

# Introducción a las licencias de software libre

Jorge Nonius.

v. 0.92, 16 de abril de 2002

## Resumen

Este artículo introduce a los usuarios de Debian GNU, con poco o ningún conocimiento jurídico, en el Derecho español sobre propiedad intelectual y su efecto en las licencias de software libre. Continúa la serie dedicada a la propiedad intelectual que se inició con la “Introducción a la propiedad intelectual” publicada también en La Espiral. Se completa con tres apéndices en el apartado 7.

El artículo se centra en el Derecho español, pero las normas sobre propiedad intelectual de programas de ordenador son prácticamente idénticas en todos los Estados miembros de la Unión Europea.

El punto de vista adoptado no es siempre el del autor o fabricante de software, sino que más bien se les trata de igual a igual con los usuarios, consumidores y demás personas con derechos y libertades implicados en la creación, explotación y utilización del software.

© 2001, Jorge Nonius. La versión más actualizada se encuentra disponible en <http://www.laespiral.org/xml/>.

Para ponerse en contacto con el autor: [jnonius@terra.es](mailto:jnonius@terra.es). Este artículo puede ser copiado y distribuido en las condiciones de la licencia GNU para documentación libre, GFDL (<http://www.gnu.org/copyleft/fdl.html>).

[Nota de La Espiral: El autor, que firma con seudónimo, es usuario de Debian GNU y Licenciado en Derecho.]

## 1. Los derechos del autor del software

Este artículo continúa la serie dedicada a la propiedad intelectual que se inició con la “Introducción a la propiedad intelectual” publicada en La Espiral, pero, a diferencia de allí, aquí no se trata sólo de los derechos de autor, sino también de los derechos de los consumidores y usuarios, de ciertas libertades públicas, y en suma de un abigarrado conjunto de situaciones jurídicas que aparecen en los conflictos, teóricos o prácticos, acerca de las licencias de software libre. Utilizaremos siempre el término “libertad” en su acepción técnica estricta: Libertad es la situación jurídica en que se encuentra uno cuando no le alcanza una prohibición. Las prohibiciones, que adoptan muchas formas y muchas más denominaciones (deberes, obligaciones, cargas, sujeciones) provienen de muchas fuentes: directamente de la ley, por medio de un contrato, de una demanda judicial o de una sentencia, etc. De estas prohibiciones trata este primer apartado.

En el segundo apartado trataremos las libertades y restricciones a los usuarios implicadas en una licencia de software, y en especial en una de software libre. Pero antes daremos un repaso

rápido a la contrapartida de las libertades de los usuarios: los derechos del autor del software, reconocidos y garantizados por la LPI.

Vamos a explicar cuáles y cómo son los derechos del autor del programa. Por ahora trataremos al software como una obra intelectual más, sin fijarnos demasiado en aquello que lo caracteriza y distingue de, por ejemplo, una novela o una canción. Después de tratar algunas cuestiones generales (1.1), definiremos quién es el titular de los derechos de autor de un programa (1.2), sobre qué objetos recaen y sus tipos (1.3), qué es y qué implica la divulgación y la publicación del software (1.4), cuál es el multiforme contenido del derecho de autor y sus límites (1.5), su duración (1.6), las formas de explotación y cesión de derechos (1.7) y finalmente las garantías legales de todo esto (1.8).

El lector que conozca los fundamentos jurídicos de la propiedad intelectual puede saltarse este apartado 1 y pasar directamente al 2.

## **1.1. Cuestiones generales sobre el software como obra protegida**

La LPI sólo protege los programas originales generados por el intelecto. Ésta es la definición legal de programa protegible. Un programa no expresado (p. ej. una idea) no es un programa, ni tampoco lo es un programa no original, aunque veremos que se reconocen ciertos grados de originalidad en la LPI, por ejemplo para los programas derivados. No están protegidos pues los programas no originales y en cierto modo tampoco los que se encuentren en dominio público, que son aquellos para los que ha transcurrido el plazo de duración. Tampoco están protegidos los programas que, aunque originales, estén destinados a producir fallos en el funcionamiento del sistema (virus, etc). Estas exclusiones se irán detallando en su lugar más adelante, pues deben ser analizadas con cuidado. [P. ej.: La divulgación de programas inéditos que estén en dominio público genera derechos de propiedad intelectual a favor del divulgador].

### **1.1.1. El software no es patentable. Excepciones**

La propiedad intelectual de los programas se reconoce y regula en la LPI ya que el software es oficialmente considerado “obra generada por el intelecto”. Por contra, el software no tiene consideración oficial o legal de objeto patentable, pues en derecho español sólo son patentables las invenciones nuevas de aplicación industrial. Resulta que los programas de ordenador no se consideran invenciones (?), en un juicio más formal que de fondo, y por lo tanto no pueden patentarse.

Dicho de otro modo, un programa no puede ser objeto de propiedad industrial, que es el conjunto de derechos de los inventores sobre sus inventos y de las empresas sobre sus marcas y rótulos comerciales, o de los ingenieros sobre topografía de semiconductores, etc. Es una propiedad “incorporal” o “inmaterial”, lo mismo que la propiedad intelectual, pero se regula no en la LPI sino en la LP y LM, conforme a reglas y mecanismos diferentes.

No obstante, un programa protegido por la LPI puede ser también objeto de protección por la LP si forma parte de un invento patentado. En ese caso, ambas vías de protección de derechos, la garantizada por LPI y la que garantiza la LP, son independientes, compatibles y acumulables. Véase en el Apéndice B la referencia de las leyes citadas.

### **1.1.2. ¿Es el software equivalente a una obra literaria?**

No es realmente necesario etiquetar a los programas de ordenador como obras literarias, artísticas o científicas. Puede leerse en tratados internacionales y normas de la Unión Europea que los programas de ordenador han de quedar protegidos como “obras literarias”, por alguna extraña razón; tal vez porque, como no son obras científicas (?) ni artísticas (??), en algún cajón hay que meterlos (???). Al fin y al cabo, el programa fuente viene expresado en lenguaje humano, aunque sea tan poco literario como C++. Este asunto es demasiado general, y no hace falta tratarlo aquí. Mejor veamos qué protección dispensa la LPI a los programas de ordenador, comparémosla con la dispensada a las obras literarias, y concluyamos sobre las diferencias encontradas.

Adelantemos que hay diferencias, y muy notables. Probablemente a causa de que los programas no son en absoluto equivalentes a las obras literarias.

## **1.2. Los titulares de los derechos de propiedad intelectual sobre un programa**

### **1.2.1. Menores de edad y asalariados**

Los autores de programas que sean menores de edad son por supuesto considerados titulares únicos de sus derechos, igual que los mayores de edad. Pero sólo los menores de 18 años y mayores de 16 independientes -de acuerdo con sus padres o tutores- pueden ceder sus derechos de explotación del programa sin la autorización de quien les tenga a su cargo.

Si un asalariado crea un programa original durante y con motivo de su relación laboral con un empresario, se entiende que cede a éste en exclusiva sus derechos de explotación sobre el programa, salvo pacto en contra. Pero el empresario no puede disponer del software con fines distintos de los de su actividad empresarial habitual. Más adelante volveremos sobre este delicado asunto.

### **1.2.2. Programas colectivos y en colaboración**

La LPI nunca considera “autoras” de las obras intelectuales a las personas jurídicas (asociaciones, sociedades anónimas, fundaciones), sino sólo a las personas naturales o físicas, con una excepción: ¡los programas de ordenador! Técnicamente hablando hay otro caso en el que también se dice que una persona jurídica queda equiparada al autor de la obra: las obras colectivas.

Obra colectiva es un concepto difícil de definir con precisión. Para la LPI, programa de ordenador colectivo es el generado por iniciativa y coordinación de una persona (natural o jurídica), que lo edita y divulga bajo su nombre. El programa colectivo está constituido por aportaciones de diferentes programadores, de las que resulta una creación única y autónoma, sin atribución de partes o cuotas a cada aportador, y sin que uno solo de ellos pueda atribuirse derechos sobre el conjunto del programa.

Programa colectivo no es lo mismo que programa creado en colaboración, que nace del trabajo de varios coautores y permite la explotación separada de cada aportación. Volveremos sobre esta distinción enseguida, apartado 1.3.

### 1.2.3. Titulares originarios y derivados

El autor es el titular originario de los derechos de propiedad intelectual sobre su programa. Pero muchos de esos derechos, como veremos más adelante, pueden ser cedidos a otras personas, que no por ello pasan a ser autores obviamente, pero sí titulares de los derechos. Decimos en este caso que son titulares derivados, o simplemente titulares. Al hablar de titular originario diremos simplemente “autor”.

### 1.3. Tipos de programas

Los programas pueden clasificarse según varios criterios con arreglo a la LPI:

1. Por la autonomía del programa tenemos programas independientes y programas dependientes.
2. Por el número de autores y su forma de cooperar tenemos programas individuales, programas en colaboración y programas colectivos.
3. Por su originalidad tenemos programas estrictamente originales por un lado y programas derivados y compuestos por otro.

Ahora nos interesa sólo dar algunas definiciones. Llamamos programa independiente al constituido como una “creación autónoma”, aunque se publique conjuntamente con otros programas. Se distingue del programa compuesto, formado por varios programas independientes preexistentes.

Decimos que un programa es realizado en colaboración si resulta unitariamente del trabajo de varios desarrolladores, en el que es posible separar las aportaciones de cada cual y de explotarlas independientemente. En este caso, los programadores son co-autores, y pueden entre ellos pactar lo contrario y explotar por su cuenta cada cual su parte. Si no hay acuerdo, el único límite a la explotación separada consiste en no perjudicar la explotación común. Para divulgar y modificar un programa en colaboración hace falta el consentimiento de todos los coautores, que sólo el juez puede excusar. Los derechos de autor pertenecen a cada coautor en la proporción que entre ellos pacten; en otro caso, se aplican las reglas generales del Código Civil sobre la comunidad de bienes.

Programa derivado es el que se ha obtenido de un modo u otro de software anterior, p. ej. traduciéndolo, adaptándolo, modificándolo o revisándolo. En general debe hablarse de programa derivado ante cualquier transformación de un programa preexistente. Pero cuidado: La LPI protege los derechos de los dos autores: el del programa primitivo y el del derivado.

Un programa se dice “compuesto” si se ha obtenido de la incorporación de uno o más programas preexistentes y sin la colaboración de los autores originarios. Se considera obra protegida siempre que haya autorización de los titulares de los programas originarios y se respeten sus derechos sobre ellos. Es decir: el autor del programa compuesto tiene derechos sólo sobre la composición, no sobre el software que la compone. La distribución Debian GNU/Linux Potato, p. ej., es una obra compuesta, compuesta de software. Debian sólo tiene derechos sobre la composición en sí, no sobre los programas independientes incluidos en la distribución.

Un programa (obra intelectual) se distingue del soporte en que está contenido (bien mueble, como puede ser un CD). El soporte del programa es el material en que se plasma, no es lo mismo que el programa. [Advertencia probablemente superflua: El significado de soporte al que nos referimos nada tiene que ver con el utilizado constantemente en informática de “servicio de apoyo”]. Lo importante es que son distintos e independientes los derechos sobre el programa (derechos inmateriales, de propiedad intelectual) y los derechos sobre el CD (derechos materiales, de propiedad común). Al cederse los derechos de propiedad intelectual no necesariamente se ceden los derechos sobre el soporte. Viceversa y más importante: ser dueño del soporte no significa ser titular de los derechos sobre el programa que incorpora.

#### **1.4. Divulgación y publicación de un programa**

Divulgar un programa es expresarlo de modo que se haga accesible al público por primera vez en cualquier forma. La divulgación es facultad exclusiva y personalísima del autor, se dice incluso que es un “derecho moral” (véase más adelante). Lo importante de todo esto es la fecha de divulgación, porque a partir de ella se cuenta el plazo de duración de los derechos de propiedad intelectual.

La publicación del programa es una forma de divulgarlo, de las más importantes pero no la única. Publicar un programa es expresarlo de modo que lo hace accesible al público mediante ejemplares o copias. Verdaderamente es la forma principal de divulgación del software, por eso no trataremos otras, como la comunicación pública, apenas concebible en el ámbito de los programas de ordenador. No obstante, véase el apartado 1.7.

#### **1.5. Facultades del autor del programa y sus límites**

Los derechos del autor se manifiestan ante todo en dos grupos de facultades: 1º Los derechos morales, o facultades personalísimas que tiene sobre los programas que ha creado; y 2º Los derechos patrimoniales, como la facultad exclusiva de explotarlos en cualquier forma y obtener remuneración por ello; o el derecho a obtener remuneración por el simple acceso a las fuentes; y el derecho a autorizar o prohibir su uso, divulgación y explotación; etc.

El derecho moral es una figura que sólo encontraremos en los derechos continentales, no en las leyes anglosajonas, al menos con el mismo aspecto. No puede cederse en vida, como parece deducirse del texto de la LPI. En realidad son varios los derechos morales del autor:

1. Decidir si su programa ha de divulgarse y en qué forma;
2. Decidir si el programa aparecerá con su nombre, bajo seudónimo o anónimamente;
3. Exigir el reconocimiento de su condición de autor del programa, y el respeto a su integridad, sin deformaciones, modificaciones o atentados que perjudiquen el interés o reputación del autor;
4. A modificar el programa cuando le plazca, aunque ha de respetar los derechos adquiridos por otras personas. Volveremos sobre esto al tratar de las modificaciones de los programas;

5. A retirar su programa por cambio de convicciones (derecho de arrepentimiento), indemnizando a quienes perjudique la retirada, normalmente los usuarios y el explotador del programa. Esta facultad y las dos siguientes no es probable que un programador las ejercite nunca;
6. A acceder al ejemplar único o raro de su programa que se halle en poder de otra persona, indemnizando los posibles perjuicios;
7. A publicar su programa en colección escogida o completa.

Para más detalles sobre el derecho moral, véanse los artículos 14 a 16 LPI.

Los derechos patrimoniales son los que tienen relevancia económica. Los trataremos muy sintéticamente en el apartado 1.7, dedicado a la explotación del software.

### **1.6. Duración de los derechos. Programas en dominio público**

Los derechos de propiedad intelectual nacen con la simple creación del programa, no es preciso anunciarlo ni registrarlo. Pero, como en las demás propiedades intelectuales, los derechos no duran indefinidamente; se disfrutan por un tiempo y después se extinguen; se dice entonces que el programa pasa al dominio público.

Esto es esencial, al menos en teoría. Veamos: Una vez creado el programa, nacen los derechos de propiedad intelectual sobre él, que duran toda la vida del autor y 70 años tras su muerte, contados desde el 1 de enero del año siguiente al de la muerte, y después se extinguen. Hay reglas especiales para los programas anónimos y seudónimos, los realizados en colaboración y los programas colectivos, los programas publicados por partes, que no detallaremos aquí (véanse los artículos 26 y 30 LPI).

Sin embargo, el derecho moral dura toda la vida del autor, pero sólo dos de sus facultades duran después indefinidamente sin límite de tiempo: exigir el reconocimiento de la autoría y exigir la integridad del programa. El resto de las facultades se extinguen con la muerte del autor, salvo la divulgación del programa inédito durante su vida, pero éste es un caso muy extraño y tampoco lo trataremos.

Cuando los derechos de explotación se extinguen por transcurso del plazo, el programa pasa al dominio público, es decir, puede ser utilizado por cualquiera siempre que respete la autoría e integridad del software. Trataremos de nuevo el dominio público, con más fundamento, en el apartado 4.

### **1.7. Formas de explotación del programa. Cesión de derechos**

Explotar un programa es difundirlo en cualquier forma con obtención de beneficio. Comprende todas las modalidades posibles de ganar utilidad con el programa, pero las más importantes son las formas de explotación tipificadas por la LPI, que son las habituales: fijación o grabación, reproducción, transformación y distribución.

Los beneficiarios de la explotación son en principio los autores, a quienes se llama también titulares originarios de los derechos de propiedad intelectual sobre el programa, pero es muy

normal ceder la explotación a empresarios especializados, quienes pagan al autor por ello. A estos les llamamos titulares derivados.

No podemos ver aquí en detalle cuánto hay detrás de las reglas sobre explotación de los programas, recomendamos al lector interesado que acuda a la “Introducción a la propiedad intelectual” publicada en La Espiral. Nos arreglaremos con una sinopsis:

- Todo el que no sea titular ha de obtener autorización para explotar el programa, salvo en casos tasados que veremos después. Al titular corresponde la facultad de explotar el software, con los medios presentes o futuros, ya que los tipos legales (reproducción, distribución, transformación) sólo son algunos de los posibles. Cualquier otra modalidad corresponde siempre y en exclusiva al autor, mientras no la ceda a otra persona. Cada modalidad de explotación es además independiente una de otra.
- No hemos citado una modalidad de explotación, la comunicación pública, porque es dudoso que sea apta para el software, como sí lo es para, por ejemplo, una obra musical. Remitimos al lector al otro artículo de esta serie para algunos detalles sobre esta jurídicamente compleja forma de explotación, de todos modos seguramente inaplicable a los programas, pues no hay forma de acceder a ellas si no es mediante copia.
- Las formas típicas de explotación son, repetimos, la fijación, la reproducción u obtención de copias, la modificación y la distribución. Nos ocuparán el resto de este artículo, así que ahora no diremos mucho sobre ellas.

Pero lo dicho no es suficiente. A riesgo de resultar esquemáticos en exceso, aunque con la seguridad de no dejar cosas importantes sin atender, recapitulemos las facultades del autor de un programa (se encuentran principalmente en el art. 99 LPI):

1. Derecho exclusivo de autorizar o prohibir la divulgación del programa, derecho al reconocimiento de la autoría, y demás derechos llamados morales. Los trataremos con un enfoque diferente en el apartado 4.2.
2. Derecho exclusivo de explotación del programa. Aquí comienzan los obstáculos para las libertades de los usuarios que expondremos después. Algunos de ellos no sólo igualan, sino exceden, las facultades de los autores de los demás tipos de obras. Resulta que la explotación de un programa de ordenador se entiende que incluye:
  - a) La reproducción incluso para uso personal, o sea: la copia privada, que por tanto está expresamente prohibida. Es ilícito copiar programas sin autorización del autor, autorización normalmente expresada en una licencia. La prohibición de copia es muy completa: Cuando la carga, presentación, ejecución, transmisión o almacenamiento de un programa requiera copiarlo (reproducirlo), debe disponerse de autorización del autor;

- b) La transformación y su reproducción. De todos modos, es muy difícil transformar un programa si se carece del código fuente, código que el autor no tiene en modo alguno obligación legal de ceder a nadie;
  - c) La distribución pública. Pero como incluso la reproducción privada está prohibida, como acabamos de ver, resulta que también está prohibida o imposibilitada la distribución privada, aunque la LPI no lo diga expresamente.
3. Cuando se produce la cesión del derecho de uso, es decir cuando tenemos a un usuario legítimo (después definiremos este concepto), se entiende que es una cesión no exclusiva -el autor del programa puede ceder el uso a más personas, o crear más usuarios legítimos, dar más licencias en suma-; se entiende que la cesión es intransferible -el usuario no puede dar licencias a su vez-; y la finalidad de la cesión es satisfacer las necesidades únicamente del usuario y de nadie más.

Este es el panorama con que se enfrentan las licencias de software libre, que vienen a subvertir los términos: No limitar al usuario, que explote el programa a su entero placer, sin restricciones. ¿Permite esto la LPI española? ¿No se encontrará una licencia de software libre, y especialmente las copyleft que son las más interesantes desde el punto de vista teórico-jurídico, con el muro infranqueable de algún derecho del autor que sea inviolable, ni siquiera contando con la propia voluntad del autor? A esto tratan de responder los apartados siguientes. Pero antes, y para rematar el cuadro, trataremos brevemente el sistema de garantías de los derechos de autor.

## 1.8. Garantías legales de los derechos del autor del programa

### 1.8.1. Registro de programas

Ya sabemos que la autoría se reconoce por la simple creación del programa, no es preciso inscribirlo en ningún Registro. Pero el hecho es que tener un programa registrado refuerza muchísimo la prueba de la autoría (art. 101 LPI). Los programas no se patentan, no pueden patentarse, pero sí pueden registrarse. Por cierto que el software libre puede registrarse, precisamente como software libre, y esto no quita nada a la licencia, que sigue siendo de software libre en sus términos literales. El Registro se limita a dar publicidad de su existencia y a dar fe de su validez. De hecho es la misma licencia lo que se inscribe.

Pero todo esto es teoría. Los programas se inscriben en la llamada Sección VII del Registro General de la Propiedad Intelectual, cuya organización y funciones básicas se exponen en la “Introducción a la propiedad intelectual”. En realidad sólo se inscribe en él una descripción del programa o la determinación de los elementos que permiten su completa identificación, que se entiende contenida en las diez primeras y diez últimas hojas del código fuente (?), o en un resumen de un máximo de 20 folios del manual de uso (??), “siempre y cuando [dice el Reglamento del Registro] éste reproduzca los elementos esenciales del programa” (???) (arts. 13 y 14.7 del Reglamento de 1993). Si el programa es inédito (o sea: si no se ha publicado) entonces debe adjuntarse todo el código fuente (????). Estas son las reglas. No se dispone de datos acerca del uso que los programadores hacen del Registro, pero al parecer sí lo usan.



¿Cómo da publicidad el Registro a los programas inscritos? Esto tampoco es lo que parece: El Registro es ciertamente público, pero de un programa inscrito sólo podemos consultar los datos personales del autor y sus derechos sobre el programa (esto es: la licencia si existe). Y sólo nos proporcionarán el título y la fecha de publicación. Por supuesto, jamás nos permitirán consultar el código fuente ni los manuales. Así lo dice el art. 32 del Reglamento.

### 1.8.2. Infracciones de los derechos del autor del programa

Hay que estar de acuerdo con la FSF en que las leyes de copyright presentan como infracciones lo que, visto desde el punto de vista del usuario del software, son libertades truncadas que éste podía esperar disfrutar en el uso normal de un programa, libertades que de hecho se reconocen para otro tipo de obras. Una vez uno compra un libro (soporte físico de una obra intelectual, literaria tal vez) no necesita licencia especial para leerlo. ¿Por qué ha de ser así con el uso del software? Volveremos después sobre esto. Pero nuestro esquema no estaría completo si no se citaran aquí lo que la LPI llama claramente infracciones del derecho de autor. [Un tratamiento más extenso del sistema de protección de los derechos de autor se encuentra en la “Introducción a la propiedad intelectual”]. Ahora nos centraremos en las infracciones de los derechos de autor de los programas de ordenador. Se encuentran especificadas en el art. 102 LPI y son éstas (asústese el lector):

- Poner en circulación una o más copias de un programa, conociendo o pudiendo presumir su naturaleza ilegítima (no tener licencia para copiarlo). Está prohibido copiar un programa de ordenador sin licencia. Esta es la regla y sólo una licencia puede excusar de su cumplimiento. En los apartados 4.3 y 6.2 trataremos este asunto de nuevo.
- Almacenar o simplemente tener con fines comerciales una o más copias ilegítimas.
- Poner en circulación o tener con fines comerciales instrumentos cuyo único uso es facilitar la supresión o neutralizar sin autorización cualquier dispositivo técnico utilizado para proteger un programa de ordenador. ¿Hay que entender que además de los virus, este tipo de software tampoco es obra protegida? No, pues la ley sólo se refiere a ponerlo en circulación. Éste es de todos modos un caso dudoso.

En fin, si no se arregla de otro modo y por las buenas, el autor o titular que considere violados sus derechos ha de acudir al Juez de 1ª Instancia de la localidad en donde se haya producido la infracción, al que pedirá que se condene al infractor a devolverle el beneficio ilícito, a indemnizarle por los perjuicios, a detener la actividad ilegal e impedirle que pueda reanudarla. Entretanto estudia el litigio y dicta sentencia, el juez puede adoptar medidas cautelares que llegan a ser muy graves para el presunto infractor, como el secuestro de los equipos y materiales de reproducción y copia, etc. No podemos entrar en detalles, además muy técnicos y farragosos, fuera del objetivo de esta “Introducción”. Para las infracciones de las licencias de software libre, véase el apartado 4.7.

## **2. Las libertades de los usuarios**

Después del examen general del apartado anterior, toca ahora cambiar el enfoque para tratar un tipo muy especial de fórmula de cesión de derechos de explotación todavía poco conocida en los medios profesionales jurídicos: la licencia de software libre. Para esto es necesario partir de las libertades del usuario del programa, no de los derechos de autor. Éstos se encuentran protegidos por la LPI, como hemos visto en el apartado 1. Las libertades de los usuarios por contra tienen sus garantías (en la Constitución y en otras leyes que se citarán) muy difuminadas y dispersas, no en un único cuerpo legal sistemático.

Para empezar despejaremos algunos problemas de nomenclatura. Reservaremos el término software libre (“free software”), que abreviaremos *sl*, para los programas que se ajustan a la especificación de la Free Software Foundation, que es la más rigurosa, pero puede utilizarse también como denominación genérica del conjunto de licencias que liberan las facultades típicas del copyright básicas para la libre utilización del software, aunque no se ajusten estrictamente a la definición de la FSF y siempre que del contexto se deduzca a qué nos referimos (en los apartados 3.1 y 4.1 se encuentran las definiciones). Sin embargo, no nos atenderemos necesariamente al criterio de la compatibilidad de las licencias con la GPL, no porque este punto de vista no tenga importancia, en realidad es tal vez el más importante, sino porque la propia FSF dispone de documentación apropiada -véanse las referencias al final en el apéndice C- y porque aquí deseamos examinar solamente la compatibilidad de las licencias de software libre con el Derecho español.

Huelga decir que el objetivo es el de saber a qué atenernos en España cuando surjan conflictos de intereses relacionados con el software libre. Y es que la FSF opera obviamente con categorías jurídicas anglosajonas, sutilmente diferentes de la europeas continentales en lo sustancial, y decididamente distintas en los formalismos. Tenemos que asegurarnos de que hablamos consistentemente, pues las palabras son importantes. Para empezar, los hispanohablantes no tenemos ningún problema en distinguir algo que es gratis de algo que es libre, pequeñez que a los angloparlantes les ha costado mucha tinta.

El primer objetivo es entonces disponer de un vocabulario apropiado, no obligatoriamente castellanizado, por exigencias prácticas obvias y que sirva para entendernos en nuestras discusiones.

Pero hay un segundo objetivo: Comprender lo mejor posible las licencias de software, que es el instrumento utilizado por el movimiento del *sl* en general, y por el copyleft en particular, para articular jurídicamente un fenómeno que sobrepasa el ámbito del software, alcanza a la documentación técnica y científica y comienza a sustentar la distribución de otro tipo de bienes y productos (véase el apartado 5). No es que software libre y copyleft hayan surgido de las licencias, sino que éstas han “instrumentado” tales movimientos, se han servido de ellas para recuperar las libertades académicas, científicas y de los usuarios. De hecho, parece que el software libre es incluso un modelo de negocio, pero también es un fenómeno social, un método de investigación y de docencia.

Orbitando las licencias de software se encuentran muchos asuntos que no podemos tratar, pero tan importantes que se citan a continuación algunos:

- Conflictos graves, comerciales o no, entre distribuidores de software, p. ej. el pleito ATT-Berkeley de 1992, o la cuestión GNU v. KDE de 1997.
- Aparición de sistemas operativos libres, con representantes como la rama BSD o GNU/Linux, en competencia insospechada con los grandes fabricantes de software cerrado (expresión que usaremos en lugar de software propietario, más adelante se explica por qué).
- Alianzas expresas o tácitas entre fabricantes de software libre y hardware, en competencia también inesperada con los chips dominantes. Es el caso del acuerdo Apple-Universidad Carnegie Mellon para el MacOS X, o el impulso dado por el software libre a los chips Alpha, SPARC...
- Nuevos modelos de negocio, como el de Cygnus con el compilador GCC, acuerdos de Red Hat con Penguin Computer y con la misma Cygnus Solutions (noviembre de 1999).
- La aparición de un instrumento incomparable de colaboración profesional y científica, y de fórmulas como la FDL para la transmisión de documentación técnica. Véase sobre esto el apartado 5.
- Iniciativas legislativas, como la reciente sobre uso de software libre en la Administración pública del Perú, 9 de abril 2002.

Todos estos son asuntos del mayor interés, en el apéndice C se encontrarán algunas referencias. También se trata de cuestiones complejas, pero alejadas en cierto modo de nuestro tema, mucho más restringido: las licencias de software según el Derecho español, y especialmente las de software libre.

Comenzaremos con unas cuantas frases fuertes y un esquema lo más breve posible, para comodidad del lector, de lo que ya se ha ido apuntando con otro enfoque en el apartado 1, es decir: Qué exige la LPI española a las licencias de software libre para considerarlas viables o atendibles por los jueces (éstos como último recurso, claro está).

## 2.1. Premisas del software libre

El sistema jurídico español, como todos los europeos y anglosajones de corte “constitucional”, se basa en la libertad, en el sentido de que uno puede hacer lo que guste mientras no esté prohibido. Esta es una afirmación muy general, pero nos sirve para enfocar la cuestión tal y como interesa, o sea: desde el punto de vista de las libertades del usuario, y no el de los derechos de autor. Este último, así lo esperamos, ha quedado expuesto ya en el apartado 1. Para más detalles, véase la “Introducción a la propiedad intelectual” publicada en La Espiral.

En nuestros modernos sistemas legales se entiende que la libertad tiene límites, no se garantiza la libertad absoluta. Un género esencial de esos límites a la libertad son los derechos y libertades de los demás. Así que nuestra primera afirmación queda “uno puede hacer lo que guste mientras no dañe los derechos y libertades de los demás”. Por supuesto, los derechos y libertades de los demás tampoco son absolutos.

### 2.1.1. Las libertades del usuario de software

Algunos derechos y libertades se consideran fundamentales, se les garantiza una protección reforzada sobre los demás derechos y libertades ordinarias. De entre ellos, las licencias de software se encuentran con los siguientes, que vamos a clasificar en tres grupos:

1) En el primer grupo tenemos los fundamentos de nuestra sistema político. No se trata de un auténtico reconocimiento de libertades y derechos fundamentales, sino de su basamento:

- La libertad es el fundamento de todo lo demás. Se ha dicho muchas veces, y no se trata sólo de una dificultad idiomática del inglés, que el concepto software libre trata de la libertad, no del precio. El concepto de libertad es calificado por nuestra Constitución, nada menos que en el primer párrafo de su primer artículo, como uno de los valores superiores del ordenamiento jurídico español, junto con la justicia, la igualdad y el pluralismo político.
- Es cometido de los poderes públicos promover las condiciones para que la libertad y la igualdad de los individuos y grupos sean reales y efectivas; remover los obstáculos que impiden o dificultan su plenitud; y facilitar la participación de todos en la vida cultural (art. 9.2 CE).
- El libre desarrollo de la personalidad es uno de los fundamentos de nuestro orden político (art. 10.1 CE).

2) El segundo grupo es el más importante a efectos prácticos. Se trata de las libertades y derechos fundamentales que directa y necesariamente los jueces han de proteger. Los más importantes son los tres últimos de los que se citan a continuación, directamente esgrimibles ante argumentos del tipo “el software libre atenta contra la libre expresión, embota la creatividad y vulnera el copyright”, etc. No es así, sino justamente al contrario. En términos jurídicos, el usuario de cualquier software debería poder argüir conforme a los siguientes ítems, que incluso tienen protección de amparo garantizada hasta el recurso ante el Tribunal Constitucional, y aplicables según las circunstancias. Son los siguientes:

- Derecho a la igualdad y no discriminación por condición social (art. 14 CE).
- El uso de la informática será limitado por la ley para garantizar el pleno ejercicio de los derechos de los ciudadanos (art. 18.4 CE). Este artículo se incluyó en la Constitución con una finalidad relativamente clara: evitar que mediante la informática se alcanzara un control excesivo sobre las personas, y con la misma finalidad se promulgó la Ley de protección de datos personales de 1998. Pero cabe una segunda interpretación interesada, y un poco traída por los pelos, pero no irrazonable: Las limitaciones al uso del software, especialmente algunas cláusulas abusivas (véase el apartado 6), dificultan de tal modo el pleno ejercicio de los derechos de los consumidores y ciudadanos en general que deben quedar restringidas por la ley [Me complace hacer constar aquí que estando ya estas líneas escritas, encuentro este mismo argumento como motor del proyecto (proposición) de ley sobre uso del software libre en la Administración pública remitido al Congreso peruano en abril de 2002 por

los congresistas VILLANUEVA NÚÑEZ y RODRICH ACKERMAN, así como en la admirable carta que el primero dirigió al gerente de Microsoft del Perú. Los textos se pueden encontrar en <http://www.gnu.org.pe/rescon.html>]

- Derecho a la libre expresión de pensamientos e ideas (art. 20.1.a CE).
- Derecho a la producción y creación literaria, artística, científica y técnica (art. 20.1.b CE).
- Derecho a la educación y a la libertad de enseñanza. Aunque la libertad académica, y una de sus componentes, la libertad de estudio, no figuran expresamente en el art. 27 CE, puede entenderse que se encuentran implícitamente reconocidas por la Constitución, pues son necesarias y congruentes con el modelo docente general que pretende garantizar.

3) Finalmente tenemos los principios rectores de la política social y económica. No son derechos y libertades fundamentales propiamente dichos, sino directrices que, aplicadas a éstos, han de inspirar su efectividad y garantía. Se trata concretamente del mandato que la Constitución contiene en el art. 44, dirigido a los poderes públicos, de promover y tutelar el acceso a la cultura, a la que todos tenemos derecho, así como a la ciencia y la investigación científica y técnica en beneficio del interés general.

### 2.1.2. El derecho de autor no es un derecho fundamental

Por contra, y desde el punto de vista del autor del programa, no puede decirse que haya derechos fundamentales implicados. No es pensable una lesión a los derechos de propiedad intelectual que afecte también a un derecho fundamental de los que acabamos de citar, ni a ningún otro, con una sola excepción, por lo demás bastante rara en la práctica: el atentado contra el honor, la reputación, la imagen del autor, derivado de una infracción de los derechos de propiedad intelectual (véase el art. 18.1 CE). Este es un caso poco usual y no vamos a tratarlo, salvo algunos apuntes en 4.5.2.

En realidad, sí tiene que ver la propiedad intelectual con un derecho fundamental, pero de los que la Constitución considera de segunda categoría, el derecho de propiedad (art. 33 CE). No puede negarse que tanto el fenómeno del software libre, como más acentuadamente el copyleft, parecen superficialmente ir directos contra el derecho de propiedad (intelectual). Nada más falso. El software libre se basa en el derecho de autor para, sobre él, modular sus facultades intrínsecas a las necesidades prácticas de las libertades que hemos citado antes, sin machacarlo en modo alguno. Simplemente el software libre tiene copyright.

Asimismo, una licencia copyleft, que impide la redistribución de software con restricciones añadidas a las de la distribución originaria, no atenta contra el derecho de autor del programa originario, pero esto no es obvio y en lo que sigue tratará de demostrarse. Tampoco atenta contra los derechos del autor del programa derivado, que modificó el software porque la licencia copyleft se lo permitía, si no no hubiera podido hacerlo; y se ve obligado a redistribuirlo con licencia copyleft por el mismo motivo, es decir: por haber aceptado previamente una licencia copyleft, una decisión voluntaria y libre.

Naturalmente que cuando uno acepta una licencia (copyleft o no) ve limitadas algunas de sus libertades y derechos, se dice que asume obligaciones, deberes y sujeciones, lo mismo que cuando se casa o cuando firma un préstamo hipotecario. Simplemente acepta de forma libre los términos que se le ofrecen. En este sentido puede decirse que la GNU-GPL es un tratado de desarme (WAYNER), porque da total libertad a todos, salvo a quien quiere apropiarse -para sí y con exclusión de los demás- de la libertad que recibió, la que le permitió y permite explotar el programa.

El derecho de propiedad, decimos en España, queda delimitado por su función social de acuerdo con la ley (art. 33 CE). No puede haber otra finalidad del Derecho de propiedad intelectual que garantizar al autor la percepción de los beneficios de su explotación, lo que es perfectamente acorde con los postulados del software libre. Pero tampoco hay función social de la propiedad del software que no sea su libre uso y explotación por quien sepa hacerlo.

No debe olvidarse que el autor tiene derecho al honor y a la propia imagen, ya se ha dicho; pero también él mismo está manifestando, al escribir código, su libertad de expresión de pensamientos e ideas y su derecho a la producción científica y técnica.

En resumen, las licencias de software son expresión de un derecho individual ordinario: el derecho de autor. Aunque la autoría de una obra tiene mucho que ver con algunos derechos fundamentales, es muy dudoso que, fuera de los aspectos relacionados con la reputación del autor, los demás sean considerados por un juez como expresión de sus derechos fundamentales. Por contra, los usuarios sí pueden esgrimir sus derechos y libertades fundamentales frente a ciertos atentados contra el software libre (véase el apartado 4.7).

### 2.1.3. Las normas imperativas de la LPI

Los límites de las libertades de la gente sobre los programas de ordenador, y los derechos sobre ellos de sus autores o titulares, se encuentran principalmente en la LPI, aunque no solamente. Debe tenerse en cuenta también toda la legislación existente sobre protección de los consumidores y usuarios, competencia desleal, condiciones generales de la contratación y tantos otros asuntos. No se hará aquí así, nos limitaremos al campo de los derechos de autor, o de propiedad intelectual. Más adelante añadiremos de todos modos algunas notas sobre estas cuestiones.

La LPI está pensada sobre todo para proteger a los autores, es decir, sus derechos ordinarios sobre la obra; no para proteger las libertades de los demás (usuarios, otros programadores), aunque no falten artículos que garantizan algunas, muy escasas e indefensas, como vamos a ver.

Efectivamente, ya se habrá advertido que hay reglas claramente limitativas a los usuarios o destinatarios de los programas o a quienes los explotan; y otras por contra limitan a los autores. Esto es lo que puede esperarse de un sistema jurídico que no admite libertades o derechos absolutos. Pero lo importante ahora está un paso más allá: Hay reglas de la LPI que son solamente indicativas, pueden no seguirse sin cometer ninguna ilegalidad (se llaman reglas “dispositivas”); y también hay reglas que necesariamente han de seguirse, a riesgo de que después el sistema no te proteja si las infringes, se llaman reglas “imperativas”.

Uno no puede saltarse las reglas imperativas de la LPI impunemente. ¿Cómo se sancionan sus infracciones? Depende del grado de la infracción, pero para resumir diremos que va desde tener

la falta por inexistente -como una cláusula inválida de una licencia, simplemente no se aplica hasta la prisión -desde luego, sólo en casos muy graves y poco frecuentes. Ahora lo que interesa es recalcar que si una licencia de software contiene cláusulas contrarias a las normas imperativas de la LPI, tales cláusulas no valen, incluso si el perjudicado hubiera dado su acuerdo para aceptar la licencia (p. ej., porque desconocía que tales cláusulas eran ilegales).

**Beneficios irrenunciables de los autores de software.-** Aunque una licencia de software libre no implica renuncia alguna, al menos en sentido técnico, conviene aclarar algunas cuestiones que al parecer sus críticos mantienen en reserva. Para empezar, en España es en general posible la exclusión voluntaria de la ley aplicable, lo mismo que la renuncia de derechos reconocidos en la ley, a condición de que no se contraríe el interés o el orden público ni se perjudique a terceros (art. 6.2 CC). Lo que no se permite es obligar a nadie a renunciar a sus derechos irrenunciables, y justamente para prevenir esta posibilidad se establecen las reglas que se citan a continuación, pensadas para proteger al autor de contratos leoninos con empresarios sin escrúpulos, no para proteger a distribuidores de software dominantes frente al desorganizado público de usuarios, en su mayoría desconocedores de las posibilidades inauditas de sus máquinas.

Por lo tanto, interesa saber cuáles son esas normas imperativas. Relacionarlas todas no es fácil ni por suerte tampoco muy útil. Basta conocer las más importantes. Hay de todo: beneficios renunciados sólo por acuerdo de las partes, ventajas irrenunciables... Nos quedaremos únicamente con las reglas imperativas relevantes sobre los programas de ordenador:

1) Los derechos morales son irrenunciables y no transmisibles (artículo 14 LPI). Veremos más adelante que las licencias de software libre no afectan a esta limitación, véase el apartado 4.2.

2) La cesión de derechos de autor no puede alcanzar nunca a las modalidades de utilización o medios de explotación o difusión inexistentes o desconocidos al tiempo de la cesión (artículo 43.5 LPI). En general, para la cesión de derechos y el efecto de esta regla sobre el software libre véase el apartado 4.2.

3) También son irrenunciables los beneficios que la LPI otorga a los autores en los actos de transmisión de sus derechos, o contratos de cesión de derechos de autor. Así lo dice el artículo 55 de la Ley. Esto tiene mucha más importancia, y de hecho algunos de los siguientes ítems nos dará algún trabajo después. Pero en general las licencias de software libre no se ven afectadas por estas reglas sobre “derechos irrenunciables”, aunque parezca paradójico. Piénsese que al fin y al cabo la GPL (p. ej.) no supone renuncia alguna para el autor, sino la cesión voluntaria de sus derechos transmisibles. Todo esto se tratará después, ahora nos limitaremos a enumerar, sólo aproximativamente, los beneficios irrenunciables de los autores al ceder sus derechos:

- Es nula la cesión de derechos de explotación respecto del conjunto de obras que pueda el autor crear en el futuro (art. 43.3).
- Es obligatorio documentar las cesiones (art. 45).
- Durante 10 años desde la cesión de la explotación, el autor puede exigir la revisión de la cantidad con que se le ha remunerado si considera (y logra probar) que es inequitativa o

desproporcionada al beneficio obtenido por el cesionario o explotador del programa (art. 47).

- Supongamos que Juan cede sus derechos a Pedro mediante la licencia L-1, y Pedro los cede a su vez a otro mediante L-2. Si L-2 no respeta los términos de L-1, esta primera licencia puede quedar sin efecto y dejar de amparar a Pedro, a requerimiento de Juan (art. 68.1.d LPI).
- Las cesiones no exclusivas son intransmisibles (art 50.1).
- En general, las obligaciones de los editores suponen en muchos casos -en el polo opuesto- derechos irrenunciables del autor del programa.

**Garantías a favor de los usuarios.-** Por otro lado están las normas imperativas de la LPI que establecen garantías a favor de los usuarios, insoslayables para el autor. A diferencia de antes, algunas garantías no desaparecen ni siquiera mediante pacto en contra, pero otras sí (se encuentran en el art. 100 LPI). Las primeras son éstas:

- El usuario legítimo -véase más adelante su definición- siempre ha de poder hacer una copia de seguridad, si es necesaria (?) para la utilización del programa.
- El usuario legítimo siempre puede observar, estudiar y verificar el funcionamiento del programa para determinar las ideas y principios implícitos [algoritmo] en cualquier elemento, mientras lo haga durante cualquiera de las operaciones de carga, visualización, ejecución, transmisión o almacenamiento del programa que tiene derecho a hacer.

Las siguientes son facultades del usuario legítimo, pero son renunciables mediante pacto contrario entre el usuario y el autor:

- Puede reproducir o transformar un programa, incluida la corrección de errores, si es necesario para su utilización legítima y conforme con la finalidad. Pero cabe pacto en contra, que es lo normal, pues mal pueden corregirse los errores sin tener a mano el código fuente.
- El autor no puede oponerse a que el titular de derechos de explotación realice o autorice versiones sucesivas y programas derivados. El lector ha leído bien: la LPI garantiza al titular de derechos de explotación (aparentemente no al mero usuario) de un programa del que no sea autor la posibilidad de modificarlo o de que otros lo hagan con su autorización. Este apartado supone un aparente obstáculo a algunos requerimientos del software libre, lo despejaremos después. Aunque no parece que un usuario realice con el mero uso una explotación del programa, la LPI no parece excluirlo, tal vez a causa de una redacción defectuosa del precepto o por falta de sistemática. Nuestra conclusión provisional podría ser: La LPI garantiza a cualquier usuario o explotador legítimos hacer este tipo de transformaciones en el código. Pero desde luego, necesitará para ello el código fuente. Y la interpretación estándar no es ésta, sino la siguiente: Esta regla se refiere sólo a los programas realizados por encargo, o sólo a los titulares no autores de derechos de explotación distintos del mero uso.



- Tampoco puede el autor de un programa oponerse a su reproducción y transformación si se dan todas las condiciones siguientes:
  1. Que sea indispensable para obtener la información necesaria para la interoperabilidad con otros programas de un programa creado de forma independiente;
  2. Que la reproducción o transformación la haga el usuario legítimo o persona facultada para utilizar una copia del programa, o persona autorizada en su nombre;
  3. Que la información necesaria para conseguir la interoperabilidad no haya sido puesta a disposición, fácil y rápidamente, de las personas citadas antes;
  4. Que la reproducción o transformación se limiten a las partes del programa original necesarias para la interoperabilidad;
  5. Que el resultado se utilice sólo para la interoperabilidad del programa creado de forma independiente; sólo se comunique a terceros si es necesario para la interoperabilidad; y no se utilice para el desarrollo, producción y comercialización de un programa sustancialmente distinto en su expresión o para cualquier otro acto que infrinja los derechos de autor.

Estas reglas sobre la interoperabilidad (excepción o límite a la prohibición de reproducir y transformar un programa sin autorización del autor) no pueden interpretarse de modo que se perjudique injustificadamente los legítimos intereses del titular de los derechos de autor, o se contraría la explotación normal del programa. Está claro además que esta garantía para el usuario, muy limitada y borrosa, no es efectiva si no se proporciona el código fuente, al menos de las partes del programa necesarias para la interoperabilidad.

Una interpretación restrictiva acerca de los derechos sobre los interfaces es la siguiente: La especificación del interfaz está protegida, pero no lo están los protocolos en que se base y que sean necesarios para escribir código que cumpla las especificaciones. Tales protocolos no pueden ser obra protegida.

Hasta aquí llega la protección que garantiza la LPI a los usuarios de programas de ordenador. No olvidemos que esta ley, como todas las de su clase en los demás países, y lo mismo en los tratados internacionales, está pensada para proteger los derechos del autor. De un autor preconcebido, idealizado en el literato, en el pintor, indefensos frente a los editores, los galeristas. Un autor de aspecto distinto al del titular de derechos de explotación de un programa de ordenador, p. ej. un gran fabricante de software cerrado. Sea como sea, la LPI reconoce y protege sobre todo los derechos del autor de un programa, no los del usuario del programa.

Ahora estamos listos para acometer nuestra tercera tarea: Seguir aclarando la terminología, dar algunas definiciones más, desentrañar el contenido estándar de una licencia de software y, digamos en general, perder un poco el respeto a las más abtrusas discusiones jurídicas e incluso poder participar en ellas. Para esto, disponer de un lenguaje preciso es esencial.

### 3. Cuestiones generales y terminológicas sobre las licencias de software

Inevitablemente y antes de nada, hemos de ponernos de acuerdo sobre las definiciones, sobre el significado de los conceptos que vamos a emplear, que confrontaremos con los de la ley española para obtener conclusiones congruentes. Añadiremos más definiciones, que no necesitaremos hasta entonces, en el apartado 4.1.

#### 3.1. Qué es una licencia de software

Es un tipo de contrato de software, de software ya creado. Recae sobre los derechos de propiedad intelectual. En este artículo no se tratan los contratos de software por crear, como el contrato de desarrollo de programas por encargo, servicios de adaptación de software; ni contratos como “escrow”, etc. Pero ojo: sí trata de los derechos de propiedad intelectual originados por motivo de esos contratos, o de cualquier otro por el que se genere software original.

Externamente una licencia puede adoptar muchas formas, desde un documento en papel hasta un archivo electrónico de texto, parte de un ejecutable, etc. Puede ser un acto jurídico independiente o puede integrarse documentalmentemente en el seno de otro contrato, aunque la LPI exige documentos independientes. La licencia puede recaer sobre software también muy variado, aunque la LPI hable sólo de programas. En esencia es una oferta de acuerdo realizada por el autor o titular del programa, que si es aceptada por un usuario o explotador del software, pasa a convertirse en contrato entre las partes. Aquí hay varios conceptos involucrados, necesitamos desmenuzarlos.

##### 3.1.1. Qué es una licencia

En realidad lo que llamamos licencia pasa por varios estadios: Primero es una declaración unilateral del autor del programa en la que expresa las condiciones en que se puede acceder a él y explotarlo. Como tal declaración prácticamente no tiene ningún valor legal, sólo lo adquiere (se dice que pasa a ser ley entre las partes) cuando otra persona acepta sus términos. Como es lógico, cuando la licencia se rechaza, o no se acepta, simplemente no llega a tener efecto.

Es preciso recalcarlo: Aunque la licencia es unilateral, pues la origina el autor voluntariamente y en los términos que le interesen, está pensada para ser aceptada o rechazada por otros, normalmente quienes van a usar el programa o van a explotarlo de algún modo. Si la contraparte rechaza la licencia no hay más que hablar: probablemente el autor del programa no cederá su software, p. ej. no permitiendo su instalación. Pero si la licencia es aceptada, entonces deja de ser una declaración unilateral y se convierte en un negocio bilateral, entre licenciante (el autor o el titular de los derechos de autor) y licenciatarario (quien va a usar o explotar el software).

A este negocio puede llamársele “licencia” o “licencia contractual” o también “acuerdo de licencia”. Su denominación técnica precisa es, para la mayoría de los casos, la de “contrato de cesión de derechos de uso y/o explotación del programa de ordenador”.

Es normal que el documento de licencia contenga otras cuestiones, como garantía, servicios de soporte y postventa, que no tienen nada que ver con la propiedad intelectual ni con el software

libre, y no los tratamos en este artículo. Tal vez haya ocasión en versiones sucesivas de tratar alguno de estos asuntos.

### 3.1.2. Qué es “software”

La LPI no habla nunca de software desde luego, sino de “programas de ordenador”, que define (art. 96) como 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.

No es una bella definición, ni tampoco un modelo de precisión. Dice la LPI que gozan de la misma protección que los programas tanto la documentación preparatoria como la documentación técnica y los manuales de uso. Ya sabemos además que se protegen las versiones sucesivas y los programas derivados, pero no los creados con el fin de ocasionar efectos nocivos a un sistema informático. Tampoco están protegidas las ideas y principios en que se base cualquier elemento de un programa, incluidos los que sirven de fundamento a los interfaces. Esta exclusión parece referirse a los algoritmos y otros elementos, que no necesitamos determinar completamente para saber a qué nos referimos con el término legal genérico “programa de ordenador”. En la práctica, el problema de la definición se plantea ante casos como los sistemas expertos, los interfaces, etc.

Los programas no pueden patentarse, pero sí formar parte de un objeto patentado. Entonces, la protección de la Ley de Patentes también se activa a favor del programa, aunque sea indirectamente. [Nota sobre patentes: Recuérdese lo dicho en el apartado 1, en Derecho español son patentables las invenciones nuevas de aplicación industrial, pero los programas de ordenador no se consideran invenciones, y por lo tanto son no patentables]. Asimismo un programa puede incorporar una marca comercial, sea su mismo título u otra marca. La marca comercial del programa no es objeto de protección por la LPI pero sí por la Ley de Marcas, lo mismo que antes.

Por supuesto podemos considerar incluidos en la definición legal de programa todo aquello que técnicamente lo es: ejecutables de cualquier tipo, módulos, controladores, aplicaciones de usuario y sistemas operativos, suites, paquetes y distribuciones, con toda la documentación. La GPL concretamente se aplica a programas y a “cualquier otro tipo de trabajo”. No olvidemos que la LPI exige que la secuencia de instrucciones sea original, obra del intelecto, y se destine a un sistema informático. La Directiva 1991/250, traspuesta a la ley española en 1993, incluye los programas “incorporados al hardware”. En España, por otra parte, la topografía de semiconductores está protegida en una ley propia de 1988; véase el art. 104 LPI.

Son programas protegidos tanto los originales como los derivados, las versiones sucesivas y las originadas en bifurcaciones. Tenemos obras independientes, como un kernel; y obras compuestas, como paquetes y distribuciones. Tenemos obras originales como el primer núcleo Linux, y obras derivadas como un kernel 2.2.x.

Sin embargo, puede que no encontremos software en dominio público por expiración del plazo de duración de los derechos, pues no han transcurrido años suficientes desde la aparición de los primeros programas a mediados del siglo XX. Es cierto que, además, este software sólo tiene utilidad histórica.

¿Puede un autor poner su software en dominio público voluntariamente? No en Derecho es-

pañol, para el cual una obra está en dominio público sólo cuando se extinguen todos los derechos de explotación por transcurso del plazo de duración. No es exactamente lo mismo que carecer de copyright, como lo definen la FSF y la OSI en su digamos “plataforma jurídica anglosajona”. Volveremos sobre todo esto más adelante en 4.1.

### 3.1.3. **Quién es el autor del software**

Aquí no vamos a extendernos, porque esto ha debido quedar claro desde el apartado 1. Recordemos los conceptos de autor asalariado, y de obra colectiva frente a obra en colaboración. Un programa como Windows XP es obra colectiva, creada por asalariados de Microsoft, que asume la autoría del programa y sin que ninguno de los desarrolladores puedan reclamar la explotación separada sobre su parte. La distribución Debian GNU/Linux es una obra en colaboración en cuanto a los componentes individuales, pero la distribución en sí es una obra colectiva, compuesta, cuyo titular es una asociación de desarrolladores voluntarios que se sirve de la organización Software In The Public Interest, Inc. para dotarse de personalidad jurídica, titular de los derechos de autor de la distribución en sus diferentes versiones [Esta explicación es conforme con el Derecho español, y en realidad vale para todo el mundo. Se incluye aquí a título de ejemplo, esperemos que apropiado].

Una observación común en la literatura jurídica acerca de lo ventajosa que resulta la protección del programa no libre para una empresa de software por las reglas del derecho de autor, se basa en que normalmente los autores de los programas son anónimos y quien se beneficia de los derechos de autor (la empresa) no es autor. Pero lo cierto es que, primero, la observación también rige para los autores de obras literarias salvo excepciones; segundo, la observación no se aplica al software libre, cuyos autores no son casi nunca anónimos; y, tercero, la mayor parte de los derechos de explotación -y por tanto su protección legal- queda cedida a la comunidad de usuarios y por tanto las vías para obtener beneficio no derivan ya de la exclusividad.

### 3.1.4. **Quién es “usuario legítimo” del software**

Este es un concepto mucho más importante, aunque la LPI no lo define. Podemos suponer, en un examen superficial de las premisas del apartado 2.1, o simplemente deduciéndolo de nuestra experiencia cotidiana, que usuario legítimo es quien ha comprado el software, y en efecto así es cuando la compra del soporte incorpora -mediante la licencia- la autorización para usarlo.

Pero hablando estrictamente, no existe la “compraventa de software”. Lo que uno compra en la tienda (tal vez un CDROM con un juego o una distribución GNU) no es el programa, sino sólo su soporte más una oferta de licencia para uso y explotación -licencia que después habrá de aceptar. Y esto es todo (y nada menos, dirán el vendedor, el distribuidor, el titular de derechos de explotación y/o el autor del programa). Esta explicación suele encontrarse en las licencias de software propietario, que en este artículo y por las razones que se explicarán después, preferimos denominar “software cerrado”. No es una explicación realmente necesaria, pues todo software nace con copyright.

Pero también puede adquirirse software por ftp anónimo gratuitamente con licencia copyleft,

y por supuesto quien lo obtiene así puede usarlo muy legítimamente; simplemente la LPI no estaba pensando en esta circunstancia. Cuando se aprobó la LPI en 1987, incluso cuando se aprobó en Bruselas en 1991 la directiva que obligó a hacer algunas modificaciones en la ley española en 1993, el software libre no era un movimiento lo suficientemente relevante en Europa y menos en España.

Lo relevante para nosotros ahora es otra cosa: No hay otro tipo de obras para las que la LPI distinga entre usuarios legítimos e ilegítimos, sólo hace la distinción para los programas de ordenador y para las bases de datos. No se habla nunca de usuario ilegítimo de un libro, o espectador ilegítimo de un cuadro. Esto parece absurdo, y puede que lo sea en cierto sentido que vamos a explicar. Ante todo, no estamos hablando de quien roba un CD que contiene un programa, o roba un libro, o entra en un museo sin pagar. Estamos hablando de quien usa un programa que instaló desde un CD prestado por un amigo, de quien ha leído un libro prestado por un amigo, de quien contempla en casa un cuadro prestado por un amigo. Es evidente que en los dos últimos casos no hablamos de “lector ilegítimo” ni de “espectador ilegítimo”, pero para la LPI el primer fulano, el del programa de ordenador prestado, ése es un usuario ilegítimo. Esta es una extraña asimetría. Nos dará que hablar después.

### **3.1.5. Quién es el responsable ante el usuario**

El autor tiene siempre algunas obligaciones frente a quienes explotan su obra. Ante todo responde de la autoría y de la originalidad de la obra. Responde también de su propia capacidad jurídica para licenciar el programa. El supuesto que más problemas puede dar es el de un redistribuidor de software que él cree libre y que en realidad no lo es, por error, a sabiendas o mediante engaño (y esto tendrá que probarlo).

Ocurre que las exigencias jurídicas del software libre (sl) pueden confundir: Redistribuir software libre del que no se es autor no traslada automáticamente al redistribuidor las responsabilidades del autor. Para empezar, la habitual cláusula de ausencia de garantía deja claras ya algunas cosas. En general, si el licenciante del programa original (normalmente el autor pero no necesariamente, p. ej. en el caso de los asalariados) y el licenciatarario (quien tal vez lo va a modificar y redistribuir) acuerdan válidamente los términos de la licencia, está claro que de la autoría de los programas responde cada cual: del original el autor o licenciante; y del derivado el licenciatarario, pero por motivo de una segunda licencia, en la que es él ahora el licenciante de un tercero licenciatarario, y así sucesivamente. Más en general no es posible, ni seguramente útil, tratar los diferentes supuestos de responsabilidad (patrimonial o no, ya sea civil, penal o administrativa). Es un asunto demasiado amplio, árido y complejo. Y a efectos prácticos no muy útil. Tal vez sea éste, como otros del presente artículo, objeto apropiado para una apartado FAQ en versiones sucesivas. De todos modos, uno debería ser capaz de deducir la respuesta a su duda a partir de cuanto contienen la presente “Introducción” (ésa es su finalidad).

### 3.2. Contenido deseable de una licencia de software

Lo que sigue pretende indicar al lector, que probablemente no será ducho en cuestiones jurídicas, en qué debería fijarse al leer una licencia para entenderla correctamente y sin mucho trabajo. El método no va a ser la presentación de un prototipo de licencia abstracta, sino el examen de la mejor licencia concreta que hemos podido encontrar, ya sirva a un programador tal cual, o como modelo para obtener otra a su gusto, o tal vez de anticristo para estigmatizarla a placer. Se trata de la GPL.

La GNU-GPL es una pieza jurídica de gran valor. Entre otras utilidades, contiene la estructura completa del sistema de cesión de derechos de autor sin atentados al copyright y respetuosa con los derechos y libertades de los usuarios. Es superior técnicamente a los mejores ejemplos de licencias de software no libre, es más completa que las licencias breves tipo BSD, y mucho más clara y fácil de leer que cualquier otra de software no libre que conozcamos.

Para empezar, la GNU-GPL carece de traducciones oficiales. Pero esto no es ningún problema práctico, primero porque hay traducciones oficiosas; segundo, porque el inglés original es fácilmente traducible a términos jurídicos de cualquier país; tercero, porque el texto evita deliberadamente los tecnicismos y expresiones o rodeos oscuros. Sin ser coloquial, que casi lo es, pasa por ser un modelo de redacción jurídica.

Estas cualidades no se deben sólo a su punto de vista distinto al de la propiedad intelectual. Por cierto, la GPL no se opone a la propiedad intelectual, pero su enfoque no es desde luego el de la protección de los derechos de autor -para eso ya está la LPI- sino el del respeto de la libertad de los usuarios. Para redactar una licencia que acabaría por dar nombre a una forma de distribución (“copyleft”) la FSF debió sortear más de un serio escollo, además de enfrentarse con críticas no siempre benévolas, con el punto de mira desviado y finalmente incapaces de demoler el imponente edificio que se estaba levantado bajo su protección. Por ahora no disponemos de un modelo mejor, aunque todo es perfectible. Las directrices Open Source son muy prácticas, pero técnicamente hablando no son un modelo de licencia, y en términos jurídicos significan un paso atrás sobre el esquema de la FSF, como se tratará de demostrar después. No son tampoco fáciles de entender. Pero su importancia e influencia son enormes y le dedicaremos el apartado 4.5.2. Iremos dando indicaciones por orden, para desmenuzar la licencia deseable, aunque no llegaremos a los detalles.

La forma más segura de licenciar el programa consiste en incluir un anuncio al principio de cada fichero fuente, unas líneas de indicación de autoría y año de publicación (es decir, lo que se llama “línea de copyright”) y la indicación de uno o dos lugares fácilmente accesibles donde encontrar el texto completo de la licencia.

Una licencia no necesita un preámbulo que exponga la justificación de las cláusulas o cuerpo de la licencia, pero la GPL tiene uno, y muy útil porque sirve para solventar las dudas que pueden aparecer al leer o al aplicar las cláusulas. La GPL no es neutral, pretende ser interpretada en un sentido dado y no en otro distinto u opuesto. Su sentido es el de la libertad, y está recogido justamente en el preámbulo, que forma parte de la licencia misma, aunque esto la FSF no se ha ocupado de indicarlo así. De todos modos, la licencia es toda ella autoexplicativa, e interpretarla debería de resultar fácil. No puede decirse lo mismo de muchas otras licencias estándar que hemos

podido consultar.

En el cuerpo de una licencia de software debe encontrarse 3 grupos de cláusulas, sólo en lo que se refiere a los derechos de autor; habrá más apartados si se tratan asuntos sobre garantías, servicios de apoyo, pagos y demás, pero las materias ajenas a la propiedad intelectual no son tratadas en estas notas. Los grupos de cláusulas son los siguientes, y se incluye después de cada uno, como ejemplo, las cláusulas correspondientes de la GPL. La discusión del grupo segundo, el cuerpo principal sobre explotación del programa, la dejamos para el apartado 4.

1. Cláusulas generales
  - a) Definiciones y ámbito de aplicación de la licencia. Advertencias de copyright (GPL cláusula 0)
  - b) Formas de aceptación de la licencia (GPL cláusula 5)
2. Uso y explotación del programa
  - a) Copia, modificación y distribución libres (GPL cláusulas 1 a 3)
  - b) Copyleft, o persistencia de la libre distribución de programas derivados (GPL cláusulas 4, 6 y 10)
  - c) Integridad del sistema copyleft en caso de impedimento forzoso a la libre distribución (GPL Cláusula 7)
  - d) Posibilidad de límites geográficos a la libre distribución (GPL cláusula 8)
3. Intangibilidad de la licencia. Versiones sucesivas (GPL cláusula 9)

Las diferencias entre el orden lógico y el de presentación por la GPL se deben a necesidades prácticas de exposición de la FSF. Comprobaremos que para analizar el funcionamiento de una licencia de software libre es más apropiada la ordenación lógica. A continuación tratamos las cláusulas generales, y como se ha dicho dejamos las relativas a la explotación para el apartado 4.

### 3.2.1. Cláusulas generales

**Definiciones, ámbito de aplicación de una licencia y avisos de copyright.-** Estas declaraciones de la licencia no tratan directamente de la explotación del software, e incluso pueden ser teóricamente innecesarias, pero siempre ayudan a la comprensión del cuerpo principal. En cuanto a las definiciones, podemos usar las contenidas en las leyes o bien habremos de hacerlo nosotros mismos. Son esenciales las de programa u objeto licenciado, programa derivado (el obtenido a partir del que ahora licenciamos) y las formas de explotación. De todo ello se encuentra información en los apartados anteriores. Es típico de las licencias, como de muchos otros contratos, fijar los términos importantes que vayan a usarse más a menudo: “usted” puede ser el licenciatarario, “titular del copyright” o simplemente “titular” es el autor o el derecho-habiente de las facultades de explotación que van a autorizarse, “versiones y/o programas derivados” son el resultado de

cualquier modificación del programa, incluida la traducción, etc. Todo esto depende de las concretas necesidades en cada caso.

Una licencia debe delimitar claramente su ámbito geográfico de aplicación, su duración (que puede ser indefinida) y las formas de explotación que se van a tratar, las que se retienen y las que se ceden. Aquí bastará limitarse a definir las lo mejor posible y siempre que parezca conveniente o necesario. Hacen referencia a cuestiones generales tratadas en otras partes de este artículo, así que no las repetimos. Es clásico advertir que la licencia no se aplica a la entrada o a la salida del programa, salvo que se diga otra cosa, es decir: siempre que una y otra no sean a su vez obra protegida.

Es casi esencial que el programa incluya de algún modo uno o más avisos de copyright y de la licencia, como ya hemos reseñado antes.

La copia impresa debe prevalecer sobre la información que muestre la pantalla, porque es más sencillo hacer modificaciones de última hora en aquélla.

**Formas de aceptación de la licencia.-** Este asunto es clave, al menos formalmente, pero no debe dar problemas en su puesta en práctica. No hay auténtica licencia hasta su aceptación por el destinatario, esto ya lo sabemos. La forma de la aceptación es variada, las hay muy rebuscadas, incluso puede encontrarse algunas definitivamente abusivas para el usuario (véase el apartado 6.1). Aquí nos referiremos sólo a las habituales. En esencia, se trata de que queden claras las voluntades del licenciante y del licenciatario, por cualquier medio admitido.

Primero, es conveniente advertir al destinatario que no está obligado a aceptar la licencia para el uso y la copia privada, pero sí para la modificación y distribución (o redistribución) del programa. Los dos primeros actos son privados, normalmente; pero los segundos involucran a terceras personas. A primera vista, de lo dicho se podría deducir que este software no va a tener “usuario legítimo” en el sentido genuino de la LPI, ni estará prohibida la copia privada, también en contra de la LPI. Pero no es así. El uso y la copia privada son actos que normalmente sólo conocen los mismos usuario y copista, p. ej. si se realizan en casa. Por tanto no tiene mucho sentido exigir la aceptación de la licencia para estas dos formas de explotación. Esto es así sobre todo para el software libre, en donde por definición prácticamente todo usuario es legítimo y quedan autorizadas las formas principales de explotación. Obviamente, para el software cerrado la situación es muy distinta, ya que su explotación está radicalmente restringida desde el mismo uso. De hecho, desde antes del uso, pues para algunos fabricantes el romper los precintos del paquete de CDs supone la aceptación de la licencia (puede comprobarse en las licencias de conocidas casas comerciales).

Suele darse por válido que la realización de actos de explotación permitidos por la licencia suponen su aceptación. Por supuesto, pulsar “aceptar” en el ejecutable interactivo tiene exactamente -jurídicamente- ese valor, aunque debería darse la oportunidad al usuario de poder usar el programa durante un tiempo para comprobaciones y ajustes, antes de la aceptación. En fin, que no hace falta una declaración personal por escrito, firmada y fechada, para aceptar una licencia. Sí tal vez para rechazarla, si uno cree que ha realizado, por error o defecto del programa, algo que puede significar la aceptación de lo inaceptable. El software libre, de todos modos, no se enfrenta con estos problemas casi nunca.



### 3.2.2. Intangibilidad de la licencia

El contenido de una licencia, sobre todo de una licencia de software libre, no es libre. Dicho de otro modo, el efecto de una licencia de sí no es reflexivo, no se aplica a sí misma. Una licencia no puede permitir su propia modificación entretanto esté en vigor, salvo por acuerdo expreso de ambas partes. En Derecho español se dice, más en general, que los términos de un contrato no pueden quedar al arbitrio de uno solo de los contratantes. Por todo esto se exige que la licencia sea intangible, intocable mientras esté en vigor.

Tenemos que distinguir las novaciones, o cambios que puede sufrir una licencia por acuerdo entre las partes o por sentencia judicial, p. ej. si un tribunal anula una cláusula abusiva; de las revisiones de una licencia-modelo o general, como p. ej. la GNU-GPL. El supuesto interesante es el segundo. Una licencia-tipo, como la GPL o la FDL, que al fin y al cabo son obras literarias, están protegidas por las leyes de derechos de autor, aquí aplicados estrictamente con la finalidad de mantener el texto sin cambios. Estas licencias están sujetas al copyright, en este caso de la FSF, con domicilio en Boston-MA. Esto significa que quien use la GPL para licenciar su programa y mantenga el nombre de la licencia en el ejemplar que utilice para su programa, debe mantenerla íntegra y sin modificaciones. En otro caso, y si prefiere el autor realizar algún cambio, no será ya “licencia GPL” y no podrá utilizar tal denominación. A esto se refiere la GPL en la advertencia © que va antes del preámbulo.

La intangibilidad de las licencias-tipo se debe a su papel de destinataria de tantas remisiones que circulan por ahí. La seguridad del tráfico exige que los términos literales no cambien. Aún así, una licencia modelo puede pasar revisiones, y por tanto podemos encontrarnos con versiones distintas, pero no deberían serlo mucho sino sólo en mejoras, aclaraciones y tratamiento de casos nuevos. Todas las versiones deben ajustarse al espíritu de la licencia original, en otro caso habrá de redactarse una licencia distinta y con otro nombre o identificador. La FSF admite además que si alguien licencia su programa con la versión x puede hacerlo al mismo tiempo con referencia a cualquier versión posterior. Esto se contrapesa estableciendo que cuando no se especifique el número de versión de la licencia, el destinatario elegirá la que más le convenga, algo perfectamente válido y respetuoso con el usuario.

**¿Puede revocarse una licencia?.-** Una licencia es revocable por el autor en muchas circunstancias y en ejercicio de varias facultades. Una de ellas, el caso del llamado derecho de arrepentimiento, por cambio de convicciones del autor, forma parte del derecho moral. Por supuesto tiene un límite: indemnizar los perjuicios que pueda producir a terceros. Y lo mismo ocurre con cualquier revocación unilateral (no pactada) de la licencia. Si hay acuerdos sobre este asunto, habrá que estar a lo acordado. En el sí la situación no tiene mayor relevancia, salvo en un caso: la revocación de la licencia copyleft ¿Es ello posible? Y de serlo, ¿qué consecuencias tiene para el autor? ¿Y para los licenciarios antiguos y nuevos? En el software de uso masivo, sea libre o no, las dificultades en la aplicación práctica de las reglas sobre revocación de licencias son tan grandes que se usa otra fórmula: nueva licencia para una nueva versión del programa; o bien la doble licencia. Los problemas resueltos de esta forma resultan más manejables, pero no dejan de ser serios. Su tratamiento aquí excede del ámbito de las presentes notas, aunque se espera poder

tratarlo mínimamente en versiones sucesivas.

### 3.3. Tipos de licencias de software

Aunque la GNU-GPL, BSD, XFree86, Mozilla, y otras muchas son las licencias más conocidas, lo cierto es que no podríamos enumerarlas todas porque cada autor puede tener la suya, y una distinta para cada programa. Pero esto no es ningún problema, por varias razones. Primera, porque disponemos de un instrumento de análisis de las licencias: las premisas del apartado 2.1 de este artículo. Segunda, porque hay varias licencias que sirven como modelos para otros programas, y así hablamos de “tipo BSD” para referirnos a licencias similares a la del sistema operativo de Berkeley. Hay incluso licencias y modelos creados en abstracto, sin referencia a un concreto programa, como la misma GNU-GPL o la Open Source Definition. Las licencias típicas son muy cómodas de usar ya que el autor de un programa dado, conector de la que más le conviene, la incluye tal cual o hace sólo las modificaciones que precisa, sin el trabajo de redactar una por entero, o encargarla a un abogado. Por otro lado, quienes van a usar o explotar un programa (los licenciarios) también conocen estas licencias típicas y saben de antemano a qué atenerse. Igualmente, los abogados y los jueces tienen mejor conocimiento de ellas que de una licencia nueva u original. Todo esto simplifica las relaciones y los negocios y resulta útil.

Nosotros vamos a ocuparnos sólo de las licencias típicas. Las más interesantes son las licencias de software libre, ya que en una licencia se materializa la voluntad del autor sobre cómo desea que su programa se use y explote, y las de software libre materializan una voluntad radicalmente contraria a la que la LPI espera de un autor. En particular, las licencias copyleft revierten literalmente las relaciones autor-usuario que la LPI presupone.

Por contra, las licencias de software no libre, exactamente las de software cerrado, se asientan en la LPI y desde ella pueden incluso lanzarse más allá en la limitación de los derechos y libertades de los usuarios. Pero esto sólo es válido, como ya sabemos, si no se atenta contra las normas imperativas, que conocemos del apartado 2.1.3. Por eso se dice que las licencias de software no libre están acompañadas con la LPI y son por lo tanto mucho menos interesantes. La lectura completa de una de estas licencias no libres puede resultar además una penosa experiencia.

Con todo, sólo las licencias típicas forman ya un buen montón. Tampoco tiene mucha utilidad hacer una selección, pues la FSF y la OSI ya han hecho algunas muy valiosas, aunque no siempre detalladas, y de las que en el apéndice C se encuentran las referencias. Nosotros vamos a examinarlas de una forma distinta, y nos evitaremos tanto el tedio de la exposición de licencias una por una, como una lectura árida o abrumadora. De todos modos, se encontrará en 4.6 una breve discusión final sobre los criterios de las clasificaciones principales.

Y llegamos por fin al núcleo de la cuestión.

## 4. Las licencias de software libre

### 4.1. Más definiciones

Necesitamos aún algunas definiciones más para seguir aclarando la terminología que estamos usando y otra nueva que introduciremos enseguida. Las definiciones más útiles para hablar de software libre son, de nuevo por su rigor jurídico, las de la FSF, que usamos en este artículo por convención.

Tenemos dos grandes superconjuntos: el software libre y el resto del software, que por tanto llamamos no libre. Cualquier otra terminología para estos superconjuntos no es aceptable en español. Por cierto, software no libre no es sinónimo de “software propietario”, y como esta última expresión es horrible además de inexacta, nosotros no la utilizaremos, además en el fondo nos sobra. En caso necesario hablaremos de “software cerrado”. Es más preciso hablar en general de software no libre, y de paso englobamos a los semipropietarios, semilibres, sharewares y demás. Esta terminología también nos facilitará la comprensión cabal de un cuadro bastante grande de licencias.

“Software libre” (sl) es el que incorpora una autorización general no discriminatoria para usar, copiar, modificar y distribuir el programa original o sus derivados, gratuitamente o no. Debe proporcionarse las fuentes, directa o indirectamente, pero siempre de forma fácil y asequible. Todo programa que no incorpore esta autorización no es libre, decimos que es software no libre.

Abusaremos un poco del lenguaje llamando “licencia libre” a la licencia de un programa libre.

Estas definiciones no son pacíficas. Nosotros las usaremos convencionalmente, pero ignorar cuánto hay detrás es un pobre servicio al conocimiento, porque no se trata de pequeñeces. Para empezar, las definiciones de software libre (“free software”) y fuente abierta (“open source”) no son coincidentes, aunque vienen ciertamente a significar casi lo mismo. Pero estas diferencias no son importantes por ahora, las dejamos para el apartado 4.5.2.

También es esencial distinguir sl (superconjunto) de copyleft (subconjunto). Éste es software libre cuyos términos de distribución no permiten a los re-distribuidores añadir a su licencia restricciones adicionales a las de la licencia de que se sirvieron. Esto supone la perpetuación de la condición de libertad del software hasta su extinción. El copyleft determina la imposibilidad (jurídica) de apropiarse del software libre. Y éste es el hallazgo de la FSF, al que dedicaremos por entero el apartado 4.5.1. El resto del sl que no es copyleft puede ser modificado añadiendo restricciones a la libre distribución que no se encontraban en la licencia del programa originario. Los ejemplos característicos son las licencias BSD y X11.

Esta segunda dicotomía “sl-copyleft v. sl-no-copyleft” es en cambio pacífica, porque los términos de distribución de, p ej., la GNU-GPL son claros y terminantes. Es cierto que una cuestión de estrategia invitó a la FSF a redactar la Lesser GPL, y que se han detectado una o dos lagunas relativamente importantes, pero procedentes sólo de una interpretación forzada del sentido de su texto. Un programa es copyleft o no lo es, y esto es fácil distinguirlo. Aun así, la FSF habla de “grados” de copyleft (hay programas más copyleft que otros), y finalmente introduce una última subdivisión: “copyleft compatible GNU v. copyleft no compatible GNU”. Lo mismo hace la OSI con su calificación de compatibilidad Open Source. Pero estas expresiones, graduación y com-

patibilidad, forman otro de los asuntos en los que, dado el ámbito de este artículo, no podremos entrar a fondo hasta una versión ulterior.

En Derecho anglosajón se incluye dentro del *si* el software que se encuentra en dominio público, pero esto generalmente no es así en los Derechos continentales. En primer lugar, no es fácil con arreglo a la ley española encontrar programas en dominio público por las razones expuestas antes, simplemente no ha transcurrido suficiente tiempo desde la aparición de los primeros programas protegidos para que se haya producido la extinción de los derechos de autor sobre ellos (recuérdese: toda la vida del autor y 70 años más). Segundo, probablemente las fuentes pueden no estar disponibles. Tercero, el Derecho español admite la apropiación del dominio público inédito (art. 129.1 LPI), y desde luego la apropiación de las obras derivadas del dominio público, pero no del dominio público mismo. Con razón la FSF considera que el software en dominio público no es en modo alguno copyleft, sobre todo en Derecho anglosajón. Ni siquiera tiene que ser necesariamente libre, p. ej. no lo es si las fuentes no están disponibles. Volveremos aún sobre el dominio público en 4.5.1.

#### **4.2. Encaje general de las licencias de software libre en la ley española**

Sin rodeos, no plantean problemas en Derecho español, son perfectamente válidas y viables. Trataremos de demostrarlo en lo que resta de este apartado 4. En primer lugar:

1. No afectan a los derechos morales del autor, aunque la existencia del llamado derecho de arrepentimiento, típico de los Derechos continentales, parece sugerir otra cosa. Pero ya tratamos este asunto (superficialmente, es cierto) en el apartado 3.2.2 al hablar de la revocación de las licencias. Aunque allí quedaron temas por tratar, éste en particular quedó allanado. Se trata de un asunto más teórico que otra cosa, de todos modos. Y aún volveremos de nuevo a él cuando tratemos de la Open Source Definition.
2. En cuanto a los derechos patrimoniales del autor, tampoco hay nada en las licencias de software libre que infrinja las normas imperativas de la LPI. En efecto:
  - a) El derecho de autorizar o prohibir la explotación de la obra se manifiesta justamente en la facultad del autor de dar licencia a su programa, siempre que no se vulneren otras normas imperativas u obligaciones asumidas.
  - b) Las licencias de software libre no implican renuncia del derecho de remuneración, aunque en muchos casos se renuncie de hecho a una remuneración, que es cosa distinta, perfectamente renunciable. Es claro que el autor no renuncia a la explotación por sí mismo. Y respecto de la explotación por los demás, la cede con causa (liberalidad, prestigio, obtención de una marca comercial, de apoyo en el mantenimiento del programa, etc, etc).
  - c) Las licencias regulan sobre todo la cesión de los derechos de explotación, que es su cometido, y por tanto son el vehículo apropiado para contener las condiciones de uso y explotación de un programa de ordenador.

Una dificultad más seria se encuentra en la limitación que la ley española impone a cualquier cesión de derechos de explotación: Queda limitada siempre a los medios de explotación existentes en el momento de la cesión, esto es, en el momento de la aceptación de la licencia; y no se extiende por tanto a los medios futuros de copia, modificación, etc, ni a los inexistentes. Por su parte la GPL deja bien claro que la explotación libre autorizada se refiere “a cualquier medio” (cláusula 1) y queda restringida a determinadas formas de explotación (y no a otras), concretamente la copia, modificación y distribución. Si se produce una explotación de distinto tipo, la GPL deja de amparar al licenciataria (cl. 1 y 4), pero el sentido de la GPL es el de respetar las libertades del usuario, no la restricción injustificada del uso y explotación de los programas; y no entra, salvo lo ya señalado, en buscar limitaciones más allá de las necesarias a su finalidad. Por lo tanto se concluye que la GPL ampara formas de explotación sobrevenidas después de la aceptación de la licencia, pero siempre que no se atente contra el espíritu de la GPL. Aunque nunca está de más avisar al licenciante, puede incluso ser imprescindible.

También hemos de resolver una aparente contradicción entre la regla imperativa de la LPI que dice “las cesiones no exclusivas son intransmisibles” (art. 50.1) y el hecho de que una licencia de sl, que consiste en una cesión no exclusiva, justamente permite la transmisión ulterior de derechos por el licenciataria si éste crea un programa derivado. Pero no hay tal contradicción. Por al menos dos razones: 1ª La esencia del sl está en la cesión de derechos de explotación sin exclusiva, y no en ninguna renuncia del copyright. Mediante la licencia libre el autor cede sus derechos de explotación sin exclusiva, pero ello no permite al licenciataria re-licenciar la obra originaria, ni licenciar su obra derivada en términos contrarios a los aceptados en la primera licencia. 2ª El art. 50.1 no es realmente una norma imperativa, existe para proteger al autor, pero éste puede disponer de ella, es en realidad una norma dispositiva. Se incluyó en su momento como imperativa por pura precaución, pues el tenor literal de la LPI da que pensar. Para los más juristas: una licencia de sl tiene algo de donación modal (te doy algo si haces esto), y esta es otra forma de demostrar que la contradicción es sólo aparente.

A continuación veremos con más detalle las libertades aparejadas a una licencia de software libre. Seguiremos el orden expuesto al final del apartado 3.2, o sea: el segundo grupo de cláusulas que teníamos pendientes. Puede adelantarse que no se trata de un examen pormenorizado de cuantas cuestiones suscitan los rúbulos de los apartados, sino un vistazo general. Esto podemos en cierto modo permitirnoslo porque, primero, estamos tratando con el negativo de las habituales licencias de software cerrado, innecesariamente prolijas y obsesivas. Segundo, estamos tratando acerca de las libertades, más simples de expresar que las sujeciones.

### 4.3. Libertades de uso y reproducción

Estas libertades no nos darán ya mucho trabajo. Todo ha quedado definido en apartados anteriores y sabemos por tanto que una licencia de sl otorga libertad prácticamente plena para utilizar y copiar el programa cuando, como, cuanto y donde a uno le apetezca. Suele incluirse restricciones formales, como el mantenimiento del aviso de copyright, que si son razonables no hacen al programa no libre.

Por supuesto, es lícito bajo licencia de sl cobrar por el acto físico de transferir copias del

programa (p. ej. en CDROM), así lo dice entre otras la GPL en el último inciso de la cláusula 1. ¿Significa esto que la cesión de derechos de explotación del programa (que es una transferencia de objeto inmaterial) ha de ser gratuita si se utiliza la GPL? Aquí la GPL parece confundir el programa (obra intelectual, inmaterial, objeto de los derechos de autor) con el soporte de la obra (un binario o código fuente, grabados en un CDROM). Debería estar claro que la GPL no exige gratuidad en la cesión de derechos de explotación, no hay restricción a los derechos de autor ni a la libertad del usuario, pero es un hecho la cesión gratuita en muchas ocasiones. Además, ¿a quién se cobra por una cesión de derechos de explotación muy amplia y con destinatario indeterminado? Este asunto es muy teórico y no merece más atención en este momento.

#### 4.4. Libertad de modificación

Tampoco aquí encontraremos a estas alturas dificultades mayores, aunque siempre es posible complicarse la vida. Uno puede modificar libremente un programa libre, que lo es porque entre otras cosas se dispone de su código fuente. Puede traducirse, transformarse, combinarse con otros, o dividirse. Todos los programas o colecciones de programas obtenidos son obras derivadas, pero (un gran pero) no necesariamente libres.

¿Qué ocurre si redistribuimos un programa en el que hemos incluido parte del código de un programa libre? Que el programa obtenido ha de ser libre en su totalidad. Es decir, no puede extraerse de un programa libre copias u obras derivadas que a su vez se licencien como libres sólo en la parte derivada. En varias ocasiones la GPL tiene en cuenta este supuesto, especialmente para los trabajos derivados (cláusula 2), exigiendo que la licencia se aplique al programa “como un todo”, de modo que el carácter libre se transfiera al programa derivado. Aunque hay excepciones.

Esta es una exigencia coherente con las bases del sistema copyleft, de hecho es la primera exigencia del copyleft, en el ámbito de la modificación de programas. Es el primer supuesto que nos encontramos en nuestro recorrido con el llamado incorrectamente “virus copyleft”, calificado también de efecto contaminante. Desde el punto de vista de las libertades del usuario es más bien un efecto supermineralizante, reconstituyente. Estos calificativos no tienen mucha importancia, sí el efecto mismo por supuesto. Pero esta restricción no es toda la cláusula copyleft, en realidad no lo es en absoluto en cuanto a las modificaciones que no se redistribuyen. Más bien, la transmisión del carácter libre de un programa original a sus derivados es una exigencia del copyright: Si el programa original es copyleft, porque el derecho del autor del programa derivado se origina en una licencia copyleft, de la que no puede sustraerse. Y si el programa original no es copyleft, exactamente igual.

Luego el mal llamado carácter contaminante del copyleft resulta que se da bajo cualquier licencia, como cualquier jurista esperaría. Las críticas habituales suelen tener lugar en otro plano, sea económico, empresarial o político. Por ejemplo, que la contaminación por licencias no libres es una calamidad para el usuario, que la producida por licencias Open Source no copyleft es incierta, y que la producida por el copyleft es definitivamente una bendición para el usuario y la libre computación en general, porque mantiene la libertad. Nada de esto se deduce del Derecho, que, todo lo más, establece reglas muy generales, como que lo accesorio sigue la suerte de lo principal (art. 379 CC) y que los derechos sobre una mezcla indivisible son proporcionales a los

elementos mezclados (art. 381 CC), siempre que no haya otros pactos.

Puede darse el caso de un programa que posee partes identificables no derivadas de un programa libre. Entonces, y siempre que se trate de trabajos independientes y separados (“autónomos” en nuestra nomenclatura del apartado 1), sólo entonces no se transmite el copyleft. Pero si esas partes se distribuyen como un todo derivado del programa libre, la distribución del todo debe producirse según la licencia libre, cuyas autorizaciones se extienden al todo. La finalidad de esto (véase cl. 2 GPL) no es otra que la de controlar la distribución de los trabajos derivados del programa libre. Además, y esto es esencial en las distribuciones y paquetes, la reunión o colección de trabajos libres y no libres en un volumen de almacenamiento o medio de distribución NO hace que unos trabajos pasen al ámbito licenciado por otros. Esto se ajusta como un guante a las reglas comunes, o sea las del CC que hemos citado. Vamos a ser más explícitos.

#### 4.4.1. Adquisición de propiedad: Unión y especificación de cosas

Como base del debate sobre cómo han de transmitirse los efectos de las licencias de programas originales a los programas derivados, vamos a utilizar las reglas del CC citadas arriba, y algunas más. Se trata de saber cómo se adquiere la propiedad sobre objetos derivados, bien por unión de cosas distintas (mezcla y adjunción), bien por especificación de una cosa. Este es un viejo asunto de los juristas, desde hace más de 2000 años. “Nada más complicado y de más difícil apreciación jurídica en la vida real... Las relaciones que supone... han llegado a [considerarse] de casi imposible determinación...”, así se expresaba un magistrado, J.M<sup>a</sup> MANRESA, hace un siglo aproximadamente. Con esta alentadora perspectiva, vamos a basarnos en las normas del CC aplicables a los bienes muebles, porque de cierto que los programas de ordenador no son inmuebles (!).

**Unión de programas.-** Tenemos dos tipos de unión de cosas, la mezcla y la adjunción. La mezcla de elementos supone que éstos resultan después inseparables, de modo que cada propietario adquiere un derecho proporcional sobre la parte que le corresponde según el valor de las cosas mezcladas (art. 381 CC).

La adjunción ocurre por la unión de cosas heterogéneas que se unen indisolublemente para constituir un solo y nuevo objeto, no desmontable. En este objeto pueden distinguirse tal vez sus antiguos componentes, pero no pueden ya separarse. En este caso, la regla es que lo accesorio sigue la suerte de lo principal (art. 375 CC). Accesorio es lo que facilita el uso o perfecciona lo principal (art. 376 CC). Pero suele utilizarse como criterio más práctico el del valor económico de los componentes (art. 377.1º CC).

**Versiones de programas o especificación.-** La especificación consiste en dar a una cosa una nueva forma. Más estrictamente, es dar nueva forma a una cosa ajena, creándose así una nueva cosa de más valor. El art. 383 CC dice que el especificador hace suya la cosa nueva, si efectivamente es de mayor valor, aunque habrá de indemnizar al dueño de la cosa especificada (“versioneada”). En otro caso, éste puede optar por quedarse con la nueva especie, indemnizando el valor de la obra nueva; o pedir indemnización de la materia original.

Todo esto sólo vale si quien mezcla, adjunta o especifica actúa de buena fe.

#### 4.4.2. Exigencias razonables para la modificabilidad del software

Por supuesto, no hay modificación factible de un programa de ordenador si no se dispone del código fuente. La disponibilidad del código recompilable puede darse de varias formas, pero sólo algunas son admitidas en la definición de software libre. Habrá de escogerse alguna de éstas en tal caso y dependiendo de la explotación que se prevé:

- Acompañar las fuentes completas en formato electrónico y en el soporte habitualmente utilizado para intercambiar programas. Actualmente estos medios y soportes son casi siempre ftp anónimo y/o CDROM. La carencia simultánea de estos dos citados resulta inaceptable.
- Acompañar un compromiso escrito, válido por un plazo razonable (la FSF exige 3 años), de proporcionar las fuentes a quien las pida a un coste no superior al de la distribución física por medio estándar (ftp y CDROM al menos).
- Una tercera forma de poner a disposición el código fuente, y que normalmente sólo debería aplicarse a usos no comerciales, es la de acompañar el programa con la información recibida por el licenciatarario sobre la oferta anteriormente citada del código fuente. Esto sirve para disminuir costes y no cargar a ciertos usuarios con elementos posiblemente inútiles.

[Queda claro que las fórmulas segunda y tercera sólo tienen sentido cuando la distribución del programa únicamente es en programa objeto o binario ejecutable]

La LPI no define “código fuente”. La GPL y las OSD dicen que es “la forma preferida del trabajo cuando se hacen modificaciones”. Para un ejecutable, fuente es el código completo de todos sus módulos, ficheros asociados de definición de interfaces y guiones utilizados para controlar la compilación e instalación del ejecutable. No comprende necesariamente el código que suele acompañar a los componentes principales del sistema (el compilador y el núcleo sobre el que funciona el ejecutable), salvo que el propio componente principal acompañe al ejecutable.

#### 4.5. Libertad de distribución

Todas las formas anteriores de explotación pueden ser realizadas individualmente y sin conocimiento por nadie. Pero la distribución es inherentemente relacional, ya que hay intercambio, y es en este punto en el que reside una de las más importantes polémicas dentro del sl. En su seno, la libertad de distribución no se pone en duda, libertad sin restricciones aparentes, salvo por cuestiones formales, algunas limitaciones geográficas lógicas, asuntos de poca monta comparados con la general libertad de distribuir y redistribuir programas originales o transformados.

También está claro para todos que la distribución sólo es libre si puede tener por causa cualquiera que sea lícita: ánimo de lucro, altruismo, proselitismo... Asimismo no admite dudas que el sl sólo es libre si la licencia del programa libre original persiste durante toda la vida útil del software y de sus derivados, que por lo tanto también han de ser libres. La cuestión es la de la



transmisión de efectos de las licencias, que ya avanzamos en el apartado anterior. ¿Se transmite el carácter libre de un programa a sus programas derivados siempre, o sólo para determinadas modificaciones (derivaciones, agregaciones, paquetes, bibliotecas, distribuciones), tal vez sólo para la redistribución? ¿O se reconstituye plenamente el copyright? Esto depende de la licencia misma, y por eso el movimiento del software libre se articula no mediante una forma especial de escribir código, ni por una nueva mercadotecnia, ni por el apoyo del sector empresarial, todo eso son consecuencias de mover la palanca sobre cierto punto de apoyo: esos extraños e indigeribles documentos llamados licencias de software.

El criterio para resolver el dilema reside, obviamente y una vez más, en las libertades ciudadanas, incluidas las exigencias de la libre competencia, entre las que no está -más bien al contrario- la limitación de entrada en el mercado por apropiación de resultados obtenidos en el desarrollo de software, mucho menos de software libre. Así, el copyleft es una exigencia de la auténtica libre competencia.

No se conoce ninguna norma jurídica que prohíba la apropiación del software, ahí está la LPI para garantizar que los programas pertenecen a sus autores. Pero el art. 81 del Tratado de la Comunidad Europea (ojo: en la nueva numeración de Amsterdam) califica de incompatibles con el mercado común, y quedan prohibidas, las prácticas tendentes a impedir, restringir o falsear la competencia; y en particular el limitar o controlar la producción y el desarrollo técnico, subordinar la celebración de contratos a la aceptación por la contraparte de prestaciones suplementarias sin relación con su objeto. También prohíbe el art. 82 abusar de la posición dominante, p. ej. limitando la producción o el desarrollo técnico en perjuicio de los consumidores. No es el sí el que usa estas prácticas, de hecho las dificulta. Por contra, hay paladines del software cerrado y (teóricamente) de la libre competencia que se encuentran expedientados en Bruselas por presuntas prácticas ilícitas.

Este apartado 4 trata de las libertades del usuario de software, no de los derechos de autor. Muchos entienden que la libertad del usuario no puede restringirse mediante el uso de la misma libertad. Si quien modifica un programa libre hace uso de sus derechos de autor (de su libertad, dirá él tal vez) añadiendo en la licencia del programa derivado (y en perjuicio de sus usuarios) restricciones que no figuraban en la licencia del programa libre original, lo que está haciendo no es ejercitar sus derechos de autor -que están indeleblemente unidos a la licencia libre original- sino apropiarse ilegítimamente de algo que no le corresponde a él en exclusiva, la libertad de los demás a usar y explotar libremente el nuevo programa, libertad garantizada por las constituciones, sin duda por la Constitución española cuando reconoce el derecho fundamental a la producción científica y técnica (art. 20.1.b CE).

Las licencias libres tipo BSD-original no pudieron tener en cuenta que el software estaba empezando a ser usado masivamente y a gran escala, los fabricantes intentaban patentar los productos y prescindir de una vez por todas de los desarrollos abiertos. Habiendo encontrado mejor cobijo en la ley de derechos de autor que en la de patentes, actualmente han debido frenar su expansión tras la recuperación del modelo de software libre.

Este modelo se ha repuesto del declive de los años 80 añadiendo a su definición la única prohibición importante que contiene la licencia deseable: Está prohibido al destinatario de un programa libre restringir la libertad de explotación de los programas derivados creados por él. Esta prohibición es tan importante que se dedica el apartado siguiente sólo a ella. Tiene incluso

nombre propio.

#### 4.5.1. Copyleft o prohibición de añadir restricciones sobre los programas derivados de un programa libre

La esencia jurídica del software libre se encuentra en la libre explotación de los programas por los usuarios, sin discriminación. A su vez, el autor de un programa derivado tiene el derecho exclusivo de autorizar o prohibir la explotación de su obra. Pero el programa derivado existe y es legítimo porque la licencia del programa original facilita su creación, porque es libre. Y como es libre, exige la persistencia de la libertad de uso y explotación sobre los programas derivados. De haber resistido las universidades en las últimas décadas sus carencias financieras de otra guisa, y no aprovechando a toda costa el modelo de patentes, tal vez no hubiera sido necesario tener que recordar semejantes afirmaciones. Pero ha sido necesario. El recordatorio de la existencia de libertades fundamentales, constitucionalmente garantizadas, ha tomado la forma de una cláusula prohibitiva, la cláusula copyleft.

El mejor ejemplo de cláusula copyleft es, de nuevo, la GNU-General Public License, o simplemente GPL, publicada por la Free Software Foundation. En realidad la cláusula se halla dividida en tres apartados, números 4, 6 y 10 de la versión 2 de 1991.

Las modificaciones sobre la licencia original pueden ser irrelevantes (correcciones gramaticales, de ordenación, inclusión de asuntos ajenos al derecho de autor). El copyleft se ocupa sólo de las modificaciones relevantes, que afectan a los términos de la explotación del programa. Pueden a su vez ser de dos tipos: las que hacen la explotación más libre (difíciles de imaginar) y las que la restringen, por ejemplo cerrando el software, licenciándolo sólo para uso no comercial, impidiendo su modificación, copia o redistribución. De éstas sí hay numerosos ejemplos. Son las del segundo tipo las modificaciones prohibidas por el copyleft. La LesserGPL permite una excepción “estratégica”, para las bibliotecas y otros elementos, que no podemos tratar aquí, ni es para nuestro estudio demasiado importante.

**Persistencia de la libertad del software.-** El copyleft pretende justamente la transmisión de los efectos de la licencia del programa originario a las licencias de los programas derivados, como cualquier licencia, aunque no se trata sólo de eso: Requiere seriamente la persistencia del carácter libre del software libre modificado y (re)distribuido. El copyleft preserva el carácter de si prohibiendo que de un programa libre se obtenga otro no libre o que se redistribuya con restricciones adicionales a las libertades de los destinatarios.

El copyleft no afecta directamente a los derechos del autor del programa originario, pero sí a los del autor del programa derivado en el momento de su redistribución. ¿Cómo es esto posible? Porque el segundo aceptó la licencia del primero. Copyleft no es lo contrario de copyright. El copyleft contiene lo que técnicamente se conoce en Derecho como “condición resolutoria” [otros preferirán hablar de “modo”, pero esto apenas tiene importancia]. Se trata de un suceso (condición) que, de darse, produce determinado efecto en los derechos. El suceso en nuestro caso es la infracción de una licencia, que por tanto queda desactivada (resuelta). La condición resolutoria

implícita en el copyleft se produce al añadirse restricciones a la libre explotación del programa derivado sobre las que figuraban en la licencia del programa original.

La cláusula copyleft la impone el autor de la obra original en uso de sus facultades de copyright, no en contra de tales facultades, que ya hemos demostrado no quedan afectadas por ello. Y la aceptación por el destinatario de la licencia con cláusula copyleft supone que si vulnera la cláusula, la licencia deja de ampararle a él y al programa derivado que redistribuya, que pasa a ser automáticamente ilegítimo. Lo mismo que en cualquier explotación de otras obras intelectuales distintas del software.

**Estructura de la cláusula copyleft.-** El copyleft es fácil de entender, se condensa en una prohibición. Pero lo cierto es que se encuentra formada por varios elementos:

1. Una sujeción: No cabe explotación del programa sino en los mismos términos copyleft. Cualquier explotación en términos diferentes no queda amparada por la licencia. Esto es un requerimiento general en casi cualquier contrato de cesión de derechos. La explotación indebida por alguien no afecta a todos los demás que sí ajusten el uso del programa copyleft a sus términos.
2. Una obligación: Quien redistribuya el programa copyleft u otros derivados de él, ha de poner ipso facto a disposición del receptor una licencia copyleft equivalente, sin restricciones adicionales. Como aclara la GPL (cl. 6), el licenciatario original, ahora licenciante del programa derivado, no es responsable del incumplimiento de la licencia original por terceras personas. Pero sí es responsable de ajustar la redistribución a los términos copyleft.
3. Una carga: Si se desea incorporar partes del programa copyleft a otros programas libres que tengan condiciones de distribución distintas, debe obtenerse permiso del autor de aquél. Es decir, la incorporación es posible pero su legitimidad no es automática, depende de que en la transmisión de derechos se preserven las condiciones que hacen libre al programa incorporado, y se promueva o fomente el uso compartido y la reutilización del software en general.

La cláusula copyleft se complementa con una aclaración y una excepción, ambas de poca importancia en el fondo:

- Integridad del copyleft en caso de impedimento forzoso a la libre distribución (cl. 7 GPL): No puede redistribuirse un programa copyleft si, por impedimento forzoso (decisión judicial, vínculo con patentes, etc), la redistribución no va a poder ser copyleft.
- Posibilidad de limitaciones geográficas a la distribución libre (cl. 8 GPL): Normalmente por motivos de vinculación con patentes o interfaces bajo copyright, pero siempre que se incluyan en la licencia indicaciones al respecto, claras y prominentes.

Y esta es la construcción jurídica, tomada de la FSF aunque podría servir cualquier otra. Ha sustentado, desde el Derecho y sin litigios judiciales, la realización de sistemas como GNU, del

núcleo Linux, de colectivos como Debian y de innumerables foros; ha estimulado la formación de empresas, proyectos editoriales y docentes; la aparición de cuerpos orgánicos de software libre en distribuciones multiformes... No son sus únicos frutos, como se verá en el apartado 5.

**Notas finales, un poco fuera de lugar.-** El copyleft, mediante un dispositivo jurídico impecable, da y asegura la libertad, protege al autor favoreciendo la explotación de su programa, incluso por sí mismo si lo desea, e impide en fin que nadie, salvo eventualmente él mismo y con dificultades, tome demasiado control en el desarrollo. Es un artilugio equivalente, salvando ciertas distancias, a la división de poderes del Derecho constitucional (LOCKE, MONTESQUIEU y demás). El control por el explotador del programa nunca podrá ser absoluto, pero el de los demás usuarios y desarrolladores tampoco. Y el programa con más distribución será necesariamente el mejor posible, en otro caso será corregido rápidamente. Etc.

Esto es teóricamente cierto, pero la realidad no se ajusta exactamente a esta descripción. ¿Por qué? Porque la rentabilidad financiera del copyleft sólo se manifiesta si alcanza a cubrir una rama de desarrollo mínima explotable, sea un solo programa o una plataforma completa. Si no es el caso, no deja por ello de ser rentable, pero no necesariamente en términos financieros, de generación de ingresos, de inversión y crecimiento. Hay otras rentabilidades buscadas por el copyleft, prioritarias en realidad. Hay economistas que pueden demostrar si es o no cierto lo anterior, al cabo sólo una conjetura.

Evidentemente no es cierto que el copyleft haga a un programa menos libre porque “limita la libertad del autor del software derivado”. Esta es una apreciación incorrecta. Por una parte, el copyleft preserva el carácter libre del software sin afectar en nada a la esencia del copyright (sí por supuesto al ejercicio de determinadas facultades del copyright, como cualquier contrato de cesión de derechos de autor). Segundo, entre la libertad de un número determinado de usuarios que desean apropiarse del software derivado y la del número indeterminado de usuarios que no tienen tal intención, el copyleft opta por éstos, pero no exactamente quitando libertad a aquéllos, no restringiéndoles su libertad de elección (lo que sí se produce mediante determinadas prácticas comerciales en perjuicio de los usuarios) sino hasta después de aceptar la licencia, que por tanto han de respetar. Para todos los demás usuarios, los que no desean vulnerar la libertad de distribución, el copyleft simplemente no les supone ni siquiera una prohibición, porque la mayoría nunca agotamos toda la libertad que se nos ofrece. Si el copyleft es una camisa de fuerza, al menos se la pone uno mismo. Pero más bien el copyleft es un pacto de no agresión, y esto es saludable, no vírico. Al contrario, hay pautas comerciales que sí pueden resultar víricas, lo son de hecho. Pero es tan fácil hablar en términos tales (contagio general-monopolio, etc) como poco seguro.

El copyleft, o más bien las manifestaciones de su potencia, son una materia enorme, que excede lo jurídico con creces. Se ha pretendido con lo anterior dar breve cuenta de ello. Quedan muchas cuestiones por tratar y explicar, como las dobles licencias, la remota posibilidad de un programa copyleft que entra efectivamente en dominio público, los problemas específicos de los paquetes, distribuciones, medios de almacenamiento y canales de distribución. Confiamos en una versión ulterior de este artículo para tratarlos apropiadamente.

No obstante la claridad de su construcción jurídica, y por razones que en este artículo no podemos tratar sino muy por encima, a los hombres de negocios el sistema copyleft no les gustaba.

Temían su aspecto comunista, anarquista o libertario. En esencia, no les gustaba la cláusula que impedía apropiarse, al no restringir la libertad de los demás, del trabajo realizado a costa de su dinero. También se encontraban los programadores que sufrían un temor análogo respecto del futuro fruto de su esfuerzo. Este temor provenía de un malentendido, inexcusable pero persistente, acerca de cómo rentabilizar los proyectos copyleft, normalmente origen de riqueza (fondo) si son buenos claro está; pero no necesariamente generadores de dinero (flujo) si la explotación es defectuosa. En el software no copyleft no pasa exactamente lo mismo, porque los costes de desarrollo y mantenimiento no garantizan siquiera lo primero. Por supuesto, también el software no libre puede ser objeto de explotación defectuosa. El caso es que en el movimiento del software libre se produce una bifurcación en 1997-1998. Para nosotros es “la” bifurcación, porque afectó a las definiciones y a las licencias y su clasificación.

#### 4.5.2. Autorización de restricciones adicionales: Open Source

La distribución de software, y en particular la re-distribución de software modificado, es el punto crítico del sl. La Iniciativa Open Source (OSI) surge de las Directrices Debian de Software Libre (DFSG), adaptadas en 1997 sin cambios sustanciales a unos términos más generales bajo el rótulo que se usa actualmente: Definición de Fuente Abierta (Open Source Definition - OSD). No se trata de una licencia, ni siquiera de un modelo de licencia, sino de directrices para la clasificación y adopción de licencias en productos de software (programas, paquetes y distribuciones).

La definición de fuente abierta no es del todo equivalente a la de software libre. Las diferencias esenciales son dos: Una fundamental, pues la OSD entiende por libertad del usuario una situación jurídica menos amplia que el sl, según la definición de éste hecha por la FSF, que es la que estamos utilizando en este artículo por su notable rigor y ajuste a las exigencias constitucionales y legales españolas. La otra diferencia es instrumental, está ubicada en la cláusula de persistencia del sl, que en la OSD no es ya necesariamente software libre tras la distribución, con o sin modificaciones del software originario. Se permite añadir determinadas restricciones a los términos de distribución de originales y redistribución de derivados. Esto es una simplificación compacta que debe explicarse y matizarse.

El punto de vista de la FSF no es sólo económico, que no se excluye de todos modos, sino ultrajurídico pues comprende también postulados éticos y por lo tanto filosóficos. El punto de vista de la OSI, que reconoce inspirado por la FSF, es sin embargo sólo económico (disminución de precios, apertura de nuevos mercados). Se trataba de atraer a los hombres de negocios y a importantes sectores de desarrolladores al movimiento del sl, lo que se consiguió en parte. Pero no se logró mediante la atracción de aquéllos al ámbito del sl, sino eliminando la necesidad de persistencia del sl, vale decir llevando a los usuarios al coto de los temerosos. El copyleft quedó eliminado de la definición deseable de sl, se volvió a las definiciones más antiguas, perfeccionadas es cierto, de lo que podía entenderse por software libre, expresión que además se cambió por “fuente abierta”, menos expresiva en inglés y todavía menos en castellano. Para el Derecho, la OSD no impide restringir la libertad del usuario, facilita la apropiación del software derivado, aunque su intención primordial sea asegurar al autor la posibilidad de apropiación exclusiva del fruto principal del esfuerzo invertido, sobre todo en los desarrollos más originales, con marca, etc.

**Estructura de la OSD.-** Para examinar detalladamente la OSD bastaría con tratar cuatro de los diez apartados que tiene la versión 1.0, ya que el último es tan sólo un ejemplo, sin valor normativo, y los cinco restantes tratan cuestiones que han quedado ya examinadas en apartados anteriores. Pero sólo daremos unas breves notas.

Interesan los apartados que eliminan la necesidad del copyleft y, en general, la persistencia de la libre explotación, que son los números 3, 4, 8 y 9. El número 3 permite cambiar los términos de sl del programa originario a términos no libres en el programa derivado (tipo BSD), y se justifica -erróneamente- en la necesidad de evitar al autor del programa originario el riesgo que supone a su reputación la derivación de un programa con código muy defectuoso que se pudiera atribuir no al autor derivado, sino a él, al autor del programa original. Lo mismo para evitar caballos de troya, prohibiciones locales sobre transferencia tecnológica, como en criptografía, etc. El apartado 4 exige la integridad del código original (algo legítimo) con medidas innecesariamente restrictivas. El punto 8 sirve para separar claramente un programa libre de una colección no libre de software. El programa libre lo seguirá siendo en cualquier caso, incluso si después se desagrega de la colección. Y el 9 señala que la parte no transmite su carácter al todo. Esto supone la estanqueidad de las licencias. La eficacia de la licencia de un programa no dependerá de las de los otros que se encuentren en el mismo medio o soporte.

El argumento de la posible mayor vulnerabilidad del sl frente al software no libre, por ejemplo ante caballos de troya, simplemente no tiene que ver con las condiciones jurídicas de modificación y distribución del software, y naturalmente no es tratado aquí. El software no es vulnerable en ese sentido por motivo del tipo de licencia que lleva aparejada. Sí es necesario, en ciertos ámbitos de distribución y ante cierto tipo de modificaciones (es decir, no siempre) un seguimiento adecuado de las versiones y de su autoría. Pero estas exigencias no se atienden alterando el sistema copyleft, basta el sistema de marcas, que nada tiene que ver con los derechos de autor; y un buen sistema de control de versiones, se supone. No podemos tratar estos asuntos ahora, y tal vez estemos en un error. Pero hay indicios de que estos argumentos contra el sl y el copyleft son inválidos.

Para disipar los temores de los hombres de negocios, o cualquiera que vea tras el copyleft al gran satán, es aconsejable muchas veces el uso adecuado de la Ley de Marcas. Quien desee apropiarse de sl para mantener la autoría sin riesgos de confusión hará mejor en registrar una marca, creándose así versiones oficiales de fácil control y gestión. Esto es lo que se ha hecho con Linux -que es GPL- y tantos otros casos. Todo ello sin tocar la licencia libre, incluso manteniendo la cláusula copyleft.

#### 4.6. Clasificaciones de las licencias

No vamos a reproducir las clasificaciones más importantes, que puede encontrar el lector en [www.gnu.org/philosophy/categories.html](http://www.gnu.org/philosophy/categories.html) y en el artículo de B. Perens en "Open Sources..." citado al final, Apéndice C. Aquí nos basta retener tres clasificaciones muy generales pero importantes:

1. La división básica con la que se está de acuerdo, aunque no completamente, entre software libre-software no libre. Esperamos haber convencido al lector de que el término open source-fuente abierta es menos riguroso.

2. La división GNU añade dos criterios más a su clasificación de licencias: “grado” de copyleft y compatibilidad con el sistema GNU, que aunque son de interés no podemos tratar ahora.
3. La compatibilidad Open Source se basa en criterios más simples que los anteriores de GNU, éstos más rigurosos en todos los sentidos.

Confiamos en versiones posteriores de estas notas para desarrollar todo esto como merece.

#### **4.7. Infracción de una licencia de software libre**

Para este asunto se recomienda consultar [www.gnu.org/licenses/gpl-violation.html](http://www.gnu.org/licenses/gpl-violation.html). Allí se ofrece sintéticamente el camino para reaccionar ante lo que uno cree que es una infracción de las condiciones de explotación y uso del software libre. En resumen, se trata de comprobar la infracción, documentarla, e informar al titular del copyright, único legitimado en apariencia para actuar contra el infractor. La FSF presta asistencia para reaccionar contra las infracciones de la GPL.

##### **4.7.1. Reacción por quien no es titular del copyright**

En este subapartado presentamos una conjetura, probablemente válida, pero de la que desconocemos aplicaciones prácticas: Es muy probable que en España dispongamos actualmente de una o dos fórmulas de reacción adicionales frente a un atentado contra una licencia de sl, vías que GNU-España extrañamente no menciona, y que trataremos de demostrar sumariamente.

Es cierto que, bajo una licencia de software libre con copyleft, estrictamente hablando sólo el titular del copyright (¡que no es necesariamente el autor!) tiene acción jurídica contra la infracción de su derecho, manifestado en la licencia violada. Pero también lo es que los usuarios del programa copyleft cuya licencia alguien ha infringido tienen interés legítimo en que la infracción se rectifique. Los usuarios pueden incluso haberse organizado, por ejemplo en un colectivo de desarrolladores o simplemente en una asociación de consumidores. Pues bien, la suma [interés legítimo + organización de un grupo de consumidores] abre claramente la vía judicial también para quienes no son autores, vía no basada en el copyright directamente sino indirectamente, a través de la licencia. De hecho, debería bastar el interés legítimo esgrimido por un solo usuario, pero esto puede resultar a menudo impracticable y es de más difícil digestión por los jueces.

Esta vía está fundamentada muy sencillamente en el ensamblaje de varios preceptos de la Constitución española y algunos más de otras leyes ordinarias. En síntesis, se trata de la protección judicial de un genuino interés legítimo, para el que los grupos, y singularmente las Asociaciones de Consumidores, se dice que tienen “legitimación activa”. Puede llegar a obtenerse incluso protección administrativa (oficinas municipales de consumo, p. ej.). No nos extenderemos aquí, sólo citaremos los artículos, que el lector puede consultar por su cuenta: arts. 9.2 y 24.1 CE, 2.1.e y .f LCU y 6.7º LEC.

Con esto ha debido de quedar tratado lo más importante. Para completar el cuadro sólo resta tratar brevemente dos asuntos complementarios y de distinto interés.

## 5. Expansión del modelo de las licencias de software libre

Este apartado se incluye como ilustración del potencial de la construcción jurídica contenida en las licencias de software libre, y no pretende ni de lejos dar cuenta al mismo nivel que en los apartados anteriores de los modelos que se citan.

### 5.1. La Licencia de Documentación Libre. Otras licencias similares

No vamos a tratar de la licencia de documentación libre, GNU-FDL (GFDL o simplemente para nosotros FDL), propuesta por la FSF para los manuales de uso, documentación técnica y otros textos. El lector sabrá ahora leerla y comprender su significado por sí solo, es similar a la GPL. Estrictamente hablando, en Derecho español no es necesaria, pues la documentación de un programa se protege con el mismo copyright que el del programa mismo. Pero en la práctica es muy útil, por dos razones: En primer lugar permite modular las exigencias de una licencia copyleft, pensada para el software, a las de los textos escritos, por ejemplo preservando algunos fragmentos, normalmente todo el contenido no técnico, cláusulas externas, etc, de la característica autorización de libre explotación. Es la licencia apropiada para la documentación del software, pero no sólo. Efectivamente, y en segundo lugar, una licencia como la FDL permite, sin merma de los derechos de autor, la formación de proyectos colectivos de documentación libre, normalmente técnicos y científicos. No es apropiada para libros de poemas y memorias, al menos no se pensó con esa finalidad. En general, la documentación FDL debe permitir la libre modificación del contenido técnico, no del valorativo-personal.

Para la FDL, en la documentación libre debe distinguirse claramente los siguientes textos de un manual o trabajo técnico cualquiera:

- Textos de Cubierta, que son los de Portada y Contraportada.
- Página de Título, que incluye la página de título misma y además las necesarias para contener legiblemente cuanto la FDL requiere. En los libros con formatos carentes de esta página, se llama Página de Título al texto inmediato a la aparición más prominente del título de la obra, que precede al comienzo del Cuerpo del Texto.
- Cuerpo del Texto.
- Secciones Secundarias, para materias como notas editoriales, advertencias legales, valorativas, etc, sin conexión directa con el Cuerpo del Texto. Dentro de ellas se encuentran las Secciones Invariantes, inalterables en las modificaciones.
- Puede, incluso a veces debe, incluirse secciones como Historia, Reconocimientos y Dedicatorias. Obligatoriamente han de quitarse en las modificaciones las aprobaciones (endorsements), homologaciones o similares que recaigan sobre el texto original, pues son exclusivos de éste.

La advertencia de copyright y el aviso de licencia FDL deben contener las siguientes indicaciones: a) cuáles son los Textos de Cubierta, si los hay, ya sean de portada o contraportada; y b) cuáles



son las Secciones Invariantes. No puede exponerse ahora ni siquiera los aspectos más relevantes de este singular sistema de liberación de facultades de propiedad intelectual.

Tal vez a los teóricos del *sl* les sorprenda que el ejercicio del Derecho sea una profesión abierta, en el sentido de que los profesionales intercambiamos libremente nuestros conocimientos y argumentaciones jurídicas (¡no los derechos sobre nuestros libros!). Es un hecho normal, indiscutido, el que unos a otros nos fusilemos nuestros textos, desde siempre jamás, con ciertos límites, claro está. La FDL y otras licencias parecidas tal vez completen el régimen de libertad intelectual de que disfrutamos los juristas, propiciando p. ej. textos universitarios libres, de los que tan necesitados están los estudiantes, los profesores y los profesionales.

El examen de la documentación libre requeriría un trabajo por sí solo, pero en cuanto al contenido jurídico de las licencias lo esencial ya ha quedado expuesto antes. Además de la FDL se puede encontrar documentación amparada en licencias libres como la FreeBSD Documentation Lic., la Apple's Common Documentation Lic., la algo menos estricta Open Publication Lic., que puede o no aplicarse "copylefting", y algunas más. También se dispone de la Open Science Lic., licencia libre y copyleft, redactada para dar esa cobertura a datos en general, no sólo software.

Adviértase que otra idea surgida en el seno de Debian, Open Hardware, no tiene directamente que ver con las licencias de software. Consiste en un programa de certificación, exactamente de calificación de hardware según ciertas especificaciones, diseñado para verificar que una configuración de máquina es apta para un sistema Linux o FreeBSD. Incluye la promoción de un certificado de la verificación, llamado "certificado open hardware" para uso por los vendedores y consumidores.

## 6. Software no libre

Comprende todo el software que no es libre. Es un género, con muchas especies.

Primero tenemos el software semilibre, cuya licencia contiene autorización para la libre copia, modificación y distribución, pero siempre que no tengan carácter comercial. Esta es una restricción muy importante sobre todo a la distribución, que desnaturaliza la finalidad del *sl*, por eso no lo calificamos como libre. Por ejemplo, el software semilibre no permite su incorporación a paquetes copyleft. La restricción de la autorización a la explotación no comercial añade muy poco a la libertad de uso, o más bien restringe la posibilidad de ser utilizado. Parte de una idea errónea o de una valoración incorrecta del ánimo de lucro. La primera versión de la GPL encajaba en esta definición, pero la versión 2 (1991) retiró la restricción, por innecesaria.

El segundo grupo de software no libre es el "propietario". Esta denominación es inexacta por más de un motivo. Con ella quiere denominarse lo que la FSF llama "software no libre que no es semilibre". Podría decirse entonces que es software aprisionado, enjaulado, inaccesible. Desde luego está lleno de limitaciones al libre uso, es intransferible, incopiable sin convertirse uno en infractor, inmodificable (no mejorable por los usuarios directamente) y de ningún modo redistribuible. En este artículo preferimos denominarlo software "cerrado". Así entendidos, estos programas ni siquiera forman parte de la máquina. No serían ni contendrían piezas que uno puede examinar, reparar, o sustituir si son defectuosas. Pero la realidad es que el software sí forma parte

de la máquina, y no hay razón para que quien puede reparar o repintar o recauchutar un automóvil utilitario no pueda hacerlo en un programa utilitario, sobre todo si es un sistema operativo o un compilador.

Hay otras subespecies de software no libre, las repasamos en los apartados siguientes, apuntando rápidamente las restricciones que imponen a las libertades básicas de los usuarios.

En este artículo no tratamos el pseudotipo “software comercial”, categoría para nosotros inútil porque no es puesta en cuestión por el sí; aunque sí por el software semilibre, que no admite el uso comercial de los programas licenciados. Tampoco podemos tratar fenómenos tan extraños como MSAgent, una especie de “software libre revocable” que no es en modo alguno software libre.

### 6.1. Restricciones de uso

Lo que califica a un programa como semilibre es que su uso y explotación quedan limitados a un destino no comercial. Esta restricción no resulta aceptable, ya lo hemos dicho, y no la trataremos más.

También el software cerrado impone graves restricciones de uso del software. Se prefiere en este artículo llamar al software propietario “software cerrado” no por afán de neologismo sino para ilustrar mejor que su uso y explotación no son libres, además de por corrección idiomática y jurídica. Sobre él no es preciso decir mucho más de cuanto ha quedado expuesto en los apartados anteriores, y quien desee información adicional hará bien en acudir a un manual de propiedad intelectual sobre programas de ordenador, cosa que este artículo no pretende ni puede ser.

No tiene sentido examinar las limitaciones impuestas en las licencias estándar de software cerrado, ya se han citado algunas en apartados anteriores, como la fórmula de aceptación de licencia consistente en romper los precintos del paquete de CDs, es decir, antes de toda posibilidad de uso y comprobación, contra el art. 10bis LCU. En el software cerrado simplemente la licencia no tiene otra finalidad que la de plegarse de modo obsesivo a las facultades del copyright (de la LPI en España), prescindiendo de cualquier consideración sobre las libertades constitucionalmente garantizadas de los usuarios de software, sin otra finalidad que la explotación exclusiva, normalmente no por su autor sino por un titular derivado, y lesionando (sin advertirlo, claro está) sus propios intereses de mercadotecnia y desarrollo.

El uso queda prohibido sin licencia aceptada, imposibilitándose la instalación, restringiéndola a monopuesto, o necesitándose una reactivación de la licencia si se supera determinado número de componentes del hardware (!). No pueden evitar estas licencias el que se apliquen las normas imperativas que hemos visto en el apartado 2.1.3, pero quedan restringidas al máximo, de hecho se imposibilitan porque las modificaciones necesarias no son factibles sin el código fuente, nunca disponible.

Con lo visto hasta ahora, queda claro que la subcategoría “shareware” se refiere a programas cerrados que restringen el uso si no se paga una cantidad de dinero adicional pasado un período de prueba. Normalmente son programas incompletos, mejor dicho mutilados.

## 6.2. Limitaciones a la libre reproducción y copia

Estas actividades no son autorizadas en el software cerrado, e incluso pueden impedirse automáticamente. Las copias de seguridad se restringen en lo posible. La copia privada es calificada incorrectamente de piratería en varias licencias que se ha podido consultar.

## 6.3. Límites a libre transformación y modificación

En el software cerrado y sus variantes la transformación está prohibida, y ni siquiera es factible, pues el código fuente nunca se acompaña en la distribución. Los componentes del software no pueden separarse legítimamente. Caso de hacerse alguna transformación legítima, si la licencia se revoca los archivos fuentes derivados han de destruirse (!). El shareware tampoco es modificable, pues casi siempre es cerrado y además no acompaña las fuentes.

El denominado “freeware” es normalmente software cerrado que, aunque puede redistribuirse con muchos límites -no libremente- no puede sin embargo ser modificado, entre otras razones porque el programa fuente no está disponible. No es en modo alguno software libre contra lo que indica su (impropio) nombre.

## 6.4. Límites a la libre distribución

En el software cerrado la distribución y la redistribución están sencilla y rotundamente prohibidas, salvo en los casos del llamado freeware. Esto incluye la prohibición de alquiler y préstamo.

## 7. Conclusión

El matemático David Hilbert, refiriéndose a la teoría de conjuntos creada por Georg Cantor, objeto del desprecio de otros matemáticos, decía “nadie podrá expulsarnos del paraíso que Cantor ha creado para nosotros”. Desde los ingenieros y programadores hasta los usuarios menos avezados, pasando por distribuidores, empresas comerciales y universidades, formamos legión quienes hacemos correr sobre nuestras máquinas programas libres, hackeados y compilados por nosotros mismos conforme a nuestras necesidades y gustos, que nos prestamos y copiamos libremente, con la destreza y seguridad que permiten el mejor banco de pruebas posible, una variedad inagotable de soluciones, siempre en renovación, y una documentación de calidad superior a la estándar de los programas no libres. El software libre, sobre todo si es copyleft, mantiene e impulsa el entusiasmo universal en la computación, especie en peligro en las dos últimas décadas, e incluso comienza a servir de referencia para otros ámbitos de la libertad intelectual.

No es seguro que esta apreciación de la realidad sea completa, pero es la de mucha gente. Además, es el objeto de una polémica contemporánea trascendental, que no se trata en este artículo sino muy de pasada. Las piezas puestas en juego por el software libre son muchas y poderosas. Es cierto que el software libre trata de la libertad, y que éste no es un asunto sólo comercial o

industrial. Simplemente, es un asunto muy grande, de enjundia, origen o final de conflictos a veces muy serios.

En fin, tal vez la cosa esté a estas alturas algo más clara y podamos parafrasear a Hilbert tranquilamente, pero sin bajar la guardia: Nadie podrá expulsarnos del paraíso que la GNU-GPL ha creado para nosotros.

## **8. Apéndices**

### **8.1. Apéndice A - Abreviaturas utilizadas**

- CC - Código civil
- CE - Constitución española
- CP - Código penal
- Disp. Ad. - Disposición adicional
- Disp. Trans. - Disposición transitoria
- FSF - Free Software Foundation
- GNU-GPL - Licencia Pública General del proyecto GNU - GNU's Not Unix
- LCU - Ley de consumidores y usuarios
- LEC - Ley de enjuiciamiento civil
- LPI - Ley de propiedad intelectual
- MEDC - Ministerio de Educación, Cultura y Deporte
- OSD - Definición de Fuente Abierta, de la Open Source Initiative
- OSI - Open Source Initiative
- RD - Real Decreto
- RTV - Radiotelevisión
- sl - Software libre
- UE - Unión Europea

## 8.2. Apéndice B - Leyes, reglamentos, tratados internacionales

Disposiciones españolas:

- La LPI vigente está recogida en el texto refundido de 1996, aprobado por Real Decreto Legislativo 1/1996 de 12 de abril, publicado en el BOE núm. 97 de 22 de abril. Las últimas reformas se produjeron en marzo de 1998 y enero de 2000. Está en elaboración una nueva modificación, por motivo de la Directiva 2001/29/CE del Parlamento y Consejo, 22 de mayo, publicada el 22 de junio 2001, sobre armonización de determinados aspectos de los derechos de autor y derechos afines en la sociedad de la información.
- Ley orgánica, de protección al honor, a la intimidad personal y familiar y a la propia imagen, 1/1982 de 5 de mayo.
- Ley general para la defensa de los consumidores y usuarios, 26/1984 de 19 de julio.
- Ley de patentes, 11/1986 de 20 de marzo.
- Ley de protección de las topografías de los semiconductores, 11/1988 de 3 de mayo.
- Ley de marcas, 17/2001 de 7 de diciembre.
- El Código penal español fue aprobado por la Ley Orgánica 10/1995 de 23 de noviembre, BOE 281 de 24 de noviembre.
- Ley de enjuiciamiento civil, 1/2000 de 7 de enero.
- Real Decreto 1584/1991 de 18 de octubre, que aprueba el Reglamento del Registro General de la Propiedad Intelectual. Es el que viene aplicándose hasta que se encuentre totalmente en funcionamiento el sistema de registro diseñado por el Real Decreto 733/1993 de 14 de mayo, que aprueba el nuevo Reglamento del Registro General de la Propiedad Intelectual.
- RD 114/2000 de 28 de enero (BOE 33 de 8 de febrero), de la Comisión Interministerial para actuar contra las actividades vulneradoras de los derechos de propiedad intelectual e industrial (Comisión antipiratería).
- Reglamento del Consejo de la UE que prohíbe la comercialización de mercancías piratas y la intervención de las aduanas para impedirlo, Rgl.(CE) 241/1999 de 25 de enero, DOCE 2.2.1999 L 27.

Convenios internacionales:

- Convenio de 14 de julio de 1967 que establece la Organización Mundial de la Propiedad Intelectual.
- Convenio de Berna de 9 de septiembre de 1886 para la protección de las obras literarias y artísticas, revisado en París el 24 de julio de 1971.

- Convención Universal de Ginebra de 6 de septiembre de 1952 sobre Derecho de Autor, revisada en París el 24 de julio de 1971.

La Organización Mundial de la Propiedad Intelectual (OMPI) con sede en Ginebra, y la Organización Mundial del Comercio (OMC) son foros generadores de importantes documentos y acuerdos internacionales en materia de derechos de propiedad intelectual.

### 8.3. Apéndice C - Referencias

Se incluye sólo una breve nota de los materiales utilizados.

Un libro introductorio, que no trata a fondo las cuestiones jurídicas pero sí el tema general del software libre, WAYNER, P., “La ofensiva del software libre”, ed. Granica, Barcelona, 2001.

Los sitios web en los que puede encontrarse información sobre los asuntos tratados en este artículo son naturalmente innumerables. Un buen libro en línea se encuentra en [www.oreilly.com/opensources/](http://www.oreilly.com/opensources/), “Open Sources: Voices from the open source revolution”, con artículos de E. S. Raymond, M. K. McKusick, S. Bradner, R. Stallman, M. Tiemann, P. Vixie, L. Torvalds, R. Young, L. Wall, B. Behlendorf, B. Perens, T. O’Reilly, y J. Hamerly y T. Paquin con S. Walton, ed. O’Reilly Ass., Inc., 2000.

Para conocer los fundamentos del software libre, y no sólo los jurídicos, es evidente que el directorio recomendado está en [www.gnu.org/philosophy/](http://www.gnu.org/philosophy/). La mejor aplicación práctica de la teoría del proyecto GNU está basada en [www.debian.org/social\\_contract](http://www.debian.org/social_contract). Un buen lugar para empezar a leer sobre programas libres y sus problemas en cuanto a licencias es [www.opensource.org](http://www.opensource.org).

En castellano puede consultarse <http://gsrc.escet.urjc.es/sobre/>, grupo SoBre de software libre.

Una lectura nueva y práctica se encuentra en el proyecto (proposición diríamos en España) de ley sobre uso del software libre en la Administración pública, remitido al Congreso peruano el 9 de abril de 2002 por los congresistas E.VILLANUEVA NÚÑEZ y J.RODRICH ACKERMAN, <http://www.gnu.org.pe/rescon.html>.