE.UU.) hay tribunales que han resuelto litigios sobre licencias de software por aplicación analógica de las normas que regulan la compraventa (por ejemplo, sobre las garantías del vendedor respecto el buen estado y funcionamiento de la cosa vendida). En especial, se dice que las normas sobre contratos de compraventa pueden aplicarse analógicamente a aquellas licencias de software que por sus términos y condiciones se asemejen más a una compraventa, sin llegar a serlo.

#### Ejemplo

Pensemos en aquellas licencias de software estándar comercializadas “en masa” para usuarios consumidores, que se conceden por tiempo indefinido y abonando el usuario-licenciatario un precio a tanto alzado.

En otros supuestos, podrán considerarse de aplicación analógica normas puntuales relativas a los contratos de arrendamiento o de prestación de servicios (por ejemplo, la garantía de que la cosa arrendada es propiedad del arrendador, por lo que ningún tercero podrá impedir al arrendatario usar la cosa).

#### Ejemplo

Pensemos en aquellas licencias de software creado por el licenciante pero adaptado –parametrizado– a las necesidades del usuario, que se conceden por un tiempo determinado, por la que el usuario abona una cuota periódica, y al final de la cual debe devolver el soporte (CD-ROM) al proveedor.

**Nota**

En la unidad 6 analizaremos las normas sobre los derechos exclusivos de los licenciantes contenidas en la Ley de la Propiedad Intelectual.

Cuando las licencias se conceden gratuitamente, podrán considerarse de aplicación analógica las normas que regulan la donación o el préstamo, según si la licencia se concede por tiempo indefinido o por un plazo determinado.

También hemos de tener en cuenta que la Ley de la Propiedad Intelectual regula, al menos en parte y de forma indirecta, el posible contenido de un contrato de licencia de uso de software, pues la Ley de la Propiedad Intelectual contempla la serie de normas sobre los derechos exclusivos de los licenciantes respecto al software, sobre el modo de ceder tales derechos; así como sobre determinadas facultades, prohibiciones y límites que puede imponer o no al usuario-licenciario.

Por tanto, las normas de la Ley de la Propiedad Intelectual se aplicarán a los contratos de licencia de software en cualquier caso y con preferencia a otras normas.

#### 4.4. Marco normativo

Sabemos que el software se halla protegido fundamentalmente por las normas que regulan los derechos de autor. Por lo que respecta a las licencias de uso, en concreto, hemos expuesto en el apartado 3 que no existe una norma que regule por completo el contenido al que han de ajustarse; sin embargo, según las circunstancias y pactos particulares de cada licencia de uso, podrán serle de aplicación analógica las normas que regulan otros contratos.

Por tanto, nos preguntamos: ¿qué leyes, qué normas se aplican a las licencias de uso de software? Es decir, ¿cuál es su régimen legal o marco normativo? Determinar el marco normativo aplicable a una licencia de uso es importante, tanto las **normas imperativas** (de aplicación obligatoria, prevaleciendo incluso por encima de las cláusulas de la licencia), como las **normas dispositivas** (de aplicación sólo en caso de que no exista ninguna cláusula de la licencia que rijan el mismo punto).



Por **marco normativo** entendemos el conjunto de normas jurídicas (convenios internacionales, leyes, regla-



mentos, etc.) que son de aplicación a los contratos de licencia de software.

**Nota**

En cada país, el marco normativo aplicable varía. En este apartado nos vamos a centrar en las normas aplicables a las licencias de software en España.

No obstante, cabe precisar que el régimen jurídico en España es sustancialmente similar al del resto de estados de la Unión Europea, por cuanto gran parte de las leyes españolas que se citan a continuación son adaptación de directivas de la Unión Europea.

La norma de aplicación principal a las licencias de software en España es la **Ley de Propiedad Intelectual** (Real Decreto Legislativo 1/1996, de 12 de abril, por el que se aprueba el Texto Refundido de la Ley de Propiedad Intelectual, regularizando, aclarando y armonizando las disposiciones legales vigentes sobre la materia). Cabe destacar de la misma el libro I título VII (artículos 95 a 104), relativos a los “programas de ordenador”. Si bien esta ley no regula directamente las licencias de uso, regula los derechos –exclusivos– que tienen los autores de un software sobre dicho programa.

**Nota**

En el apartado 2.2 de la unidad 2 se da una exposición pormenorizada de los orígenes de la Ley de la Propiedad Intelectual, las leyes anteriores que regían la propiedad intelectual en España, así como los convenios internacionales y directivas de la Unión Europea que han dado lugar a la misma.

Asimismo, la Ley de la Propiedad Intelectual prevé que el autor o, en general, el proveedor del software, pueda ceder el uso del programa a terceros mediante licencias. La Ley de la Propiedad Intelectual contempla qué derechos puede ceder el proveedor del software, qué prohibiciones y límites puede imponer a los usuarios-licenciarios y qué derechos de los usuarios-licenciarios no pueden limitarse.



En particular, podemos destacar el último párrafo del artículo 99 de la Ley de la Propiedad Intelectual, que contempla una norma específica para las licencias de uso de software:

“A tales efectos, cuando se produzca cesión del derecho de uso de un programa de ordenador se entenderá, salvo prueba en contrario, que dicha cesión tiene carácter no exclusivo e intransferible, presumiéndose, asimismo, que lo es para satisfacer únicamente las necesidades del usuario.”

#### Nota

Ver el análisis en detalle de los derechos, límites y prohibiciones de los usuarios-licenciarios en la unidad 6.

#### Nota

Podéis ver las condiciones generales en el apartado 1.4.

Otras normas que son de aplicación a las licencias de software (o a parte de ellas) son las siguientes:

a) **Ley 7/1998, de 13 de abril, sobre Condiciones Generales de la Contratación.** Se aplica a casi todas las licencias de software, en la medida en que éstas contienen condiciones generales (cláusulas de la licencia prefijadas por el proveedor del software para ser impuestas a todos los usuarios). Por tanto, esta ley es de aplicación a aquellas licencias de software que se distribuyen en masa.

Como hemos visto, esta ley exige que las condiciones generales cumplan ciertos requisitos, para proteger con ello la posición del adherente (en las licencias de software, el usuario), ya que dichas condiciones generales se redactan y se imponen unilateralmente por el proveedor del software.

b) **Ley 26/1984 (reformada el 1998), de 19 de julio, General para la Defensa de los Consumidores y Usuarios.** Se aplica sólo a aquellas licencias de software, cuyo usuario-licenciario sea un consumidor. En estos casos, sus normas son de especial importancia, pues impiden que el proveedor-licenciante pueda imponer cláusulas abusivas al usuario-licenciario (consumidor).

Al igual que la Ley sobre Condiciones Generales de la Contratación, la prohibición de cláusulas abusivas que contempla la Ley

General para la Defensa de los Consumidores y Usuarios tiene como finalidad proteger la posición del consumidor; en este caso, el usuario de un software, frente a cláusulas impuestas por el licenciante que supongan a favor de éste un injusto “desequilibrio de las prestaciones”.

**Nota**

En el apartado 5.1.2 analizaremos con más detalle la incidencia de esta ley en aquellas licencias de software cuyo licenciatario es un consumidor.

- c) **Ley 34/2002, de 11 de julio, de servicios de la sociedad de la información y de comercio electrónico (la conocida LSSI):** se aplica a aquellas licencias de software que se adquieren a través de Internet (*click-wrap licenses*). En particular, son de aplicación los artículos 23 a 29 de la LSSI, que establecen ciertas obligaciones de información y pasos a seguir en el proceso de contratación por vía electrónica. Dichas obligaciones son aún mayores cuando el usuario es un consumidor.

Tal y como hemos comentado, casi todas estas leyes españolas tienen su origen en directivas de la Unión Europea. Las directivas consisten en normas que la Unión Europea dicta a sus Estados miembros para que unifiquen su régimen jurídico respecto a una determinada materia. Las directivas logran unificar las diversas normas existentes en los estados miembros, imponiendo a éstos la obligación de dictar, en un plazo máximo de tiempo, una nueva ley cuyas normas correspondan con las de la directiva.

Por lo tanto, las leyes que se promulgan en España a consecuencia de una directiva de la Unión Europea tienen un contenido muy similar a las leyes correspondientes en los otros estados de la Unión Europea.

**Nota**

La LSSI se aplica a las licencias *click-wrap*.

**Ejemplo**

Así, tenemos:

- a) La legislación sobre condiciones generales de la contratación y sobre protección de los consumidores frente a cláusulas abusivas tiene su origen en la Directiva 93/13/CEE, del Consejo, de 5 abril 1993,

sobre cláusulas abusivas en los contratos celebrados con consumidores.

- b) La LSSI tiene su origen en la Directiva 2000/31/CE, del Parlamento europeo y del Consejo, de 8 de junio del año 2000, relativa a determinados aspectos de los servicios de la sociedad de la información, en particular, el comercio electrónico en el mercado interior.
- c) Por lo que respecta a la Ley de Propiedad Intelectual, también tiene como antecedente algunas directivas de la Unión Europea, como la Directiva 93/98/CEE, del Consejo, de 29 de octubre, relativa a la armonización del plazo de protección del derecho de autor y de determinados derechos afines; o la Directiva 91/250/CEE, de 14 de mayo, sobre la protección jurídica de programas de ordenador. En cualquier caso, la Ley de la Propiedad Intelectual también tiene su origen en convenios internacionales, más allá del marco de la Unión Europea, como la Convención de Berna para la Protección de las Obras Literarias y Artísticas; y la Convención Universal sobre Derechos de Autor.

De este modo, es necesario resaltar que si bien a lo largo de estas unidades nos referimos continuamente a la Ley de la Propiedad Intelectual española, el régimen legal que establece la Ley de la Propiedad Intelectual sobre derechos de autor respecto el software es sustancialmente idéntico al del resto de la Unión Europea.

Por otra parte, y según hemos expuesto en el apartado 3 de esta unidad, relativo a la naturaleza jurídica del contrato de licencia de software, también pueden ser de aplicación analógica las normas del **Código civil** y del **Código de comercio mercantil** sobre la compraventa (para aquellas licencias de software estándar comercializadas “en masa” a usuarios consumidores, concedidas por tiempo indefinido y abonando el usuario-licenciatario un precio a tanto alzado); o, en su caso, las normas del **Código civil** relativas al arrendamiento de bienes o prestación de servicios (para aquellas licencias de soft-

ware creado por el licenciante pero parametrizado a las necesidades del usuario, que se conceden por X años y por las que el usuario abona una cuota mensual o anual).

Si las licencias de uso son gratuitas, podrían aplicarse las normas del Código civil que regulan el contrato de préstamo o a la donación.

No obstante, la aplicación por analogía de estas normas, del Código civil o del Código mercantil, no será automática y obligatoria, sino que dependerá de cada caso: en función de si lo pactan expresamente las partes o de si lo decide un Juez en un litigio.

En última instancia, sí serán de aplicación las **normas y principios generales** sobre las obligaciones y los contratos, que contemplan tanto el Código civil y el Código de comercio. Cabe destacar normas y principios tales como:

- Cumplir y ejecutar los contratos de buena fe.
- Responsabilidad por el incumplimiento doloso o negligente de las obligaciones contractuales.

## 4.5. Licencias de software especiales

Las licencias de software especiales son algunos tipos de licencias que contienen alguna especialidad respecto la licencia de uso típica.

En este apartado nos referimos a licencias de software (generalmente sobre software propietario) que incluyen alguna particularidad de especial importancia, con relación a las licencias de uso habituales. Entre estas licencias “especiales” podemos mencionar: la licencia “llave en mano”, la licencia de uso especial para *upgrade*, y la licencia de uso especial para *outsourcing*.

### 4.5.1. Licencia “llave en mano”

La especialidad de la licencia “llave en mano” es que se pacta expresamente una garantía o resultado concreto: que el software satisfaga

#### Nota

En las unidades 7 y 8 analizaremos con detenimiento las licencias de software libre y sus modalidades principales.

las necesidades concretas del usuario-licenciatarario. La licencia “llaves en mano” puede, en principio, recaer tanto sobre software propietario como sobre software libre.

Esta especialidad no se da en el software de uso doméstico comercializado en masa, sino en el caso de software de aplicaciones complejo, destinado a empresas. No se trata de un software que el proveedor desarrolle por encargo del usuario (en cuyo caso estaríamos ante un contrato de desarrollo de software), sino que el proveedor licencia al usuario un software estándar de su titularidad.

Sin embargo, aun siendo estándar, el proveedor sí que se encarga de realizar una serie de adaptaciones al software para adecuarlo a las necesidades concretas del usuario, según lo que éste le haya solicitado. Estas adaptaciones se conocen con el término de **parametrizaciones** o **extensiones** (cuando no modifican parte del código fuente), o **customizaciones** o **modificaciones** (cuando conllevan la adaptación del código fuente), siendo estas últimas las que caracterizan las licencias “llave en mano”.

En tal caso, en la licencia de uso se pactan servicios adicionales como la “instalación” del software, que queda a cargo del proveedor; así como la existencia de un **período de prueba y/o tests de conformidad**, durante los cuales el usuario comprueba que el software funciona de acuerdo con sus necesidades. Una vez transcurrido el período de prueba o superado con éxito el test de conformidad, el usuario da su aceptación al software y se inicia propiamente la licencia. Claro está, el usuario-licenciatarario abona un precio por la licencia “llave en mano” bastante mayor al que pagaría por la licencia de un software puramente estándar.



En las licencias “llave en mano” constituye un pacto esencial que el software ha de satisfacer necesidades específicas del usuario. Por ello, el software estándar deberá ser adaptado al usuario.

En estas licencias “llave en mano”, aunque el software pueda ser propietario, el hecho de incorporar adaptaciones para adecuarlo a

las necesidades del usuario puede suponer también que el usuario-licenciatarario tenga un derecho limitado a modificar el software y de acceso al código fuente; en particular, para crear, modificar o suprimir las customizaciones o parametrizaciones mencionadas.

#### 4.5.2. Licencia de uso especial para upgrade

Esta licencia especial requiere, más bien presupone, que existe una licencia de uso de software anterior, y consiste en la autorización concedida por el proveedor al usuario para que éste use el software en unas circunstancias distintas a las iniciales, a cambio de una cantidad adicional.

En la práctica, esta licencia especial para *upgrade* puede consistir tanto en un contrato distinto a la licencia de uso original, como de un anexo a la misma. Entre las nuevas circunstancias en que el usuario puede utilizar el software gracias a la licencia para *upgrade* tenemos:

- El uso del software en un ordenador distinto, de mayor potencia o capacidad (por ejemplo, de un PC a un *mainframe* o servidor, etc.).
- En un sistema operativo distinto del previsto en la licencia original.
- El uso por un mayor número de usuarios u ordenadores superior al previsto en la licencia original: por ejemplo, permitir que el software pase a estar disponible para todos los ordenadores conectados a la red interna del usuario.

#### 4.5.3. Licencia de uso especial para outsourcing

Se trata de las licencias de uso de un software que el proveedor concede a una empresa que presta servicios de *outsourcing* informático; esto es, una empresa que se encarga de prestar servicios de gestión informática a otras empresas.

En este caso, la licencia presenta la especialidad de que el software objeto de la misma no tiene como usuario final al licenciatarario (la empresa de *outsourcing*) sino a los clientes del licenciatarario, quienes no adquieren licencias de uso del software.



En las licencias de uso especial para *outsourcing*, el licenciatario quiere utilizar el software para prestar servicios a terceros.

Así pues, se trata de una excepción a la regla general que prevé el art. 99 de la Ley de la Propiedad Intelectual: la licencia de uso de software especial para *outsourcing* no tiene por objeto “satisfacer las necesidades personales del usuario”, sino la de los clientes del licenciatario.

Evidentemente, el precio que el licenciatario abona por la licencia al proveedor es elevado, porque el software lo destina a todos sus clientes. Es el licenciatario el responsable frente al proveedor por las vulneraciones de los derechos exclusivos sobre el software que puedan cometer sus clientes.

No obstante, en el contrato de prestación de servicios de gestión informática que vincula a la empresa de *outsourcing* con sus clientes, se les impone a éstos que deberán indemnizar a la empresa de *outsourcing* por cualquier reclamación que sufra del proveedor del software, debido a una vulneración de los derechos exclusivos sobre el software (por un uso, copia o redistribución no autorizada) cometida por el cliente o por sus empleados.

#### 4.6. Contratos accesorios a la licencia de software

Junto a las licencias del software de cierta complejidad destinado a empresas, aparecen unos contratos de prestación de servicios sobre el software, que denominamos **contratos “accesorios”** a la licencia de software porque **tienen como presupuesto de su existencia una licencia de software, a la que están vinculados.**

Dichos contratos pueden constar en un documento aparte de la licencia de uso; pero también como parte del contrato de licencia, ya sea dentro de sus cláusulas o en un anexo.



De estos contratos accesorios destacamos el contrato de mantenimiento, el contrato de consultoría y formación (a veces unido con el de mantenimiento) y el escrow.

#### 4.6.1. Contrato de mantenimiento

El software es relativamente inestable en su funcionamiento, y sus posibles fallos no son fáciles de arreglar, y menos aún si el usuario no dispone del código fuente; además, se queda obsoleto con cierta rapidez. De este modo, y una vez acabado el período de garantía que el proveedor ofrece sobre el software en la licencia, para el usuario puede ser imprescindible el mantenimiento del software, sobre todo en software de cierta complejidad y destinado a empresas y profesionales.

Para el proveedor del software, prestar el servicio de mantenimiento le va a suponer una fuente complementaria, e incluso muy importante, de ingresos. Además, prestar el servicio de mantenimiento va a permitir al proveedor controlar el buen funcionamiento del software, mejorarlo y arreglar aquellos fallos que le comuniquen los usuarios.

#### Definición

Mediante el **contrato de mantenimiento**, el prestador del servicio se obliga frente al usuario a mantener el correcto funcionamiento del software y/o a proporcionar sus sucesivas nuevas versiones, a cambio del pago de una cuota de mantenimiento (anual, trimestral, mensual, etc.).

#### Prestador del servicio: en software propietario y software libre

Sobre el prestador de este servicio, hemos de tener presentes los aspectos siguientes:

- a) En el caso del **software propietario**, el servicio de mantenimiento sólo puede prestarlo el proveedor del software o alguien autorizado por el proveedor: es el único que dispone del código fuente y el único con derecho a modificar el software licenciado.

Puede ser que, aun siendo formalmente un contrato accesorio a la licencia de uso, el mantenimiento constituya a veces mayor negocio para el proveedor que no el precio que obtiene por la licencia.

**Ejemplo**

Pensemos en ciertos programas de software especializado, complejo, con aplicaciones destinadas a profesionales. El usuario se ve obligado a contratar el mantenimiento durante el tiempo en que piense que el software le va a ser útil (uno, dos, tres años, etc.), para garantizarse que cualquier defecto en el funcionamiento del software le va a ser reparado.

Como el único que puede prestar dicho servicio es el proveedor, se dice que el contrato de mantenimiento permite al proveedor mantener cautivo al usuario.

**Nota**

El excesivo celo que los proveedores de software muestran en no revelar su código fuente, no siempre se debe sólo a evitar plagios del programa; en algunos casos, se debe también a su deseo de ser el único que pueda prestar los servicios de mantenimiento sobre el software, lo que le va a reportar también ingresos importantes: el modelo de negocio de estos proveedores se basa, en tales supuestos, tanto en la comercialización de licencias como en la prestación del servicio de mantenimiento.

En algunos países, como el Reino Unido, se ha discutido sobre si es legal que el proveedor obligue al usuario a contratar con él el servicio de mantenimiento, impidiendo que lo presten terceros: se considera que ello puede comportar una restricción ilegal de la libre competencia, o suponer competencia desleal. Incluso, ha habido ya algún litigio en el que los tribunales han obligado al proveedor de un software a ofrecer una “licencia de modificación” a terceros prestadores de servicios de mantenimiento.

- b) En el caso del **software libre**, el prestador del servicio de mantenimiento puede ser cualquier informático o empresa de servicios informáticos, puesto que puede disponer libremente del código fuente. De este modo, el mantenimiento se presta en régimen de libre competencia.

Curiosamente, existen modelos de negocio que tienen como esencial la prestación del servicio de mantenimiento, tanto respecto a cierto software propietario como en el software libre.

En cualquier caso, en el software propietario, el negocio es para el proveedor del software gracias a sus derechos de exclusividad (y a no revelar el código fuente) y, como se ha comentado, a la posibilidad de mantener “cautivo” al usuario. Puede optar por cobrar un precio relativamente barato por la licencia, a cambio de asegurarse el pago de una cuota de mantenimiento durante un largo tiempo.

En el software libre, el modelo de negocio puede basarse en la prestación de servicios como el de mantenimiento, pero en este caso por una razón diferente. No se trata de que sea un mayor ingreso para el titular del software, pues no existe un derecho de exclusividad sobre el mismo (por tanto, los ingresos por la concesión de licencia son nulos o mínimos) y los servicios de mantenimiento se prestan en libre competencia.

Por tanto, en el caso del software libre, las empresas informáticas pueden competir entre ellas por la prestación de un servicio de mantenimiento mejor, más barato, más fiable, etc. No obstante, a la hora de prestar dicho servicio, el creador del software siempre puede tener una ventaja competitiva sobre el resto, al ser en la práctica quien más y mejor conozca el software en cuestión. O, cuando menos, los usuarios pueden mostrarse más confiados en confiar el mantenimiento del software a quien lo ha creado.

### Modalidades del servicio

Existen tres **modalidades típicas del servicio de mantenimiento**. En muchos contratos, se contemplan varias o todas estas modalidades:

- 1) **Mantenimiento correctivo**: asistencia técnica para corregir los errores o fallos que se produzcan en el funcionamiento del software.

- 2) **Mantenimiento preventivo:** asistencia técnica, por ejemplo mediante revisiones periódicas, para evitar que se produzcan errores en el funcionamiento.
- 3) **Mantenimiento evolutivo o de actualización:** proporcionar al usuario las mejoras o nuevas versiones que el proveedor lance al mercado sucesivamente.

### Elementos principales y obligaciones de las partes

Los elementos principales en un contrato de mantenimiento son:

- a) **Características del servicio:** asistencia y corrección de errores a distancia, reparación de los errores *in situ* por un técnico (mantenimiento correctivo); revisiones periódicas (mantenimiento preventivo); provisión de mejoras y nuevas versiones (mantenimiento evolutivo).
- b) **Tiempos de respuesta:** tiempo máximo, a partir del requerimiento del cliente, en el que el proveedor ha de atender y resolver la incidencia.
- c) **Duración:** plazo de tiempo durante el cual debe prestarse el servicio.
- d) En cuanto a las **obligaciones de las partes:**
  - **Proveedor:** debe mantener el software en funcionamiento, corrigiendo los errores que se produzcan; proporcionar al usuario, en su caso, las mejoras o nuevas versiones del software.
  - **Usuario:** principalmente, abonar la cuota de mantenimiento. Frecuentemente, la cuota de mantenimiento se fija sobre un porcentaje (%) del precio de la licencia.

#### 4.6.2. Contrato de consultoría y formación



El proveedor del software presta al usuario el servicio consistente en atender sus consultas sobre la instalación y funcionamiento del software. Se diferencia del contrato de mantenimiento en que en este caso no se trata de evitar o corregir errores del software, sino de resolver dudas al usuario sobre su funcionamiento.

Al consistir en un servicio que el usuario necesita sobre todo al empezar a usar el software, cuando todavía no lo conoce bien, es posible que este servicio quede incluido dentro de la licencia, como prestación accesoria a la instalación del programa.

Suele prestarse mediante un servicio de consulta telefónica o por correo electrónico; las consultas del usuario se responden al momento o dentro de un tiempo máximo. También puede acudir un técnico a las dependencias donde el usuario tiene instalado el software.

Dentro del contrato de consultoría, tenemos como variante la **formación**: técnicos del proveedor enseñan a los empleados del usuario el funcionamiento del software, o les imparten unos cursillos de uso.

De forma análoga a lo señalado sobre el contrato de mantenimiento, en el caso del software libre, cualquier empresa de servicios informáticos, al disponer del código fuente del programa, puede prestar en régimen de libre competencia el servicio de consultoría y formación.

#### 4.6.3. Escrow

En este caso no se trata de una prestación de servicios, sino de una modalidad de depósito. El proveedor del software entrega una copia del código fuente a un tercero (depositario), normalmente un notario, una compañía de seguridad o una empresa tecnológica, quien se compromete a custodiar (y no revelar) la copia del código a cambio de una remuneración.



La especialidad de este depósito es que el depositario puede entregar la copia del código fuente del software al usuario-licenciatario, siempre que concurran determinadas circunstancias.

Dichas circunstancias son, normalmente:

- Quiebra, suspensión de pagos o disolución del proveedor.
- Incumplimiento por el proveedor de sus obligaciones contractuales como licenciante del software o prestador del servicio de mantenimiento.

**Nota**

Como podréis adivinar, este contrato sólo se da en el caso de software propietario y nunca si el software objeto de licencia es un software libre o de código abierto, en el que el usuario puede disponer libremente del código fuente.

En el caso del software propietario, el pacto de escrow se ha visto como la vía que permite compatibilizar el deseo del proveedor a conservar en exclusiva el código fuente, con el derecho del usuario a poder seguir usando el software si el proveedor –estando obligado– no le resuelve los fallos de funcionamiento o no le proporciona las mejoras.

#### 4.7. Conclusiones

En esta unidad hemos obtenido una aproximación general a las licencias de uso, tanto de software propietario como libre, de la que podemos extraer los conceptos siguientes:

- a) Las licencias de uso son un contrato, suscrito entre dos partes, que constituye el instrumento legal habitual por el que el titular de un software concede a otras personas –los usuarios– el derecho a utilizarlo.

En las licencias se estipulan, además, limitaciones y prohibiciones que el usuario debe respetar (sobre todo, en caso de software propietario); o, en su caso, la concesión al usuario de otros derechos como la modificación y la redistribución (sobre todo, en caso de software libre).

- b) El derecho que se concede al usuario sobre el software con la licencia puede no coincidir con los derechos que el usuario adquiere sobre el soporte del software. Asimismo, el soporte en el que se proporciona la copia de software al usuario condiciona la forma de la licencia (documento en papel, archivo electrónico, etc.).
- c) Las licencias de uso se emplean como instrumento legal para la distribución de software “estándar”, lo que supone que gran parte

de sus cláusulas consistan en condiciones generales que el licenciante impone a todos los usuarios.

- d) Las licencias de uso se revelan como un instrumento legal adecuado tanto para la distribución de software propietario como de software libre, por cuanto cumplen con la función jurídica y económica que se persigue en ambos casos: proteger los derechos del licenciante y favorecer sus intereses (licencias “propietarias”) o, en el otro extremo, asegurar las libertades de los usuarios e impedir que alguien se apropie en exclusiva del software (licencias “libres”).
- e) Las licencias de uso se califican como un contrato *sui generis* y atípico, de modo que no existe una ley que lo regule por completo, ni se le aplican de forma automática y en todos los casos las leyes que sí que regulan contratos tradicionales (compraventa, arrendamiento, préstamo, donación, prestación de servicios, etc.). No obstante, las circunstancias concretas de cada supuesto determinarán la posible aplicación, por analogía, de las leyes que regulan unos u otros de dichos contratos tradicionales.
- f) Las licencias de uso se hallan sujetas a lo que dispone no sólo la Ley de Propiedad Intelectual, sino también a ciertas normas en sede de derecho contractual que tratan de proteger la posición de la parte más débil en ciertos contratos: normas protectoras de los consumidores, de los que se adhieren a condiciones generales o de los que contratan por Internet (p. ej., en licencias *click-wrap*).
- g) Existen algunas modalidades especiales de licencias de software, como la “llave en mano”, y las de uso especial para *upgrade* o *outsourcing*, que presentan unas circunstancias específicas de particular relevancia.
- h) Junto a muchas licencias de uso de software, aparecen contratos accesorios, tales como los de mantenimiento, consultoría o escrow. Especial atención merece el de mantenimiento, que es en ocasiones de gran importancia para el interés del licenciante. El debate entre software propietario y software libre se refleja en este punto, por cuanto el licenciante de software propietario controla la prestación del servicio de mantenimiento sobre el mismo (tiene “cautivo” al usuario), mientras que el software libre permite que el servicio de mantenimiento se pueda prestar en libre competencia.





## 5. Elementos de la licencia de software

Al analizar las licencias de uso, tanto las de software propietario como las de software libre, merece especial consideración que estudiemos sus **elementos subjetivos** o personales: qué sujetos son las partes de la licencia de software. Por un lado, se encuentra el proveedor-licenciante, quien concede un derecho de uso sobre el software al usuario-licenciatario (en una licencia de software libre, concede además el derecho a modificar y redistribuir el software, con o sin modificaciones). Por otro lado, tenemos a ese usuario-licenciatario, quien a su vez adquiere tal derecho de uso, abonando o no un precio por ello.

El proveedor-licenciante ha de encontrarse facultado para conceder licencias de software, bien por ser su autor, el titular de sus derechos de explotación o, como mínimo, de un derecho a su distribución. Por su parte, con relación al usuario-licenciatario, es importante saber si se trata de un empresario o de un consumidor, pues de ello dependerá el régimen legal aplicable a la licencia.

Buena parte de esta unidad la ocupará, por tanto, el análisis de las figuras del proveedor-licenciante y del usuario-licenciatario. Ello, junto a la llamada *garantía de titularidad* que el proveedor-licenciante ha de prestar en la licencia de uso: asegurar al usuario y responsabilizarse frente a éste de que obtiene la licencia para usar el software de alguien facultado legalmente para concederla, sin infringir derechos de autor de terceros. También veremos la importancia específica que, en las licencias de software libre, tiene el respeto al reconocimiento y la reputación del autor original del software, lo que contribuye a garantizar las libertades de uso, modificación y redistribución.

Además, abordaremos un punto importante en las licencias de uso de software, como es la formación de las mismas y el inicio de su vigencia, en tanto que consisten en un contrato producto del acuerdo de voluntades entre dos partes (proveedor y usuario). En concreto, analizaremos la aceptación de las cláusulas de la licencia por parte

del usuario. Nos centraremos en los problemas de validez legal de las licencias *shrink-wrap* y *click-wrap*, las cuales consisten en considerar como aceptación de las cláusulas de la licencia una determinada acción del usuario (abrir el envoltorio del paquete del software, instalarlo desde Internet); así como también los problemas de validez legal que comportan las cláusulas sobre “aceptación” en la licencia GNU-GPL, licencia de software libre por antonomasia.

Por último, examinaremos dos de los principales elementos objetivos de toda licencia de software: el plazo y el precio. En particular, haremos hincapié en las distintas modalidades de plazos y de precio, así como el plazo y el precio en las licencias de software libre.

Con esta unidad de aprendizaje, el lector alcanzará los objetivos siguientes:

1. Conocer los elementos subjetivos o personales del contrato, es decir, las dos partes intervinientes en una licencia de software: el proveedor-licenciante del software y el usuario-licenciatario del software.
2. Aprender a determinar quiénes son las diversas personas que pueden licenciar un software (proveedor-licenciante): su autor/es original/es, el titular de los derechos de explotación, el distribuidor de software.
3. Saber, en particular, quién puede licenciar un software en el caso de que haya sido desarrollado por varios programadores (distinguiendo la figura de la “obra conjunta” de la “obra en colaboración”), o bien en el caso de que haya sido desarrollado por colaboradores o asalariados a instancias de una persona jurídica (es decir, una empresa, Administración pública o una organización sin ánimo de lucro).
4. Entender en qué consiste la denominada *garantía de título o titularidad*.
5. Saber, respecto al usuario-licenciatario, que sus derechos y obligaciones pueden ser distintos en función de si se trata de un consumidor o de un empresario/profesional.
6. Conocer las especialidades que las licencias de software libre suponen en la posición del proveedor-licenciante.

7. Comprender la importancia de la aceptación del usuario como forma y momento en el que la licencia pasa a ser un contrato vinculante entre el proveedor-licenciante y el usuario-licenciatario. En este sentido, analizaremos las licencias *click-wrap* y *shrink-wrap*, así como sus posibles problemas de legalidad.
8. Discernir, como caso particular, cuándo se entiende por producida la aceptación de la licencia GNU/GPL.
9. Saber analizar dos elementos objetivos esenciales en las licencias de software: el plazo de duración y el precio de la licencia. En especial, estudiaremos la duración y el precio en las licencias de software libre.

### 5.1. Elementos subjetivos o personales. Partes en las licencias de software

Al igual que en otros contratos, en toda licencia de uso de software existen dos partes, dos personas (físicas o jurídicas) a las cuales la licencia les asigna derechos y obligaciones con relación al software objeto de la licencia. Dichas partes son el proveedor-licenciante del software y el usuario-licenciatario del software.

#### 5.1.1. El proveedor-licenciante del software



El **proveedor-licenciante** es quien concede la licencia al usuario para utilizar el software, proporcionándole una copia del software licenciado.

Como hemos visto, el proveedor-licenciante del software es quien suele fijar de manera unilateral los términos y condiciones de la licencia, limitándose el usuario-licenciatario a aceptarlos o rechazarlos, pero sin poder entrar a negociar el contenido de los derechos y obligaciones de la licencia.

El proveedor-licenciante debe tener derechos suficientes sobre el software, conforme a la legislación sobre derechos de autor (o en su

caso, titular de un software patentado), para poder conceder licencias de uso. Entenderemos que el proveedor-licenciante se encuentra facultado para conceder licencias de un software cuando es:

- a) El **autor (o autores) del software**: es decir, el programador o programadores que lo han creado. Como titular original de los derechos exclusivos sobre el software, el autor será quien pueda distribuirlo mediante licencias de uso, salvo que los derechos de explotación sobre el software se hallen cedidos a otra persona, física o jurídica.
- b) El **titular de los derechos de explotación del software**: en ocasiones, el autor ha cedido (voluntariamente o por obligación legal) los derechos de explotación a otra persona, física o jurídica. Será entonces esta persona, como titular de los derechos de explotación, quien pueda usar, modificar y distribuir el software a terceros.

#### Ejemplo

Como **cesión legal** de los derechos de explotación sobre el software, tenemos como ejemplo por antonomasia el de los programadores que son empleados de una empresa desarrolladora de software: si su contrato de trabajo no estipula lo contrario, la Ley de la Propiedad Intelectual asigna automáticamente a la empresa los derechos de explotación del software que crean sus empleados, en el marco de su relación laboral.

Como **cesión voluntaria** tenemos como ejemplo aquellos programadores que, siendo independientes y autónomos, crean un software por encargo de una empresa, que se encargará de comercializarlo. El contrato que rijan las relaciones entre ambas partes (contrato de desarrollo de software) podrá estipular la cesión de los derechos de explotación sobre el software a la empresa.

- c) **Titular sólo del derecho a distribuir el software (distribuidor)**: persona designada por el autor o el titular de los derechos de explotación sobre el software, como distribuidor del software (por ejemplo, en un territorio). En tal caso, al distribuidor del software se le faculta para conceder sublicencias.



Pueden conceder licencias de uso (a) el autor o autores originales del software, (b) la persona física o jurídica que sea titular de los derechos de explotación, o (c) aquella que como mínimo tenga el derecho a distribuir el software objeto de la licencia en cuestión. Esta diversidad de sujetos con capacidad para conceder licencias de software es lo que explica que denominemos a esta parte **proveedor del software** o simplemente **licencian-te**. Se trata de expresiones más genéricas que permiten abarcar a todos los que pueden otorgar licencias sobre el software, a diferencia de otras comúnmente empleadas como *autor*, *titular* o *propietario* del software.

En los próximos apartados analizaremos con detalle cada uno de estos posibles “proveedores-licenciantes” de software.

## El autor o autores



**Proveedor-licenciante** puede ser, en primer lugar, el autor o autores del software, es decir, el programador o programadores que lo han creado, como titular/es original/es de los derechos exclusivos sobre el software. El autor podrá conceder licencias de uso siempre que los derechos de explotación sobre el software no se hayan cedido a un tercero.

Entendemos que, en principio, autor de un software sólo puede ser una o varias personas físicas porque se trata de una obra intelectual, creada por el intelecto humano. No obstante, la propia Ley de la Propiedad Intelectual prevé que, en determinadas circunstancias, una persona jurídica pueda considerarse “el autor” del programa informático, como en los casos en que un software sea una “obra colectiva”, tal y como la hemos definido en este curso.

Además, el autor persona física sólo podrá conceder licencias de uso de su software si es mayor de edad, es decir, si tiene capacidad legal

### Nota

Podéis ver la definición del software como obra colectiva en el apartado 2.1 de este curso.

para contratar. Los autores de software que sean menores de edad, lo mismo que para suscribir otros contratos, necesitarán la autorización de sus padres o tutor para conceder una licencia válida.

No obstante, la Ley de la Propiedad Intelectual permite a los autores menores de edad conceder por sí mismos licencias de software si se trata de “autores menores de dieciocho años y mayores de dieciséis, que vivan de forma independiente con consentimiento de sus padres o tutores o con autorización de la persona o institución que los tengan a su cargo”.

El software puede ser creado por el esfuerzo y la labor de un programador aislado. Será éste el único autor y, por tanto, el único facultado para explotar el software. Pero sabemos que, en la práctica, éste no es el supuesto habitual. El software suele ser el producto del esfuerzo de varias personas. En estos casos, el software puede consistir en una:

- a) **Obra en colaboración:** se trata del software creado como resultado unitario del trabajo de varios programadores, siempre que sea posible separar las aportaciones de cada uno y explotarlas independientemente.

#### Nota

Podéis ver en la unidad 2 la legislación sobre propiedad intelectual aplicada a obras en colaboración.



Como vimos, cuando el software consiste en una “obra en colaboración”, todos los programadores intervinientes tienen la consideración de autores (coautores), y son titulares de los derechos de autor sobre el software en la proporción que ellos pacten (si no hay pacto, son titulares a partes iguales). El software es una propiedad común de todos ellos, así que para conceder una licencia del mismo, será necesario en principio el consentimiento de cada uno de los coautores.

Esta forma de autoría es importante tenerla en cuenta en el caso del software libre, pues el software que sea objeto de una licencia libre puede irse desarrollando y mejorando con las contribuciones de diversos programadores. Cuando esas contribuciones constituyan módulos separables, susceptibles de explotarlas independientemente, a los efectos de determinar la autoría del software deberemos entender que se trata de una obra en colaboración: propiedad en común, en principio a partes iguales, de todos los programadores.

- b) **Obra colectiva:** es más frecuente que, cuando intervienen varios programadores en la creación de un programa informático, el resultado sea una “obra colectiva”. En este caso, la aportación de cada programador al software no puede ser objeto de explotación separada, sino que cada una de las aportaciones “se funde en una creación única y autónoma”.



Tal y como vimos, la Ley de la Propiedad Intelectual define como obra colectiva:

“la creada por la iniciativa y bajo la coordinación de una persona natural o jurídica que la edita y divulga bajo su nombre y está constituida por la reunión de aportaciones de diferentes autores cuya contribución personal se funde en una creación única y autónoma, para la cual haya sido concebida sin que sea posible atribuir separadamente a cualquiera de ellos un derecho sobre el conjunto de la obra realizada.”

#### Nota

Podéis ver en la unidad 2 la legislación sobre propiedad intelectual aplicada a obras en colaboración.

En este caso, según la Ley de la Propiedad Intelectual, el autor es, salvo pacto en contrario, la persona natural o jurídica que la edita y divulga bajo su nombre. Por tanto, sólo es esta persona (y no los creadores) quien puede conceder licencias.

#### Ejemplo

Esto significa que, por ejemplo, cuando una empresa que se dedica a desarrollar y comercializar software encarga a unos cuantos programadores (sean *freelance* o empleados suyos) que creen un software, para comercializarlo en el mercado, es la empresa y no los creadores la única autora a efectos legales. Por tanto, la empresa sería la única titular, tanto de los derechos morales como de los derechos de explotación (si bien, esta interpretación literal de la Ley es discutida por algunos juristas, que entienden que en ningún caso puede ser una persona jurídica titular de derechos morales).

La figura de la “obra colectiva” permite también a instituciones como la Free Software Foundation que se le considere legalmente como autora del software que se crea bajo sus auspicios.

Esta distinción entre *obra colectiva* y *obra en colaboración* es importante, a los efectos de desarrollo del software libre. Hemos dicho que, si a partir de un software libre original, un programador desarrolla un módulo que constituye un elemento separable, el conjunto será una obra en colaboración. Sin embargo, si las obras derivadas constituyen contribuciones a un módulo único, que no pueden ser objeto de explotación individual, estaremos ante una obra colectiva: en este supuesto, el autor será aquella persona física o jurídica (empresa, administración pública, organización sin ánimo de lucro, etc.) que dirija y coordine el desarrollo de ese software.

### Titular de los derechos de explotación o económicos

Cuando se crea un software, el **titular original** de los derechos de explotación es su autor. No obstante, puede que con posterioridad a su creación otra persona (física o jurídica) devenga titular de los derechos de explotación: hablamos entonces de *titular derivado*. En tal supuesto, esa otra persona pasa a ser la única facultada para autorizar las reproducciones del software, hacer modificaciones o distribuirlo.



Por tanto, cuando el autor cede (voluntariamente, por sucesión o por obligación legal) los derechos de explotación sobre el software a un tercero, éste pasa a ser el único capaz de conceder licencias de uso sobre ese software.

Dicho titular puede haber adquirido los derechos de explotación, por parte del autor original del software, por:

- a) **Cesión voluntaria:** por un contrato en el que el autor original le haya cedido todos sus derechos de explotación a otra persona, generalmente a cambio de una remuneración económica. En este caso, y a diferencia de los derechos que se conceden al usuario en una licencia, en una cesión el autor original transmite por completo sus derechos de explotación a un tercero, de forma irrevocable, por lo que el autor pierde tales derechos y pasa a ser titular sólo de derechos morales (en los países anglosajones, donde no se reconocen derechos morales sobre el software, el autor que cede sus derechos de explotación se queda, evidentemente, sin derecho alguno).



Cuando una empresa que se dedica a desarrollar y comercializar software encarga la creación de un programa informático a un programador o programadores que no son empleados suyos, sino independientes y autónomos (*free-lance*), el contrato que rijan las relaciones entre las partes (contrato de prestación de servicios o de obra) deberá contemplar expresamente que el programador cede los derechos de explotación sobre el software a la empresa.

#### Ejemplo

Son habituales los casos de cesión voluntaria de los derechos de explotación sobre el software: por ejemplo, cuando una empresa dedicada a comercializar software encarga el desarrollo de un programa a programadores *free-lance*. Pero han de pactarlo expresamente en el contrato que vincule a la empresa con los programadores.

Aunque en algún país, como el Reino Unido, puede entenderse que en todo caso será titular de esos derechos la empresa (supuestos de *work-for-hire*), en España sería necesario estipular expresamente la cesión en el contrato para que se entienda que la empresa adquiere realmente los derechos de explotación, en particular distribuir el software por licencias de uso.

#### Ejemplo

Pensemos en aquel software creado por encargo de un cliente: estamos ante un contrato de desarrollo de software en el que se pacta que los derechos sobre el mismo quedarán en poder del cliente. Si el software no se ha desarrollado sólo para el mero uso por el cliente, sino para que éste lo comercialice posteriormente, es el cliente quien concederá licencias de uso sobre el software que mandó crear.

- b) **Sucesión:** en caso de fallecimiento del autor (persona física), todos sus derechos sobre el software pasan a manos de la persona (física o jurídica) a quien el autor haya designado expresamente en su testamento. Si el autor fallecido no designó a la persona a quien cedía los derechos sobre el software de su titularidad, los

derechos pasan a sus herederos legales. Serán estas personas quienes, en su caso, pueden seguir concediendo licencias sobre el software creado por un autor ya fallecido.

- c) **Disposición de la ley. Los asalariados:** tal y como dispone la Ley de la Propiedad Intelectual, cuando un programador crea un software, en su condición de empleado de un empresario, los derechos de explotación corresponden al empresario, salvo que hayan pactado lo contrario. El empleado se queda únicamente con los derechos morales, pero es la empresa la que puede explotar el software, en su caso mediante la concesión de licencias de software.



En concreto, el art. 97.4 de la Ley de la Propiedad Intelectual dispone que:

“[cuando un] trabajador asalariado cree un programa de ordenador, en el ejercicio de las funciones que le han sido confiadas o siguiendo las instrucciones de su empresario, la titularidad de los derechos de explotación correspondientes al programa de ordenador así creado, tanto el programa fuente como el programa objeto, corresponderán exclusivamente al empresario, salvo pacto en contrario.”

#### Nota

Tal y como hemos visto, la Ley de la Propiedad Intelectual asegura a las empresas que se dedican a desarrollar y comercializar software, que van a ser las únicas con derecho a explotar económicamente el software que crean sus programadores empleados:

- Cuando se trata de una **obra colectiva** (lo que sucede en muchos de los casos), la empresa es la única autora a efectos legales, titular, tanto de los derechos morales como económicos.
- Si la crea **un solo programador**, o se trata de una **obra en colaboración**, los empleados se conside-

ran como los autores, pero sólo titulares de los derechos morales. Por ministerio de la Ley de la Propiedad Intelectual, se ceden automáticamente los derechos de explotación a la empresa, entre ellos la distribución del software a través de licencias de uso. El asalariado, por el contrario, se ve impedido de otorgar licencia alguna.

### Titular del derecho a distribuir el software

Éste es el supuesto de los distribuidores de software. Al igual que muchos fabricantes comercializan sus productos a través de distribuidores (encargados de revenderlos, por ejemplo, en un determinado territorio) también es habitual que el autor o el titular de los derechos de explotación sobre el software decida comercializar su software a través de una red de distribuidores. Con ello, se procura una mayor y más rentable distribución del software.

En estos supuestos, el autor o titular de los derechos de explotación sobre el software (entre ellos, la distribución) designa a varios distribuidores para que se encarguen de comercializar licencias de uso. Ambas partes se vinculan por un **contrato de distribución** en el que al distribuidor se le concede una licencia para que éste pueda, a su vez, conceder **sublicencias de uso** a los usuarios.



El **distribuidor** se encuentra facultado para conceder sublicencias de uso sobre el software mientras permanezca en vigor el contrato de distribución; cuando finalice dicho contrato, dejará de ser distribuidor y cesará, por lo tanto, su derecho a otorgar sublicencias.

#### Nota

Hablamos de distribuidores *strictu sensu*. Es decir, no todo aquel que comercializa un software en nombre de su titular, sino sólo aquel que obtiene una licencia del titular, a cambio de una remuneración, para con-

ceder, en su propio nombre, las licencias de uso (sublicencias en este caso) a los usuarios.

En este apartado no consideramos como “distribuidores”, por ejemplo, unos grandes almacenes o una tienda de informática en la que se puede adquirir software de consumo. Estos establecimientos no son, en principio, distribuidores, porque lo único que venden es el soporte (CD-ROM) en el que se halla la copia del software, pero la licencia de uso se suscribe directamente entre el “fabricante” del software y el usuario.

Aquí nos referimos a “distribuidores” como aquellos que sí que están facultados para ser “proveedor del software”, es decir, para conceder licencias (en este caso, sublicencias) de software.

El distribuidor no tiene ningún derecho más sobre el software (no puede modificarlo), salvo la posibilidad de sublicenciar. Incluso es frecuente que el autor o titular de los derechos de explotación imponga al distribuidor el modelo de sublicencia que éste ha de otorgar a sus clientes. Eso sí, frecuentemente se asigna a cada distribuidor un territorio en exclusiva (un país, una zona del país, una ciudad, etc.), en el que sólo ese distribuidor puede comercializar el software.

El distribuidor adquiere las licencias pagando por ello al titular del software, y obtiene su beneficio sublicenciando el software a un mayor precio. El titular del software no puede imponer al distribuidor el precio final de la sublicencia, aunque sí que pueden pactar una política de precios recomendados.



El titular de un software propietario está interesado en retener en exclusiva el derecho de distribución sobre el software. Como ya sabemos, según la Ley de la Propiedad Intelectual este derecho se agota con la “primera **venta** en la Unión Europea de una copia por el titular de los derechos o con su consentimiento.”

Por tanto, si la comercialización del software se produce a través de un distribuidor, el titular del software no “vende” ninguna copia al distribuidor para que éste las revenda, porque esto le haría perder su derecho exclusivo de controlar la distribución de las copias. De este modo, el titular del software se limita a conceder una licencia de uso al distribuidor, facultándole a otorgar “sublicencias de uso” a los que serán los usuarios finales.

En la sublicencia, el distribuidor advierte expresamente al usuario quién es el titular de los derechos exclusivos sobre el software, si bien es el distribuidor quien se compromete a garantizarle el uso del software durante la duración de la sublicencia. Para ello, el distribuidor tendrá a su vez una garantía de “no revocación” de la licencia por parte del titular del software: el titular del software respetará la sublicencia concedida al usuario y le permitirá el uso del mismo.



También es habitual que el distribuidor preste servicios a los usuarios, tales como consultoría técnica o mantenimiento del software. El distribuidor cobra por ello al usuario una cuota anual o mensual. También es habitual que sea el propio distribuidor quien preste la garantía propia de la licencia del programa.

### Garantía de titularidad

Hemos explicado quiénes pueden ser proveedor-licenciante en una licencia de software: el autor (creador) del software, o el titular de los derechos de explotación, o bien un distribuidor facultado para conceder sublicencias. Por tanto, sólo pueden conceder el derecho a utilizar el software aquellas personas que sean titulares de sus derechos de explotación o, como mínimo, facultadas por dichos titulares para distribuirlo.

Si la licencia la concede alguien que no tiene estos derechos, nos encontraremos con una cesión nula e ilegal del derecho de uso sobre el software.



En este caso, el licenciente habrá concedido una licencia vulnerando los derechos exclusivos de quien sea el legítimo autor o titular exclusivo de los derechos de explotación sobre el software. Éste podrá exigir y entablar las acciones legales oportunas para impedir que sigan vigentes las licencias otorgadas sin su permiso, prohibiendo el uso del software al usuario y exigiendo responsabilidades al licenciente.

La falta de titularidad de derechos sobre el software por parte del proveedor no implica, en principio, responsabilidad alguna del usuario-licenciatario de buena fe (es decir, que suscribió la licencia creyendo que el proveedor estaba legalmente facultado para otorgarla) frente al verdadero titular del software. No obstante, el usuario sí puede sufrir graves perjuicios, en particular la paralización y pérdida del derecho a utilizar el software, a consecuencia de las acciones judiciales que emprenda su verdadero titular. En su caso, además, el usuario habrá pagado por utilizar un software a quien no debía.

#### Nota

Lo comentado es algo obvio: al igual que nadie puede vender y entregar a otro una cosa que no es suya, nadie puede ceder un derecho que no tiene.



Como condición inherente y *sine qua non* para conceder una licencia de software, está la de que el proveedor-licenciente cuente con los derechos suficientes para concederla; y será responsable frente al usuario por los perjuicios que éste pueda sufrir en caso de que resulte no tener título suficiente para otorgar la licencia.

Se habla de este modo de que, al conceder una licencia de uso, el proveedor presta necesariamente al usuario una **garantía de titula-**

**ridad.** Con ello, el usuario puede estar tranquilo de que usa el software legalmente y de que lo podrá seguir usando con toda continuidad mientras dure la licencia.

Conforme al derecho español, más que una garantía, la titularidad sobre el software es una condición inherente que ha de tener el proveedor sobre el software para que la cesión del derecho de uso sea válida y no infrinja derechos de autor de terceros. Por eso la explicamos en los apartados relativos a “proveedor del software” en lugar de en los apartados que, en la unidad 6, se dedicarán a garantías y responsabilidades (sobre el buen funcionamiento del software, sobre responsabilidad por daños, etc.).

Hablamos de *garantía de título o titularidad* por influencia del derecho anglosajón, dado que muchas licencias de software son traducción o adaptación de licencias estadounidenses. En derecho anglosajón, existe la figura de las *warranties* (garantías que un vendedor, un arrendador, o un licenciante presta a la otra parte), entre las que se encuentra la *warranty of title*: en el caso de las licencias de software, significa que el proveedor asegura al usuario estar debidamente facultado para concederle el derecho de uso (o lo que es lo mismo, el usuario adquiere el derecho de uso de quien está facultado para concedérselo) sin infringir derechos de terceros.



De conformidad con el derecho español, el proveedor del software siempre debe garantizar su titularidad al usuario-licenciatario, sea consumidor o profesional, haciéndose responsable de los perjuicios que sufra el usuario para el caso de que un tercero, que resulte el auténtico titular del software, le impida seguir utilizándolo.

La garantía de titularidad resulta del propio derecho general de las obligaciones y contratos: incurre en responsabilidad la parte que incumple dolosa o negligentemente sus obligaciones contractuales. Asimismo, quien causa un daño a otro, por una conducta dolosa o negligente, debe indemnizarle (artículos 1102 y 1902 del Código civil).

#### Nota

Podéis ver los apartados que se dedicarán a garantías y responsabilidades en la unidad 6.

En el caso de que el usuario sea un consumidor, se aplica la Ley General para la Defensa de Consumidores y Usuarios (existen leyes similares en todos los estados de la Unión Europea), por la cual:

- El proveedor de software no puede limitar o excluir las garantías a prestar frente al usuario consumidor, como la garantía de titularidad.
- El proveedor de software debe responder frente al usuario consumidor del origen, identidad e idoneidad del software.

La propia Ley de la Propiedad Intelectual (artículo 96.2) dispone que los derechos de autor protegerán un programa informático “únicamente si fuese original, en el sentido de ser una creación intelectual propia de su autor.” Esto enlaza con el artículo 102 también de la Ley de la Propiedad Intelectual, según el cual se consideran infractores de los derechos de autor quienes sin autorización del titular “pongan en circulación una o más copias de un programa de ordenador conociendo o pudiendo presumir su naturaleza ilegítima”.

Por todo ello, la conclusión parece clara: el proveedor de software que concede licencias sin estar facultado para ello, comete un acto ilícito (infracción de los derechos), de modo que deberá responder frente a los perjudicados por dicho acto ilícito: tanto frente al verdadero titular de los derechos como frente al usuario a quien otorgó ilegalmente la licencia).

Por tanto, entendemos que resultan nulas aquellas cláusulas que intentan limitar o excluir esta garantía de titularidad. En este sentido, y sobre todo en el caso de software comercializado en masa a consumidores, tenemos muchas licencias de software que, como traducción o adaptación de una licencia propia del derecho anglosajón, intentan excluir la garantía de titularidad. En España, como en el resto de la Unión Europea, esta cláusula no sería válida, tanto si el usuario es profesional como consumidor.



Según comentamos, muchas licencias de software siguen el modelo anglosajón de no conceder garantía



alguna sobre el software, ni siquiera la de titularidad, manifestando que el software se entrega al usuario “tal cual” (traducción de la expresión inglesa *as is*).

Como muchos otros proveedores de software, la propia Microsoft®, en sus licencias de software, incluye una cláusula de “garantía limitada” como ésta, que excluye expresamente la garantía de titularidad:

“Inexistencia de otras garantías. Hasta la máxima extensión permitida por la ley aplicable, Microsoft y sus proveedores no responderán por cualquier otra garantía o condición, tanto implícita como explícita, incluyendo, sin limitación, las garantías de título y no infracción respecto del producto software [...]”

Pues bien, como hemos señalado, si la ley aplicable es la española (o la de cualquier otro país de la Unión Europea), esta cláusula de exclusión de la garantía de título y no infracción **no es legal**. El proveedor del software está obligado por ley a garantizar su titularidad sobre el software.

En extremo opuesto a los intentos de excluir la garantía de titularidad, también es habitual (sobre todo en las licencias de software no comercializado en masa, que se destina a usuarios empresarios o profesionales) establecer en el contrato de licencia **cláusulas de indemnidad a favor del usuario**, para proteger sus intereses en caso de que surja un litigio sobre la verdadera titularidad del software:

- a) Asumir la defensa del usuario ante cualquier reclamación o acción que un tercero, que reclame su condición de verdadero titular del software, pueda ejercer contra el usuario, sobre la base de que está infringiendo derechos de autor al utilizar un software sin su autorización.
- b) Devolver al usuario el precio que haya abonado por la licencia, y abonarle los daños y perjuicios que el usuario pueda sufrir.

**Nota**

Podéis ver la garantía de titularidad en el caso del software libre en el apartado 5.1.3.

- c) Sustituir el software por otro, o modificar el software para que cese la infracción del derecho de autor, etc.

Como veremos, en el caso del software libre, la garantía de titularidad se sigue prestando, aunque las posiciones de proveedor del software y usuario sufran ciertas alteraciones respecto las licencias de software propietario.

### **5.1.2. El usuario-licenciatario. Especial referencia al usuario consumidor**

Se trata de la otra parte del contrato de licencia de software. Es la persona (física o jurídica) que adquiere el derecho a usar el software por medio de la licencia, según los términos y condiciones que se establecen en la misma (casi siempre impuestos por el proveedor del software).



El **usuario-licenciatario** tiene como principales obligaciones pagar el precio de la licencia (cuando es de pago) y respetar las limitaciones de uso que le impone la licencia de software, un software cuya propiedad no le pertenece.

En el caso de que el usuario sea licenciatario de un software propietario, en principio serán pocos sus derechos como usuario (básicamente ejecutar el programa, aprovechar sus aplicaciones y poder hacer una copia de seguridad del mismo), mientras que las limitaciones son muchas. Por el contrario, si es licenciatario de un software libre, las libertades del usuario-licenciatario son mucho más amplias, y por ende, las limitaciones son menores: puede usar el software libremente, modificarlo y redistribuirlo con o sin modificaciones.

Si el usuario está facultado para modificar y modifica el software, puede pasar a ser el **autor de una obra derivada**, según el artículo 11 de la Ley de la Propiedad Intelectual (es decir, de la traducción o adaptación del software). Por su parte, si el usuario está facultado para redistribuir y redistribuye el software, se convertirá también en **proveedor de software**.

Para ser usuario-licenciatarario no se requiere ningún requisito especial en principio, aparte de las exigencias sobre **capacidad legal** genéricas: que el usuario persona física sea mayor de edad o, si se trata de una persona jurídica (empresa, administración, asociación sin ánimo de lucro, etc.), que ésta se halle válidamente constituida).

Es importante tener en cuenta si se emplean o no condiciones generales en las licencias de software (en casi todos los casos se emplean) y si el usuario-licenciatarario es un consumidor o un empresario, porque varía el régimen legal al que está sujeto el contrato de licencia.



A veces, en el propio texto de la licencia de uso se contemplan derechos y limitaciones distintas según si el usuario es un consumidor que va a destinar el software a un uso particular o un empresario/profesional que va a destinar el software a su actividad. Se trata de las llamadas **licencias duales**.

Cuando nos referimos a que el régimen legal varía, no hablamos de los derechos que el usuario puede tener sobre el software, porque el proveedor puede conceder o no conceder en su licencia los mismos derechos sobre el software a todos los licenciatarios, sin distinguir entre usuarios consumidores o empresarios. Sin embargo, las leyes y otras normas jurídicas que se aplican o pueden ser de aplicación a la licencia, sobre la validez, aplicación e interpretación de sus cláusulas (por ejemplo, sobre terminación del contrato, responsabilidades del proveedor, etc.) dependen de si el usuario es consumidor o profesional.

### El usuario consumidor

Si el usuario es un consumidor, se entiende que se halla en una posición especialmente débil, por lo que debe tener una protección legal frente a posibles abusos del proveedor del software, al igual que sucede en muchos otros contratos que celebran los consumidores.

En tal caso, como vimos, a la licencia le son de aplicación las normas de la Ley sobre Condiciones Generales de la Contratación y de la Ley

#### Nota

Podéis ver al respecto el apartado 4.1.4.

**Nota**

Podéis ver los requisitos generales de validez de las condiciones generales en el apartado 4.1.4.

### General para la Defensa de Consumidores y Usuarios que **prohíben las cláusulas abusivas**.

Tanto si el usuario-licenciatario es un consumidor como si es un empresario o profesional, en tanto que la licencia de uso del software contenga condiciones generales, le serán de aplicación los requisitos generales de validez que ya vimos (el usuario debe poder conocer las condiciones generales previa o simultáneamente a la suscripción del contrato, deben redactarse con concreción, claridad y sencillez; la ambigüedad en una condición general no puede beneficiar a quien la impone, etc.).

Sin embargo, sólo cuando el usuario sea un consumidor, el texto de la licencia de uso no podrá incluir nunca condiciones abusivas (si el usuario es o no un empresario o profesional, dependerá de las circunstancias de cada caso, como analizaremos seguidamente).

Normalmente, las cláusulas abusivas formarán parte de las condiciones generales de un contrato (por ejemplo, el texto de una licencia de uso de un software estándar comercializado en masa, que ha de regir para todos los usuarios). Pero también quedan prohibidas las cláusulas abusivas que sean condiciones particulares (esto es, que rijan para un solo usuario) siempre que las haya impuesto el licenciante, sin haberse negociado realmente entre las partes.

Sabemos que la Ley General para la Defensa de los Consumidores y Usuarios considera abusivas aquellas estipulaciones no negociadas individualmente que, en contra de las exigencias de la buena fe, causen, en perjuicio del consumidor, un desequilibrio importante de los derechos y obligaciones de las partes. A efectos prácticos, la Ley General para la Defensa de los Consumidores y Usuarios enumera unos tipos de cláusulas que en todo caso se consideran cláusulas abusivas.



Por tanto, por lo que respecta a las licencias de software, el proveedor del software no podrá imponer al usuario consumidor cláusulas abusivas. Si las incluye en la licencia, serán **cláusulas nulas**, de modo que no podrá exigir su cumplimiento al usuario.

Como ejemplos de cláusulas que está prohibido incluir, por abusivas, en una licencia de software que se conceda a un usuario consumidor tenemos:

a) Cláusulas que condicionan el cumplimiento de la licencia a la exclusiva voluntad del proveedor:

- Derecho del proveedor a modificar o interpretar unilateralmente los términos de la licencia, después de haber sido aceptada por el usuario.
- Derecho del proveedor a resolver libremente el contrato, sin dar preaviso ni indemnización (cláusula abusiva salvo que se reconozca el mismo derecho al usuario).
- Exigencia de una indemnización exageradamente alta al usuario consumidor, en caso de que éste incumpla alguna obligación.

b) Cláusulas que priven al consumidor de sus derechos básicos:

- Limitar o excluir las garantías que legalmente deba prestar sobre el software, así como limitar o excluir su responsabilidad por daños causados por un software defectuoso.
- Forzar al usuario a renunciar a una copia escrita de la licencia o negarse a expedir recibo por el pago del precio de la licencia.

c) Otras cláusulas abusivas:

- Obligar al consumidor a aceptar cláusulas o condiciones desconocidas.
- Obligar al usuario consumidor a adquirir bienes o servicios accesorios no solicitados.
- Imponer que, en caso de litigio sobre la licencia, tribunales distintos a los del domicilio del usuario consumidor sean competentes para resolverlo o que la licencia se someta a un derecho extranjero ajeno a las partes (por ejemplo, que un proveedor de software estadounidense imponga a los usuarios consumidores españoles que la licencia se someterá al derecho y tribunales de Singapur).

Además, cuando el usuario-consumidor adquiere la licencia de uso del software por Internet (en línea), el proveedor del software también debe cumplir con las **obligaciones de información** que le impone la Ley 34/2002, de 11 de julio, de servicios de la sociedad de la información y de comercio electrónico (LSSI):

- **Antes de adquirir la licencia**, debe proporcionarle ciertos datos sobre el proveedor y el proceso de contratación, además del texto de las condiciones generales.
- **Después de adquirir la licencia**, el proveedor debe confirmar al usuario que ha recibido la aceptación de éste.

La LSSI establece que, sea el usuario consumidor o profesional, y antes de que tenga la oportunidad de “aceptar” la licencia, el proveedor debe poner a su disposición las condiciones generales de la licencia, de manera que puedan ser almacenadas y reproducidas por el usuario. Además, cuando el usuario es un consumidor, el proveedor del software debe cumplir con obligaciones de información. Entre éstas, tenemos:

Antes de que el usuario-consumidor tenga la oportunidad de “aceptar” la licencia, el proveedor debe haberle informado de manera clara, comprensible e inequívoca de:

- Los distintos trámites a seguir para celebrar el contrato de licencia de software.
- Si el proveedor va archivar el documento electrónico en que se formalice la licencia y si éste va a ser accesible.
- Los medios técnicos que pone a disposición del usuario para identificar y corregir errores en la introducción de los datos.
- La lengua o lenguas en que podrá formalizarse el contrato.

Después de haber “aceptado” el usuario, se entiende celebrado el contrato de licencia de software, por lo que entra en vigor. Una vez

“aceptada” la licencia por el usuario, el proveedor debe confirmarle que ha recibido su aceptación, mediante:

- El envío de un acuse de recibo por correo electrónico u otro medio de comunicación electrónica equivalente, en las 24 horas siguientes, o
- La confirmación de la aceptación, por un medio equivalente al utilizado para acordar la concesión de la licencia, tan pronto como el usuario haya completado el proceso de aceptación, siempre que el usuario pueda archivar la confirmación del proveedor.

Estas obligaciones de información tienen particular relevancia en el software libre, en tanto que Internet es un canal de distribución importantísimo.

No obstante, las obligaciones de la LSSI se imponen a los licenciantes establecidos en España, mientras que si está establecido en otro Estado de la Unión Europea se discute si se aplica la LSSI o la ley equivalente en ese otro Estado, que tendrá un contenido muy similar. Si el licenciante se halla establecido fuera de la Unión Europea, sólo se aplicará la LSSI a aquellas licencias que conceda a usuarios españoles, y siempre que su página web se dirija específicamente o tenga un apartado específico para España.



Recordemos que la LSSI transpone en España la Directiva 2000/31/CE. Por tanto, sus reglas se recogen tanto en la LSSI (para España), como en las correspondientes leyes que transponen la Directiva en cada país de la Unión Europea. De este modo, los licenciantes de software a través de Internet, establecidos fuera de la Unión Europea, estarán sometidos a la LSSI y/o a las leyes equivalentes en Francia, Reino Unido, Italia, etc., sólo si con su página web se dirigen de manera específica a estos países (también, si tiene una sucursal en estos países, o inserta publicidad en medios de comunicación de estos países).

Pensemos, por ejemplo, en un proveedor de software estadounidense, que distribuye sus programas desde su web, también alojada en EE.UU., pero con una versión en castellano, francés, etc.; que señale el precio en euros, o demás circunstancias que indiquen una comercialización del software específica para un país de la Unión Europea.

### El usuario empresario o profesional

Hemos de tener en cuenta que, aunque las normas protectoras del consumidor no se apliquen cuando el usuario sea un empresario o profesional, ello no significa que el proveedor del software puede imponer sin más a este usuario cláusulas especialmente injustas o abusivas. Ocurre, no obstante, que el usuario empresario no tiene la protección legal que supone que ciertas cláusulas (las que figuran en el listado de cláusulas abusivas) se consideran automáticamente nulas por disponerlo así la ley.

Si el usuario (empresario o profesional) cree que una cláusula de la licencia es abusiva y se niega a respetarla, deberá demandar el contrato de licencia ante los tribunales. La cláusula podrá ser anulada por estimar que es contraria a la **regla general de buena fe** que ha de regir en el cumplimiento de los contratos. No obstante, ello dependerá del examen de las circunstancias de cada caso en concreto y será el juez quien decida si se trata o no de una cláusula contraria a la buena fe entre las partes.

Por tanto, en aquellas licencias de uso de software destinado a profesionales, serán de aplicación los requisitos genéricos en materia de condiciones generales, pero la legalidad o ilegalidad de una cláusula “abusiva” dependerá de las circunstancias concretas de cada caso.

#### Nota

En el apartado 4.1.4 podéis ver los requisitos genéricos en materia de condiciones generales: sobre información al adherente, incorporación de las condiciones generales al contrato, redacción clara y sencilla de las mismas, sin ambigüedades que puedan perjudicar al adherente, etc.



Determinadas circunstancias se tendrán en cuenta como relevantes a la hora de determinar si una cláusula debe anularse por abusiva o no, cuando el licenciatarlo es un empresario o profesional. Por ejemplo, si el licenciatarlo es una gran empresa o pequeña; si ha existido o no un verdadero proceso de negociación entre las partes; si el usuario-licenciatarlo ha aceptado una cláusula desfavorable para sus intereses a cambio de otra previsión favorable (por ejemplo, una rebaja en el precio de la licencia o un derecho a modificar el software, a cambio de una mayor limitación de responsabilidad para el proveedor), o simplemente el proveedor se la ha impuesto.

Del análisis de estas circunstancias podrá determinarse si la cláusula debe anularse o no, por ser contraria a las reglas generales de la buena fe en los contratos.

Este criterio de la buena fe se sigue, de forma similar, en otros países. Cabe destacar el Reino Unido, en el que los tribunales acuden al llamado *test de la razonabilidad* (*reasonableness test*), para discernir (analizando las circunstancias del caso, bajo parámetros equivalentes a los señalados anteriormente) si una cláusula de la licencia especialmente favorable para el licenciante –y desfavorable para un usuario-licenciatarlo profesional– debe anularse o ratificar su validez.

#### Ejemplo

Una cláusula típica en las licencias de software es la exoneración o limitación de responsabilidad del proveedor por los daños que un software defectuoso puede causar al usuario (por ejemplo, estipular que si el usuario sufre daños por culpa del software, el proveedor no le indemnizará o sólo le devolverá el precio de la licencia).

Si el usuario es consumidor, la exoneración o limitación son automáticamente consideradas nulas. Sin embargo, cuando el usuario es un empresario, la limitación es válida, salvo que en el caso concreto resulte especialmente injusta y abusiva; por ejemplo, en el caso de limitarse la responsabilidad del proveedor a una indemnización máxima de 10.000 euros:

- Si el usuario empresario ha sufrido un perjuicio de de 15.000 euros, la limitación de responsabilidad podrá ser válida.
- Sin embargo, puede (y dependerá de la apreciación del juez) que se entienda no válida esta limitación, si el perjuicio es mucho mayor al límite máximo de indemnización, por ejemplo 5.000.000 euros, y ese perjuicio se debe a una negligencia grave del proveedor.

### **5.1.3. Posición de los sujetos intervinientes en las licencias de software libre**

En el caso de las licencias de software libre, las posiciones tradicionales de proveedor-licenciante y usuario-licenciatarario se mantienen: el licenciante es la persona que distribuye el programa, y detenta el derecho legal para distribuirlo (porque es su autor o, cuando menos, titular del derecho de distribución); y el usuario-licenciatarario es quien adquiere el derecho a usar el software, pero también a modificarlo y redistribuirlo.

### **Consideraciones generales**

Aun siendo las partes de la licencia las mismas (proveedor-licenciante y usuario-licenciatarario) tanto para software propietario, como para software libre, las diferencias tan importantes sobre los derechos que otorgan unas y otras al usuario hace que sea conveniente tener en cuenta las siguientes consideraciones:

- a) Las licencias de software libre son el medio o instrumento legal, no para que el proveedor del software pueda rentabilizar al máximo sus derechos exclusivos de explotación, sino para garantizar a los usuarios las libertades de uso, modificación y redistribución. En el supuesto de que modifique el software, el usuario será el autor de un programa derivado.

Por tanto, el usuario-licenciatarario también puede convertirse a su vez en proveedor-licenciante de otros usuarios; bien licen-

ciendo el mismo software que se le ha licenciado a él, bien por licenciar un software derivado del original. Estos otros usuarios pueden, a su vez, modificar y distribuir el programa de nuevo, y así sucesivamente.

Esto conlleva que existan riesgos de que alguien intente apropiarse de un software libre, que reclame la exclusividad en su explotación y se atreva a sostener que los usuarios utilizan ese software sin tener derecho.

- b) Debemos tener en cuenta que las legislaciones sobre propiedad intelectual, incluida la Ley de la Propiedad Intelectual en España, conceden al proveedor de software unos derechos exclusivos en virtud de los cuales ninguna otra persona puede hacer nada con el software si no cuenta con la expresa autorización (licencia) del proveedor. Por ello, el usuario de un software libre puede beneficiarse de las libertades de uso, modificación y redistribución sólo si el proveedor del software le ha concedido realmente tales libertades por medio de una licencia de uso.
- c) Además, ni en España ni en otros países es necesario inscribir el software en el Registro de la Propiedad Intelectual para que su autor sea reconocido como tal. En principio, para ello basta con que el autor pueda probar ser el creador de un software original (o derivado, con la autorización del autor del software original). Esto propicia que existan riesgos de que alguien intente apropiarse de un software libre, que reclame la exclusividad en su explotación y se atreva a sostener que los usuarios utilizan ese software sin tener derecho.
- d) Si alguien intenta apropiarse ilegítimamente del software o pretende restringir las libertades que tienen los usuarios sobre el mismo, el verdadero autor del software es quien podrá reaccionar y ejercer las medidas legales oportunas para impedir esta apropiación indebida.

A diferencia del software propietario, el autor no reaccionará tanto para proteger sus derechos exclusivos, sino más bien para que los usuarios puedan continuar disfrutando de las libertades (de uso, modificación y distribución) sobre el software.

- e) Por otra parte, quien pretenda divulgar su software como libre debe garantizar que ese software es verdaderamente libre y que no infringe los derechos de otro software (sea libre o propietario).

#### Nota

Podéis ver la garantía de titularidad en el apartado 5.4.1.

Las licencias de software libre acostumbran a establecer que el software se ofrece sin garantías, pero como hemos visto, la garantía de titularidad es inherente a la condición de proveedor del software: no se puede distribuir un software a terceros si no se puede garantizar a los usuarios que tal distribución es legal y no viola derechos de autor de terceros. En caso contrario, el verdadero titular de los derechos sobre el software podrá reaccionar y ejercer las medidas legales oportunas; que, en el caso del software libre, pasarán también por impedir que los usuarios puedan seguir empleando el software.

- f) La concesión de una licencia de software libre implica que su titular comparte con los usuarios los principales derechos de explotación sobre el mismo. Pero en España (y en el resto de la Europa “Continental”), el hecho de que ceda a una multitud de usuarios los derechos de modificar o distribuir el software no significa que el software libre pase al dominio público. El software libre no es un software sin propietario, sino que el autor conserva su condición de autor del software y, en particular, los derechos morales sobre el software.



Como según la Ley de la Propiedad Intelectual, los derechos morales sobre el software son irrenunciables, un software sólo puede pasar al dominio público (sin autor, de libre uso y explotación por todo el mundo) una vez transcurrido el plazo de vigencia de los derechos de autor: setenta años después del fallecimiento del autor (persona física) o después de su divulgación (en caso de que el titular del software sea una persona jurídica), pero nunca por decisión voluntaria del autor.

Por tanto, y dada la rápida evolución del software, en la práctica no va a existir ningún software en dominio público hasta muchos años después de haber quedado obsoleto. En cualquier caso, que el programa no pueda cederse al dominio público no significa que no se pueda distribuir gratuitamente, sin perder el autor sus derechos morales sobre el software.

Por otra parte, en los países anglosajones (EE.UU. y el Reino Unido) sí que es posible ceder software al dominio público de forma voluntaria. No obstante, ello requiere de un acto consciente y expreso por el autor en el que manifieste inequívocamente su voluntad de ceder su software al dominio público, renunciando a todos sus derechos sobre el mismo. De este modo, las licencias en software libre tampoco suponen en los países anglosajones una cesión –sin más– del programa al dominio público.

Los derechos morales son **irrenunciables** según la Ley de la Propiedad Intelectual. En EE.UU. o el Reino Unido no se reconocen derechos morales sobre el software, de modo que las licencias de software libre (basadas en derecho anglosajón) no se refieren al respeto de los derechos morales, si bien se preocupan también de garantizar el respeto al reconocimiento y a la reputación del autor. De este modo, las licencias de software libre vienen a proteger los derechos morales en aquellos países donde sí se reconocen.



Las **licencias de software libre** se basan en el reconocimiento del autor como medio para asegurar las libertades de los usuarios y asimismo impedir que alguien intente apropiarse del software libre.

En particular, las licencias de software libre cumplen con los **principales postulados de los derechos morales** sobre el software, reconocidos por la Ley de la Propiedad Intelectual:

- 1) El autor puede exigir el reconocimiento de su condición de autor del software, y que aparezca su nombre o seudónimo al distribuirlo.
- 2) El autor puede impedir cualquier modificación o alteración que menoscabe su reputación.

Veremos a continuación cómo las licencias de software libre tienen un contenido de acuerdo con el esquema propio del derecho anglosajón,

que no reconoce derechos morales sobre el software. Sin embargo, en la práctica respetan el **reconocimiento** y la **reputación** del autor.

### Autoría en la LGP-GNU y en las directrices de la OSD sobre código abierto

En el caso de la licencia de software libre por antonomasia, la **Licencia Pública General GNU** (GNU-GPL, de la Free Software Foundation), se respeta el reconocimiento del autor del software y, como decíamos, ello sirve además para garantizar a los usuarios los derechos (de uso, modificación y redistribución) que se conceden por medio de esta licencia.

En el preámbulo de la GNU-GPL se dice expresamente que los derechos del usuario a copiar, distribuir y/o modificar el software se protegen porque “ponemos el software bajo *copyright*”.

Para copiar y distribuir copias de un programa, la GNU-GPL señala que el usuario ha de publicar en cada copia “de forma adecuada y bien visible un anuncio de *copyright* adecuado y un repudio de garantía”.

#### Ejemplo

En particular, en el apéndice de la GNU-GPL se recomienda que el autor añada al principio de cada fichero fuente una línea de *copyright*, de la siguiente forma:

```
"© 200? <nombre del autor>"
```

En el caso de modificación de un software libre, el preámbulo de la GNU-GPL ya establece como regla que los usuarios sepan siempre si el software es original o modificado, no tanto porque el autor original deba prestar garantía alguna sobre el software (en principio, las garantías están excluidas, salvo que la ley del país aplicable a la licencia exija que el licenciente las preste), sino para que la reputación del autor original no quede afectada en el caso de que las versiones derivadas del software contengan algún problema de funcionamiento.

#### Nota

La identificación diferenciada del software original y de las modificaciones sirve para mantener la reputación del autor original.

De este modo, la cláusula tercera de la GNU-GPL exige al usuario que haya modificado el software y que pretenda distribuirlo las condiciones siguientes:

- Que los ficheros modificados lleven anuncios prominentes que indiquen que los ha cambiado y la fecha del cambio.
- Que si el programa modificado es interactivo, muestre el anuncio de *copyright* al iniciarse su ejecución en uso interactivo.

En cuanto a las directrices de la OSD sobre código abierto (*The Debian Free Software Guidelines*), propias de la Open Software Initiative, también parten de la base de que el software libre se debe poner a disposición de los usuarios a través de licencias en las que se identifique y reconozca al autor del software. En concreto, la directriz 3 obliga al autor a permitir que su programa sea modificado por los usuarios y que éstos puedan distribuir el software, con o sin las modificaciones. No obstante:

- La directriz 4 permite al autor que exija a los usuarios que no distribuyan el código fuente modificado directamente, sino que lo distribuyan acompañado de los archivos fuentes originales con “archivos parche” separados.
- Asimismo, el autor original puede exigir que las obras derivadas tengan un nombre distinto o un número de versión distinto del software original.



Estas restricciones a la distribución de los programas derivados están previstas, asimismo, para proteger la reputación del autor original del software ante posibles problemas en la calidad y funcionamiento del software que puedan surgir a raíz de una modificación efectuada por un usuario.

Con relación a la autoría en el software libre, mencionaremos además una cláusula que suelen contener las licencias de tipo BSD, con la finalidad de proteger la reputación del autor original: prohíben al usuario utilizar el nombre del autor original del software para pro-

mocionar el software derivado que éste desarrolle a partir del original, a no ser que cuente con el consentimiento del autor original.

## 5.2. Aceptación de las licencias de software. Licencias *shrink-wrap* y *click-wrap*

Hemos visto que las cláusulas de las licencias de software no se negocian ni pactan por las partes (proveedor y usuario) de igual a igual. Por el contrario, los términos y condiciones de la licencia son previa y exclusivamente fijados por el proveedor (condiciones generales), y todos aquellos que quieran utilizar el software deben aceptarlas. Si rechazan las condiciones que el proveedor ofrece, no podrán adquirir la licencia.



La **aceptación** del usuario es un acto o manifestación esencial en las licencias de uso. Con la aceptación de las condiciones, la persona en cuestión adquiere el derecho a usar el software y se convierte en usuario legítimo.

Los términos y condiciones de la licencia, impuestos por el proveedor y aceptados por el usuario, pasan a ser un **contrato vinculante entre las partes**, en el que proveedor y usuario tienen unos derechos y obligaciones frente al otro y con relación al software objeto de la licencia: entregar una copia del software y permitir su uso (proveedor), respetar las limitaciones de uso y, en su caso, pagar el precio (usuario).



Hablamos del **proceso de formación** típico de cualquier contrato, que consiste en:

- a) **Oferta** de un bien, derecho o servicio, a cambio de una contraprestación o gratuitamente.
- b) En su caso, proceso de **negociación**, contraofertas, etc.



c) **Aceptación** por el destinatario de la oferta.

Cuando se produce la coincidencia o **concurso de la oferta y la aceptación**, el contrato se **perfecciona**. A partir de tal momento comienza la vigencia del contrato, al pasar sus cláusulas a ser obligatorias entre las partes.

Debemos entender que, en España y en el resto de países que se rigen por un sistema de derecho “continental”, siempre es necesario que exista aceptación por parte del usuario. La licencia de uso sólo puede entenderse como un contrato consensuado entre las dos partes (proveedor-licenciante y usuario-licenciario).

Sin embargo, en el derecho anglosajón se contempla además la posibilidad de que exista una licencia unilateral. La **licencia unilateral** consiste en una declaración unilateral del licenciante, en virtud de la cual autoriza el uso del software a los que cumplan y respeten ciertas condiciones y límites, de conformidad con la ley de derechos de autor que sea aplicable.



A diferencia de las licencias que constituyen un contrato, la licencia unilateral (repetimos, propia de países en que rige el derecho anglosajón, como EE.UU. o el Reino Unido) no precisa de la aceptación expresa del licenciario. Sin embargo, la licencia unilateral sólo podrá regular exclusivamente los términos y condiciones de derecho de uso del software, sin entrar en otros aspectos que podrían regular la relación licenciante-licenciario (pactos accesorios de confidencialidad, reglas a seguir en caso de conflicto, etc.), y sin imponer al usuario-licenciario más obligaciones que las restricciones y límites al derecho de uso previstas por la legislación sobre propiedad intelectual que le sea aplicable.

El proceso y la forma de la aceptación presenta ciertas particularidades en las licencias de software, con relación a los contratos. La prác-

tica comercial de los proveedores de software ha dado lugar a las conocidas **licencias *shrink-wrap*** o **licencias *click-wrap***, las cuales suponen que ciertas acciones del usuario (rotura de envoltorio de los CD-ROM, cliqueo en la pantalla del ordenador) comportan automáticamente la aceptación de las cláusulas de la licencia. En estos casos, se cuestiona si en efecto se produce o no una aceptación real y válida.

Además, existen licencias de software libre, como la GNU-GPL, que también contemplan una aceptación “implícita” del usuario a la licencia, por el hecho de modificar o redistribuir el software, lo que también suscita el debate sobre su validez legal.

#### **5.2.1. Licencias *shrink-wrap* y *click-wrap*; discusión sobre su validez**

En la práctica ocurre que hay muchos casos en los que no está claro cuándo se produce la aceptación de las licencias de software por parte del usuario, y si lo que el proveedor considera a veces como “aceptación del usuario” es realmente una aceptación válida y legal. Ello, porque las licencias de software (salvo aquellas de software más complejo, destinado a empresas) no se suelen plasmar en un contrato “tradicional” en papel, firmado por las partes.

Por el contrario, el software comercializado en masa se distribuye, bien por Internet, bien por la comercialización de copias (en un soporte de CD-ROM) en establecimientos comerciales, pero difiriendo la aceptación de la licencia a un momento posterior a la “compra” del soporte.



En muchos otros contratos, el momento y el modo como se produce la aceptación se reconoce fácilmente. Así, en un contrato que se redacta por escrito, la aceptación viene dada al firmar ambas partes (usuario y proveedor) las hojas del contrato, siendo la firma la fórmula típica y tradicional para manifestar la aceptación del contrato. Por otra parte, cuando alguien compra un artículo de consumo en una tienda, tampoco firma contrato alguno, sino que la aceptación se deriva de unas acciones que manifiestan de manera inequívoca

su aceptación de la compra: llevarse el artículo del mostrador y pagar el precio indicado en la etiqueta, siendo el dueño de la tienda quien le vende el artículo.

Recordemos que, cuando adquirimos una copia de un programa de ordenador en una tienda, sólo compramos el soporte pero no se adquiere ningún derecho sobre el software hasta aceptar la licencia. Esto es porque el proveedor pretende controlar en exclusiva la distribución de copias, así que ni siquiera faculta a la tienda a que sea ella quien otorgue el derecho de uso.

El proveedor quiere ser él mismo quien conceda el derecho de uso sobre el software, pero es evidente que no puede reunirse con cada comprador de una copia para obtener su aceptación, ni puede mostrar al usuario las condiciones de la licencia antes de que éste haya comprado la copia.



Los proveedores de software suelen considerar que ciertas acciones del usuario suponen su aceptación de los términos de la licencia.

Dichas acciones de aceptación serían las siguientes:

- a) La rotura del precinto del paquete o caja en la que vienen los CD-ROM con el software. Se trata de la licencia denominada *shrink-wrap*, por la cual la abertura del paquete se considera aceptación de la licencia, y así lo advierte expresamente el proveedor en el exterior del paquete.
- b) Cuando un software se distribuye por Internet, se recurre habitualmente a la licencia denominada *click-wrap*: el proveedor solicita la aceptación de la licencia al usuario durante la instalación y descarga del software, generalmente al principio. Aparece una casilla con el texto "Aceptar", "Sí" o "Acepto las condiciones de la licencia", que el usuario debe clicar para seguir adelante con el proceso de instalación. Al principio o durante la instalación, se abre un archivo de texto con las condiciones de la licencia.

- c) El mero hecho de instalar o ejecutar el software por primera vez, sin necesidad de que el usuario manifieste expresamente que acepta la licencia.

En España, como en otros países, se discute acerca de la legalidad o no de estas fórmulas de aceptación. Como regla general, podemos decir que en España no es válida la aceptación de una licencia *shrink-wrap* o *click-wrap*, si no se siguen determinados requisitos legales.

En particular, la legislación española exige para la validez de todo contrato que conste la aceptación expresa de las partes. O, como mínimo, que exista un comportamiento de las partes, del que necesariamente se derive la aceptación de todas y cada una de las cláusulas de la licencia. Además, si la licencia de uso está constituida por condiciones generales, la Ley sobre Condiciones Generales de la Contratación exige que el usuario haya tenido la oportunidad de conocer las condiciones generales antes de suscribir el contrato.

Se trata de evitar que una de las partes intente imponer cláusulas que en realidad la otra no ha aceptado, ni ha tenido ni siquiera la oportunidad de conocerlas.



La simple rotura del envoltorio del CD-ROM, iniciar la instalación o hacer clic en la casilla “Aceptar”, no puede considerarse una aceptación real y válida sin que el usuario haya podido conocer antes los términos del contrato.

El proveedor del software no puede atribuir a esas acciones del usuario el carácter de aceptación de la licencia, si el usuario no ha podido conocer previa o simultáneamente las cláusulas de la licencia. En tal caso, el proveedor no podrá acreditar que el usuario aceptó todas y cada una de las obligaciones que le impongan estas cláusulas.

Como las licencias suelen consistir en condiciones generales, se les aplica la Ley sobre Condiciones Generales de la Contratación, que contiene normas sobre requisitos que se deben cumplir para que las

condiciones generales sean válidas y formen parte de un contrato, como el de licencia de software. Estas normas se aplican tanto si el usuario es profesional, como consumidor, e incluso si la contratación se produce vía Internet. Se exige que:

- a) El proveedor informe al usuario acerca de la existencia de condiciones generales y le proporcione un ejemplar antes de aceptarlas.
- b) Que el usuario pueda aceptarlas libremente.
- c) Además, corresponderá al proveedor del software probar que el usuario conocía previamente las condiciones generales.

Evidentemente, en la práctica el usuario no suele mirarse y estudiarse las condiciones generales (la llamada “letra pequeña”). No obstante, la ley exige que el usuario, al menos, haya tenido la oportunidad de leerlas y aceptarlas, para que dichas cláusulas generales puedan formar parte del contrato.



Las licencias *click-wrap* y, sobre todo, las licencias *shrink-wrap*, tal y como las conocemos tradicionalmente, no serán válidas en la medida en que no puedan asegurar que el usuario ha podido conocer y aceptar libremente las condiciones de la licencia.

La consecuencia de ello es que, en caso de incumplimiento de una cláusula de la licencia por parte del usuario, el proveedor sólo podrá exigir su cumplimiento al usuario (o ejercitar cualquier acción legal derivada de tal incumplimiento, como la resolución de la licencia, reclamación de daños y perjuicios, etc.) si prueba que el usuario aceptó verdaderamente dicha condición, o tuvo la oportunidad de aceptarla. Con esta exigencia de que conste una verdadera aceptación del usuario, se evita que el proveedor pueda actuar abusivamente contra él.

Sin embargo, y por el contrario, ello puede dar lugar a graves consecuencias para el proveedor del software: el usuario podrá negar estar vinculado por algunas cláusulas, alegando que no las conoce ni las ha aceptado.

Por ello, los proveedores del software han buscado fórmulas alternativas para superar los problemas de validez que presentan las licencias *click-wrap* y *shrink-wrap*:

- a) En las **licencias *click-wrap***: se aseguran de que, antes de la descarga o instalación del programa, se advierte al usuario de que el uso del programa **está sujeto a licencia**; además, también antes de la descarga o instalación, se ha de proporcionar al usuario el texto de la licencia (y asegurarse de que el usuario puede leerlo antes de seguir con el proceso, por ejemplo, abriéndose automáticamente el archivo de texto con la licencia). Para poder continuar con dicha instalación y ejecución, se exige al usuario que haga clic en la casilla “Sí” o “Aceptar”, **y se advierte expresamente que al hacerlo se acepta la licencia**.

De modo parecido, en ocasiones el usuario debe marcar una casilla en blanco en la que al lado consta una frase del estilo “Sí, acepto los términos y condiciones de la licencia”. Una vez marcada, se le hace marcar otra casilla (con la palabra “seguir”, “instalar”, “finalizar”, etc.) para llevar a cabo la ejecución o instalación. Con ello, el proveedor pretende dejar acreditado que el usuario ha tenido ocasión de conocer las condiciones de la licencia antes de aceptarla.

De este modo, la licencia *click-wrap* sí que constituiría una fórmula de aceptación válida.

- b) En **licencias *shrink-wrap***: se advierte al usuario, de forma clara, en el exterior del paquete de los CD, que el uso del programa está sujeto a aceptación de la licencia (que se halla en el interior del paquete). Se le advierte, además, que si no acepta las condiciones de la licencia, debe devolver la copia y se le reintegrará el dinero pagado.

De este modo, la aceptación no sólo se deriva de la abertura del paquete, sino al utilizar el software y sabiendo el usuario que tal utilización estaba sujeta a aceptar la licencia. Con esto, el proveedor pretende dejar claro que si el usuario utiliza el software, es porque acepta la licencia, la cual ha podido conocer antes de utilizar por primera vez el software.

**Ejemplo**

Microsoft® emplea esta fórmula en sus programas de entorno Microsoft Office®. En la parte exterior del paquete inserta la advertencia:

“Para poder utilizar este producto, usted debe aceptar los términos del contrato de licencia adjunto. Si no acepta los términos de la licencia, tendrá que devolver el producto cuanto antes para que se le reintegre el importe”.

En las licencias *shrink-wrap* es más difícil superar los problemas de legalidad planteados. Con la fórmula alternativa apuntada, el proveedor tampoco puede demostrar que el usuario conocía los términos y condiciones de la licencia (¿puede demostrar que no se le olvidó colocar un ejemplar de la licencia en el paquete si el usuario niega conocerla?). Por lo menos, debería recabar también la aceptación del usuario al inicio de la instalación y ejecución (y que aparecieran en la pantalla del ordenador los oportunos archivos de texto con la licencia), como forma de asegurarse de que el usuario ha podido conocer las cláusulas de la licencia.

- c) En el **proceso de registro de usuario**: tanto en licencias *shrink-wrap* como *click-wrap*, los proveedores intentan lograr una aceptación válida con la exigencia o solicitud de que el usuario se inscriba ante el proveedor-licenciante como “usuario registrado”.

El proceso de registro puede llevarse a cabo en línea (licencias *click-wrap*) o solicitando al usuario que envíe un formulario cumplimentado con sus datos personales a la dirección del proveedor. A veces, se advierte al usuario de que, si no completa este proceso de registro, el proveedor podrá reclamarle que cese en el uso del programa.

El proceso de registro puede dar lugar a una aceptación válida por el usuario. Ahora bien, su efectividad depende en muchas ocasiones de la buena fe del usuario: ¿se dan siempre los datos personales correctos? En la práctica, ¿cómo puede conocer el proveedor la identidad de quien se ha descargado o adquirido una copia del software sin registrarse?

### **5.2.2. Aceptación de las licencias de software libre: la GNU-GPL**

Hacemos referencia a las licencias de software libre, y en especial a la GNU-GPL, porque sus cláusulas relativas a la aceptación de la licencia también revisten alguna controversia de orden legal, en cuanto a su validez.

En concreto, la GNU-GPL establece ciertas normas para su aceptación en su cláusula 5:

- Señala que el usuario puede utilizar el software sin que sea necesario que acepte la licencia (licencia unilateral).
- Sólo obliga al usuario a aceptar la licencia si quiere modificar o distribuir el software (el original o derivado). En cualquier caso, si el usuario no la acepta expresamente, pero modifica o distribuye el software, entiende que acepta la GNU-GPL y todos sus términos y condiciones (aceptación implícita).

Según hemos visto, estos postulados de la GNU-GPL pueden entrar en conflicto con el derecho español, así como con la legislación de otros países de “derecho continental”. En primer lugar, resulta discutible que el usuario pueda utilizar un software –distribuido bajo la GNU-GPL– sin que haya aceptado la licencia. También resulta controvertido que, por la mera acción de modificar o distribuir el software, deba entenderse que el usuario ha aceptado expresamente todas y cada una de las condiciones de la licencia.

Tengamos en cuenta, además, que la GNU-GPL suele acompañar como archivo de texto al software que se licencia, sin que muchas veces el usuario tenga acceso al texto de la licencia hasta después de haber descargado el programa.



Entendemos que en derecho español (y en el resto de países que se rigen por el sistema de derecho continental) se exige que el usuario haya tenido la oportunidad de conocer las condiciones de la licencia antes de aceptarlas, y que las acepte realmente.



**Ejemplo**

Así, la exoneración de garantías y responsabilidades del licenciante sobre el software que prevé la GNU-GPL puede no vincular a un usuario que se limite a utilizar el software, en tanto que no haya aceptado dicha condición.

La GNU-GPL se redactó siguiendo las pautas e instituciones jurídicas del derecho anglosajón. Como hemos visto, éste permite la existencia de **licencias unilaterales**, sin necesidad de que el usuario-licenciatario las acepte, siempre que el texto de la licencia se limite estrictamente a regular los derechos y límites al uso que del programa puede hacer el licenciatario, sin añadir otro tipo de cláusulas.

Esto explica que la GNU-GPL (cláusula 5) estipule que el usuario “no está obligado a aceptar esta licencia” si únicamente se limita a usar o copiar el software objeto de la licencia. En este sentido, la GNU-GPL (cláusula 0, 2.º párrafo) también dispone que “el acto de ejecutar el programa no está restringido”.

Sin embargo, en España y en el resto de países que siguen el sistema de derecho continental, la licencia sólo puede entenderse como contrato.

Aun en el caso de que la licencia de software libre pueda concederse gratuitamente y de forma irrevocable (por tanto, podría considerarse que la licencia se trata de una “donación” a los usuarios), sería necesaria la aceptación por el usuario para que éste quede vinculado a las condiciones y restricciones de la GNU-GPL.

Por ello parece que la GNU-GPL debería adecuarse a las exigencias del derecho español en este punto.

La propia GNU-GPL establece en su cláusula 1 que aquel que distribuya el software debe proporcionar al receptor una copia de la licencia junto con el programa. Así pues, todo aquel que distribuya el software sujeto a la GNU-GPL debería advertir expresamente al receptor de que debe leer (o puede leer) la GNU-GPL y después aceptarla.

Incluso, el proveedor del software debería procurar obtener del usuario una **manifestación expresa de aceptación** de la licencia GNU-GPL; aunque obtenga dicha aceptación vía Internet (siempre que se asegure de que el usuario, antes de hacer clic en “Aceptar”, ha podido leer la GNU-GPL).



Para que la GNU-GPL sea plenamente válida conforme al derecho español (u otra legislación de países de derecho continental) el proveedor debería recabar la aceptación expresa del usuario antes de que éste use, modifique o redistribuya el software. Y si la GNU-GPL entiende la modificación o redistribución como aceptación de sus cláusulas por parte del usuario, éste debería estar previamente advertido (por ejemplo, asegurándose el proveedor de que el usuario tiene la oportunidad de leer el texto de la licencia) de que modificar o distribuir el programa implica aceptación de la licencia.

De cualquier modo, este posible problema legal de validez de la fórmula de aceptación prevista por la GNU-GPL u otras licencias de software libre, no tiene excesiva importancia en la práctica.

Las licencias de uso para software libre se centran en conceder libertades a los usuarios, y en asegurar que éstos las respetan; no en preservar derechos exclusivos del autor o limitar las libertades del usuario. Es más, la propia licencia señala que el uso del programa no está sujeto a la aceptación de la misma.

Por tanto, aquí no nos encontraremos con posibles abusos del proveedor sobre el usuario, que se aprovecha de una aceptación de la licencia poco clara para el usuario. Por el contrario, los conflictos pueden surgir porque un usuario pretenda apropiarse del software, distribuirlo como no libre, no proporcionar el código fuente, etc.



En este sentido, en una licencia de software libre como la GNU-GPL, es al proveedor-licenciante a quien más conviene obtener una aceptación expresa de la licencia por parte del usuario-licenciario.

Por lo demás, como la GNU-GPL no contempla restricciones al libre uso del programa (sea cual sea el número de equipos *hardware* en el que el usuario lo instala para cualquier fin, ya sea comercial, doméstico, docente, etc.), en la práctica, tampoco tiene demasiada trascendencia que el usuario –que se limite a usar el software sin modificarlo o redistribuirlo– haya aceptado o no la licencia.

**Nota**

Sin embargo, sí que puede ser problemático, pero no para el usuario sino para el proveedor-licenciante, que pueda utilizarse el programa distribuido por GNU-GPL sin necesidad de aceptar la licencia, por cuanto no podría oponer al usuario la cláusula sobre exoneración de garantías y responsabilidades (*disclaimer*). Por ello, es al licenciante a quien pueda resultar beneficioso recabar la aceptación del usuario a las cláusulas de la GNU-GPL. Otra cosa distinta es que dicha cláusula de exoneración, aun aceptada por el usuario, sea válida (sobre todo, si el usuario es consumidor).

**Nota**

Analizaremos la validez o no de la cláusula de exoneración de garantías y responsabilidades (*disclaimer*) en la unidad 6.

En cualquier caso, si el usuario infringe alguna cláusula de la GNU-GPL, como las que instauran el *copyleft*, el autor del software podrá prohibirle el uso del software (y su modificación y/o distribución). Recordemos que las diferentes legislaciones sobre derechos de autor, como la *Ley de la Propiedad Intelectual* española, no conceden ningún derecho sobre el software salvo que se cuente con autorización del autor o titular de derechos de explotación.

Por tanto, si un usuario incumple o no observa lo dispuesto por la GNU-GPL, será seguramente porque no permite a otros usuarios beneficiarse de las libertades sobre el software, y no podrá demostrar que el proveedor-licenciante le autorizó a ello.

### 5.3. Elementos objetivos en las licencias de software

Por *elementos objetivos* nos referimos a aquellos aspectos de la licencia de uso relativos a su objeto: el **software** y los **derechos** que se conceden sobre el mismo.

**Nota**

Podéis ver la descripción y el análisis en profundidad de los derechos y obligaciones de las partes sobre el software en la unidad 6.

Forma parte esencial de los elementos objetivos el contenido del derecho de uso: qué derechos concede el proveedor al usuario sobre el software y con qué límites. No obstante, dejaremos para otra unidad la descripción y análisis con profundidad de los derechos y obligaciones de las partes sobre el software. Como veremos en estas unidades, los derechos y obligaciones de las partes sobre el software variarán sustancialmente en función de si la licencia es de software propietario o de software libre.

En este apartado nos centraremos en otros elementos objetivos, distintos propiamente a los derechos y obligaciones de las partes respecto al software, pero que tienen una particular importancia en la licencia de uso: el plazo de la licencia y el precio de la licencia.

**5.3.1. Plazo**

Las licencias de software suelen establecer la duración en el tiempo de la licencia, es decir, su plazo de vigencia. En principio, salvo que alguna de las partes incumpla las obligaciones que establece el contrato de licencia de uso, éste ha de permanecer en vigor durante el tiempo estipulado.



Así, hay las licencias que se conceden por un plazo de tiempo determinado, por plazo indefinido; o bien las licencias que no estipulan nada acerca del plazo.

**Licencias por un plazo determinado**

En este tipo de licencias se establece un tiempo concreto durante el cual se puede utilizar el software (por el plazo de X meses, X años, etc.). Al final del plazo, si el contrato de licencia no dice otra cosa, la licencia de software “caducará”, por lo que el usuario deberá dejar de utilizar el software.

No obstante, ello no impide que las partes acuerden suscribir un nuevo contrato de licencia de uso sobre el mismo software. Incluso es muy posible que el propio contrato de licencia prevea que, al llegar el plazo de vigencia, la licencia se entienda prorrogada tácita-

mente por un nuevo plazo, y así sucesivamente, hasta que alguna de las partes anuncie con antelación su deseo de no prorrogar la licencia (la llamada *cláusula de renovación automática*, conocida en el derecho anglosajón como *evergreen clause*).

Las licencias que se conceden por tiempo determinado son propias del software de aplicaciones más especializado y complejo, dirigido a empresas; normalmente ligadas a un contrato accesorio como el de consultoría o mantenimiento. En estos casos, como precio, el usuario suele abonar al proveedor una cuota periódica por la licencia.



En el caso del software de demostración o evaluación (las conocidas “demos”), la licencia también se establece por tiempo determinado. En este caso, la fijación de un plazo determinado (corto, y sin posibilidad de renovación) es esencial para cumplir con la finalidad que persigue la distribución de este software: que el usuario conozca durante un breve tiempo las prestaciones del software, y que al finalizar el plazo decida o no adquirir la versión completa del programa.

### Licencias por plazo indefinido

En este caso, en el contrato de licencia de software se especifica expresamente que la licencia se concede de manera indefinida, sin sujeción a un plazo concreto. El usuario puede utilizar el software mientras cumpla con los términos y las condiciones de la licencia.

No obstante, en algunos contratos de licencias de software por tiempo indefinido se prevén cláusulas que facultan a una o las dos partes a poner fin a la licencia cuando deseen, avisando o no con cierta antelación (plazo de preaviso).

#### Nota

Como hemos señalado, cuando el usuario sea consumidor, no es válido incluir una cláusula en la licencia por la que el proveedor quede facultado a resolver en cualquier momento, sin preavisar, al usuario, salvo que la misma facultad se establezca también a favor del usuario.

#### Nota

Podéis ver el apartado 5.1.2.

Las licencias por tiempo indeterminado o indefinido son más propias del software comercializado en masa, especialmente a consumidores, en los que el usuario paga de una sola vez el precio de la licencia.

### Falta de un plazo expreso en la licencia

Cuando un contrato de licencia de software no dice nada sobre su plazo de duración (ni si se concede por un tiempo determinado o de manera indefinida), no podemos deducir sin más que la licencia se ha concedido por plazo indefinido.

Por el contrario, deberíamos entender que la licencia se limita a un plazo de **cinco años**, por aplicación del artículo 43.2. de la Ley de la Propiedad Intelectual.



El art. 43.2 de la Ley de la Propiedad Intelectual está previsto para aplicarse, genéricamente, a cualquier cesión de derecho de explotación de cualquier obra científica, artística o literaria. Como sabemos, ello incluye también el software y la cesión de su derecho de uso. Pues bien, dicho artículo 43.2 de la Ley de la Propiedad Intelectual establece expresamente que: “la falta de mención del tiempo limita la transmisión [cesión exclusiva o no exclusiva de un derecho de explotación] a **cinco años**”.

De todos modos, como el software está sometido por su propia naturaleza a una rápida obsolescencia (se queda anticuado con cierta rapidez, aparecen nuevas actualizaciones de software, etc.), el plazo de cinco años es bastante largo para que el usuario pueda utilizarlo y beneficiarse de las aplicaciones del software, con lo que aplicar esta norma no dará lugar a problemas prácticos. La “vida útil” de la mayor parte de programas es menor de cinco años, así como la expectativa del proveedor en obtener un rendimiento económico del mismo.

Sin embargo, entendemos que esta regla puede no ser aplicable a licencias de software libre, de conformidad con lo que expondremos a continuación.

### El plazo en las licencias de software libre

Ante todo, conviene resaltar que las licencias de software libre se conceden, y han de concederse, por tiempo indefinido.

Aunque modelos de licencia de software libre tan emblemáticos como la GNU-GPL o la Mozilla no lo dicen expresamente, parece claro que el espíritu de las licencias de software libre radica en garantizar a los usuarios libertades sobre el software como su uso, modificación o distribución sin otras restricciones que las necesarias para asegurar dichas libertades.

Establecer un período de duración limitado en una licencia de software libre supondría añadir una restricción a los derechos del usuario (en este caso, una restricción temporal), que iría en contra de la propia esencia de la licencia del software libre: no limitar el uso del software al usuario sino garantizarle sus libertades sobre el uso del mismo.

Así pues, es comúnmente aceptado que las licencias de software libre perduran en el tiempo mientras el usuario respete sus condiciones. Algunos consideran que a la llamada “libertad 0” del software libre, consistente en la “libertad de usar el programa, con cualquier propósito” se le debería añadir la mención “...y para siempre”.

En este sentido, la Free Software Foundation exige, en su definición y filosofía del software libre, que las libertades concedidas a los usuarios por medio de la GNU-GPL sean **irrevocables mientras no hagan nada incorrecto**, lo que confirma la idea de que la licencia de software libre se debe conceder de forma indefinida.

Por su parte, la licencia pública Mozilla, versión 1.1, incorpora una cláusula (la 8) relativa a la “terminación” de la licencia: prevé causas de terminación de la licencia, como el incumplimiento de sus condiciones, pero no el transcurso de un plazo determinado. Por tanto, podemos afirmar que esta licencia es también por tiempo indefinido.

#### Nota

Definición y filosofía del software libre de la Free Software Foundation:

[www.gnu.org/philosophy/free-sw.es.html](http://www.gnu.org/philosophy/free-sw.es.html)



Debemos entender que las licencias de software libre se conceden por plazo indefinido, aunque no se diga expresamente en el texto de la licencia.

Hemos indicado que, en el caso de las licencias que no dicen nada sobre su duración, al menos en España se debería entender en principio que tienen una duración de cinco años (al aplicarse el art. 43.2 de la Ley de la Propiedad Intelectual). Sin embargo, el espíritu y finalidad de la licencia de software libre no es tanto proteger los derechos del proveedor, sino las libertades de los usuarios. Limitar una licencia de uso, para software libre, a cinco años, podría ir en contra de la intención de las partes y la finalidad de la licencia.

### 5.3.2. Precio

Otro elemento esencial en una licencia de software es fijar el precio (cantidad de dinero) que el usuario debe abonar al proveedor por la concesión de la licencia.

Decimos que el precio es un elemento esencial en la licencia, cuando menos en las licencias de software de pago, porque en tal caso la remuneración que el usuario abona al proveedor es la causa, el motivo, por el que éste distribuye licencias de software.

No obstante, en el caso del software comercializado en masa, el precio acostumbra a no venir en el mismo texto de la licencia (que contiene sólo condiciones generales), sino en la etiqueta adherida al paquete o en la pantalla de la página web.

### Modalidades de pago

El precio admite varias modalidades de pago:

- a) Que el precio se pague de una sola vez (**cantidad a tanto alzado**): al adquirir la licencia, el usuario paga el precio completo de la misma. Es típico –pero no exclusivo– de las licencias de soft-



ware comercializado en masa. Cuando el usuario adquiere un software comercializado en masa (en una tienda o por Internet), paga el precio como si “comprara” otro artículo. Sin embargo, como ya hemos comentado, compra únicamente el soporte (el CD-ROM), y sólo adquiere el derecho a usarlo, tras aceptar la licencia.

- b) Que se pague un **canon o cuota periódica**: el usuario abona periódicamente (cada mes, cada año, etc.) una cuota al proveedor. Esto es típico en las licencias de software de aplicaciones más especializado y complejo, dirigido a empresas; establecidas por tiempo determinado y vinculadas habitualmente con contrato accesorio de consultoría o mantenimiento por el cual el usuario también abona otra cuota periódica.

Cuando la licencia puede prolongarse en el tiempo, es frecuente pactar una **cláusula de revisión o actualización** de la cuota que deberá abonar el usuario. No es válido pactar que la revisión de la cuota quede a la libre voluntad del proveedor, sino que la revisión debe realizarse bien de mutuo acuerdo entre las partes o bien teniendo como referencia un índice o parámetro objetivo como puede ser el índice de precios al consumo (IPC).

- c) También es posible que se conceda una **licencia gratuita**, sin que el usuario deba abonar nada por utilizar el software.

Hay que recordar que no debemos identificar automáticamente *software propietario* con *software de pago*, ni *software libre* con *software gratis*. En el mundo anglosajón (EE.UU., Reino Unido, etc.) sí que se ha venido dando tal confusión, porque la palabra inglesa *free* significa *libre*, pero también puede significar *gratis*.

Un software se califica como *propietario* porque el proveedor no permite al usuario modificarlo o redistribuirlo, con independencia de que el proveedor conceda la licencia gratis o mediante pago. Otra cosa distinta es que el software propietario corresponda al modelo tradicional de explotación económica del software, basado en la remuneración que el proveedor obtiene por la distribución de copias.

**Ejemplo**

Todos conocemos muchos programas que se distribuyen gratis, siendo software propietario: por ejemplo, el Microsoft® Internet Explorer, los Messenger, el software “demo”, shareware, los *drivers* que permiten instalar un elemento hardware (en este último caso, podemos entender que el software es accesorio al hardware, por lo que la licencia es gratuita o se incluye en el precio pagado por el elemento hardware de que se trate).

**El precio en el software libre**

Sabemos que el término *libre* no significa que el proveedor del software licencia el programa gratis, sino que su licencia permite a los usuarios usarlo libremente, modificarlo y distribuirlo. En el caso de la licencia GNU-GPL, el proveedor puede optar por distribuir el software gratis o a cambio de una contraprestación; también puede exigir una remuneración económica por prestar determinadas garantías sobre el software, salvo que esté obligado por ley a prestar esas mismas garantías.

No obstante, aunque el proveedor de software libre tenga derecho a pedir una contraprestación económica, lo más frecuente es que distribuya el software gratis o que el precio que solicite sea mínimo (se habla de *precio residual*), sólo para resarcirse de ciertos gastos, como el de hacer la copia, entregarla en un soporte físico, etc.



De perseguir un beneficio económico con el software libre (lo que no siempre ocurre), el proveedor no lo ha de obtener tanto por cobrar un precio por la distribución de copias, sino más bien por la prestación de servicios sobre el software, como actualizaciones, consultoría o mantenimiento.

**Nota**

Podéis ver como caso emblemático Red Hat y muchos otros.

De hecho, hay modelos de negocio alternativos a la comercialización de copias de software basados en software libre. En el mercado existen soluciones basadas en software libre que, integradas por la licencia de uso sobre el software junto con servicios de consultoría y/o mantenimiento, tienen un precio elevado.

Que el proveedor de software libre no puede basar su beneficio económico en el precio de la copia, parece evidente: si permite a los usuarios distribuir libremente el software, el proveedor pierde el control exclusivo sobre la copia. No tiene sentido cobrar un precio alto por ésta cuando los usuarios podrán distribuir todas las que quieran.

Otras licencias de software libre, como la Mozilla, declaran expresamente que la licencia se concede de forma gratuita: su cláusula 2.2 indica que la licencia se concede *royalty-free*.

## 5.4. Conclusiones

En esta unidad hemos estudiado los elementos personales de las licencias de software, el proceso de formación y conclusión (aceptación), con problemática legal existente, así como elementos objetivos tales como el plazo y el precio de las licencias. Y se ha puesto una atención particular en las especialidades que, al respecto, presentan las licencias de software libre.

De los temas analizados en esta unidad, podemos extraer las conclusiones siguientes:

- a) El proveedor-licenciante está legalmente habilitado para conceder una licencia de uso, por tener los derechos suficientes sobre el software. Así pues, algunas licencias de uso se otorgan por el autor o autores originales del software, otras por la persona (física o jurídica) titular de los derechos de explotación del software, y otras por alguien que cuenta –como mínimo– con el derecho a distribuir el software.
- b) Lo que se conoce como *garantía de titularidad*, expresión frecuente en las licencias, deriva del derecho anglosajón, y no es más que la exigencia de que el proveedor-licenciante sea un legítimo titular de derechos sobre el software, que le permitan licenciarlo a los usuarios.

Con ello, el usuario tiene la seguridad de que recibe una licencia legal, sin vulnerarse derechos de terceros. Dicha “garantía” debe cumplirla el licenciante, tanto en licencias propietarias como libres.

- c) El usuario-licenciatarario es quien adquiere derechos sobre el software por medio de una licencia, en particular el derecho de uso. Si se trata de un consumidor, existen leyes que le protegen ante cláusulas de la licencia especialmente onerosas para él y favorables para el licenciante (cláusulas abusivas).
- d) Las licencias de software libre suponen que el usuario pueda pasar a ser el autor de un software derivado, y licenciante al redistribuir el software licenciando, con o sin modificaciones.
- e) En todo caso, las licencias libres se basan en el reconocimiento del autor como medio para asegurar las libertades de los usuarios e impedir que alguien intente apropiarse del software libre. Es típico de las licencias libres obligar al usuario que distribuye un software derivado a insertar anuncios de *copyright*.
- f) Las licencias *shrink-wrap* y *click-wrap* presentan problemas de validez legal, pese a ser ampliamente utilizadas. Dichos problemas de validez han de superarse, para lo cual debe configurarse el proceso de aceptación de la licencia de modo que se produzca una aceptación válida del usuario a los términos y condiciones de la licencia, teniendo la oportunidad de conocerlos previamente a su aceptación.
- g) La GNU-GPL presenta asimismo un problema de validez legal conforme al derecho español por cuanto no requiere de la aceptación del usuario para usar el programa; y la modificación o distribución del software se considera aceptación. En cualquier caso, dicho problema es más teórico que práctico.
- h) Las licencias de uso de software se conceden bien por plazo determinado, por plazo indefinido, o bien sin precisar el tiempo de vigencia. Las licencias libres deben entenderse concedidas por tiempo indefinido.
- i) Las licencias de uso pueden concederse gratuitamente o a cambio de un precio abonado de una sola vez o por cuotas periódicas.
- j) Si muchas licencias de software propietario ascienden a un precio alto en un negocio basado en la comercialización de copias, en las licencias de software libre lo más frecuente será que se concedan gratuitamente o a cambio de un precio residual: de existir un modelo de negocio en torno a un software libre, éste se basará en el otorgamiento de licencias de uso en conjunción con la prestación de servicios tales como de mantenimiento, consultoría o actualizaciones.



