

# Informe sobre licencias libres

Miquel Vidal  
GSyC/Libresoft

Diciembre 2008  
Versión 1.0

## Índice

<b>1. Introducción</b>	<b>2</b>
1.1. Qué es la Propiedad Intelectual . . . . .	2
1.2. Tipos de derechos . . . . .	3
1.3. Autores y titulares de los derechos . . . . .	4
<b>2. Licencias</b>	<b>4</b>
2.1. Licencias libres . . . . .	5
2.2. Tipos de licencias libres . . . . .	6
<b>3. Licencias libres para software</b>	<b>7</b>
3.1. La GPL . . . . .	7
3.2. La LGPL (Licencia Pública General Reducida de GNU) . . . . .	8
3.3. La licencia Affero . . . . .	9
3.4. El caso de la EUPL . . . . .	10
<b>4. Licencias libres para documentación</b>	<b>12</b>
4.1. Licencia de documentación libre de GNU . . . . .	12
4.2. Creative Commons . . . . .	12
4.3. ‘Freedom Defined’: obras culturales libres . . . . .	14
<b>5. Recomendaciones</b>	<b>15</b>
5.1. Escoger una licencia para el software . . . . .	15
5.2. Escoger una licencia para documentación . . . . .	16
<b>6. Conclusiones</b>	<b>18</b>

# 1. Introducción

## 1.1. Qué es la Propiedad Intelectual

La Propiedad Intelectual es un concepto genérico que puede referirse a cosas distintas, dependiendo del contexto y, especialmente, del sistema jurídico al que haga referencia (derecho anglosajón o derecho continental). En general, la WIPO (Organización Mundial de la Propiedad Intelectual) la define en relación a las creaciones del intelecto: «las invenciones, las obras literarias y artísticas, los símbolos, los nombres, las imágenes y los dibujos y modelos utilizados en el comercio» y lo divide en dos categorías: *propiedad industrial* (invenciones, patentes, marcas) y *derechos de autor* (copyright).<sup>1</sup> Esta definición de propiedad intelectual que abarca también a la propiedad industrial es propia del *common law* o derecho anglosajón. Por el contrario, en el sistema jurídico español se suele reservar el término *propiedad intelectual* para los llamados «derechos de autor» o *copyright*. En la práctica es solo un problema nominal, porque tanto en el derecho anglosajón como en el derecho continental la propiedad industrial y el copyright (o derechos de autor) se rigen por leyes distintas, tienen fines distintos y obedecen a lógicas distintas.<sup>2</sup>

En cualquier caso, debemos distinguir claramente entre tres conceptos que pueden o no caer bajo la denominación de propiedad intelectual:

- *Derechos de autor (copyright)*: protegen las obras literarias, artísticas o científicas de usos no autorizados (copia, modificación, distribución, etc.). Incluye a los programas de ordenador. De los derechos de autor se ocupa la Ley de Propiedad Intelectual (LPI) en España.
- *Marcas registradas (trademarks)*: Protege los signos distintivos (logos, grafismos, etc.) y los nombres (incluyendo los nombres de dominio) de una organización. Se trata de evitar que confusiones en nombres comerciales y que pueda haber dos productos que se llamen igual. En España forman parte de la propiedad industrial y están cubiertos por un registro específico. Existe también un sistema de marcas comunitario (UE) e internacional, con leyes y tribunales que resuelven conflictos, renovación, cesión, caducidad, etc.
- *Patentes*: otorgan derechos exclusivos para la explotación de una invención susceptible de ser comercializada a cambio de que el *inventor* divulgue el método o procedimiento (esto es, *registre* la patente). En España son las leyes sobre propiedad industrial, claramente diferenciadas de la LPI, las que se encargan de regular las patentes, invenciones, etc. en campos específicos como química, farmacia, biomedicina, biotecnología, etc. En Japón y Estados Unidos pueden patentarse también métodos de programación,

---

<sup>1</sup><http://www.wipo.int/about-ip/es/>

<sup>2</sup>En este texto, usaremos *propiedad intelectual* en el sentido más restringido que se le da en el derecho continental, aunque para evitar ambigüedades es preferible evitarlo siempre que sea posible y hablar de *derechos de autor* o de *copyright*.

como por ejemplo algoritmos. No así en Europa, donde estas patentes no son operativas pues se consideran ideas y, hasta la fecha, los programas de ordenador se protegen exclusivamente a través de las leyes de derechos de autor o copyright.

Una vez aclarado el objeto susceptible de propiedad intelectual (que lo es por el solo hecho de su creación y sin necesidad de registro), conviene señalar que lo que se protege es la forma o expresión concreta de una obra (la obra «como tal», línea a línea, palabra por palabra), no las ideas, argumentos o material de inspiración. En otras palabras: se protege la novela, no el argumento de la novela; se protege el programa de ordenador (su código fuente y sus binarios), pero no los algoritmos, procedimientos, ideas o técnicas empleadas para realizarlo.

En resumen, el código informático está protegido en todos los países por los derechos de autor (llamado *copyright* en el ámbito anglosajón). ¿Y qué es exactamente lo que el copyright protege?:

- El programa como tal: esto es, las instrucciones en la forma que sea (código fuente o binarios compilados).
- La descripción del programa (por ejemplo, su UML).
- Material adicional (manuales de usuario, guías, etc.).
- Interfaces (gráficos, sonido, tipografías...).
- Bases de datos.

## 1.2. Tipos de derechos

La LPI distingue y reconoce dos tipos de derechos de propiedad intelectual:

- *Derechos morales*: reconocimiento de autoría, divulgación, integridad de la obra, retirada de la obra del comercio, etc. Solo existen como tales en el derecho continental (aunque en el derecho anglosajón puede recibir protección por otras vías, especialmente en relación al plagio).
- *Derechos de explotación* (“copyright” *per se* en el derecho anglosajón): son los derechos patrimoniales y se componen de los derechos de reproducción, distribución, comunicación pública, puesta a disposición y transformación (adaptación u obra derivada). Corresponde al autor el ejercicio exclusivo de los derechos de explotación de su obra, lo cual incluye también la posibilidad de ceder todos o parte de estos derechos a terceros (intermediarios, editores, etc.) mediante el correspondiente contrato de cesión para su explotación. Se habla entonces del «titular» de los derechos (o «derechohabiente»), que no tiene por qué ser el autor. Las licencias sirven para ceder todos o parte de estos derechos.

Los derechos de explotación tienen un límite temporal: en su origen oscilaba entre 14 y 28 años y con posterioridad se ha ido extendiendo hasta fijarse en 70

años tras la muerte del autor. Después, la obra pasa al dominio público. Por el contrario, la mayoría de los derechos morales (en particular el reconocimiento de autoría) no expiran, ni siquiera cuando la obra ya está en el dominio público.

### 1.2.1. Excepciones o *fair use*

Existen algunas excepciones a los derechos de autor (llamados «límites» en el derecho español y *fair use* en el derecho anglosajón) como por ejemplo el derecho de cita y de enseñanza, el derecho de copia privada, el derecho a la información sobre temas de actualidad, derechos de préstamo por parte de entidades como museos, bibliotecas, fonotecas, hemerotecas, etc. Estas excepciones (o *fair use*) varían entre una legislación u otra, e incluso en la misma pueden variar o ser interpretados de manera distinta según el juez.

## 1.3. Autores y titulares de los derechos

Se considera autor a la persona natural (o jurídica, en los casos previstos por la ley) que crea alguna obra literaria, artística o científica. De acuerdo a la LPI, podemos diferenciar entre dos tipos de obras realizadas entre varios autores:

- *Obra en colaboración*: el resultado unitario de la colaboración de varios autores, donde los derechos corresponden a todos ellos. Los coautores de la obra en colaboración podrán explotar separadamente sus aportaciones, salvo que causen perjuicio a la explotación común.
- *Obra colectiva*: creada bajo la coordinación de una persona natural o jurídica que la edita y divulga bajo su nombre. Es una obra unitaria y está constituida por la reunión de aportaciones de diferentes autores sin que sea posible atribuir separadamente a cualquiera de ellos un derecho sobre el conjunto de la obra realizada. Salvo pacto en contrario, los derechos sobre la obra colectiva corresponderán a la persona que la edite y divulgue bajo su nombre.

Los derechos de explotación de una obra producida por un empleado es propiedad de la empresa. Sin embargo, los derechos morales pertenecen al autor original (o, en su caso, a sus herederos).

## 2. Licencias

Una licencia es una especie de «contrato» unilateral («no sinalagmático») entre el autor y el usuario.<sup>3</sup> Mediante la licencia, el autor ejerce el derecho que la ley le confiere en exclusiva para decidir los términos en que distribuye su obra,

---

<sup>3</sup>Las licencias no son exactamente «contratos», pero se trata de una discusión tecnojurídica que excede claramente las intenciones de este documento y que no tiene relevancia para los legos. Lo importante es entender que una licencia funciona de forma parecida a un contrato, una vez que el beneficiario de la licencia acepta sus términos.

pudiendo ceder todos o parte de sus derechos de explotación a los usuarios que estén autorizados a usar la obra. No es necesario firmar ningún contrato pero, para ejercer los derechos que la licencia concede, deben aceptarse necesariamente los términos de la licencia. Si no se hace así, se pierde el derecho a usar la obra en forma alguna. Para beneficiarse de los derechos de uso de la obra, a veces es necesario aceptar el acuerdo de usuario final (llamado EULA, o CLUF en español) o una «licencia de rompe y rasga» (*Shrink wrap contract*). En el software libre, sin embargo, no es necesario aceptar EULAS para ejercer los derechos que se te conceden, pese a que algunas veces se usan con carácter «informativo».

Las licencias de software se basan por tanto en la legislación sobre derechos de autor. En España, el marco legal que regula las licencias de software es la Ley de Propiedad Intelectual (LPI), una ley que ha sido reformada recientemente (2006) y que está ya armonizada con la respectiva legislación europea. Los tipos de obra que la ley considera bajo su protección incluyen cualquier obra original literaria, artística o científica expresada en cualquier medio o soporte, tangible o intangible. Esto incluye también las bases de datos, los programas de ordenador y la documentación asociada. También es objeto de propiedad intelectual las traducciones, actualizaciones y adaptaciones, los resúmenes o extractos y cualquier transformación de una obra literaria, artística o científica.

## 2.1. Licencias libres

Las licencias libres se comportan a todos los efectos como cualquier otra licencia de software, salvo que otorga ciertas libertades a su receptor, en lugar de restringírselas. Su uso es sencillo, basta con acogerse a los términos de la licencia, ni siquiera es necesario registrar la obra. Las licencias públicas libres (como las que reconoce la OSI o la FSF) se remiten a un marco jurídico internacionalmente aceptado y no requieren de adaptación local ninguna.

Las licencias libres son por tanto el dispositivo jurídico para implementar legalmente las cuatro libertades del software libre (uso, copia, modificación y redistribución). En otras palabras, las licencias libres son el dispositivo legal que convierte un programa de ordenador en software libre. De ahí su importancia.

Para considerarse software libre, deben darse las cuatro libertades que definen al software libre de forma simultánea. Son estas:

1. Libertad de ejecutar el programa como se desee, con cualquier propósito (libertad de uso).
2. Libertad para estudiar el código fuente y modificarlo para que haga lo que se desea o se necesita que haga (libertad de modificación).
3. Libertad para hacer copias y distribuirlas a otros (libertad de copia).
4. Libertad de publicar o distribuir versiones modificadas (libertad de redistribución).

No basta con que se concedan algunas de ellas, sino que han de darse las cuatro de forma simultánea (por ejemplo, una cláusula que impida su venta o restrinja su uso por razones de cualquier tipo, no es software libre).

Hay muchas licencias de software libre, cualquiera puede escribir una, basta con que conceda al usuario las cuatro libertades mencionadas. Las licencias públicas libres de propósito general (como la GPL, la BSD, la GFDL, etc.) tienen la ventaja de que ahorran al programador el trabajo de realizarlas *ad hoc*, garantizan la compatibilidad entre el software bajo la misma licencia y ofrecen tranquilidad a programadores y usuarios sobre su consistencia jurídica.

En síntesis:

- Las licencias fijan las condiciones de uso y distribución de una obra.
- Sin licencias no existiría la posibilidad de liberar las obras.
- Las licencias libres son más que una licencia: son una declaración de principios, un contrato social.

### 2.1.1. Cláusulas añadidas

En general, cualquier restricción en forma de cláusula de la licencia que se añada a las cuatro libertades citadas suele convertir una licencia en no-libre (o privativa). Por ejemplo, no se puede de ningún modo restringir el uso *ad hoc* (cláusulas que tratan de evitar usos militares, de clínicas abortistas, de ciertos grupos o países, etc.). Especialmente difundida es la cláusula «no comercial» (que excluye los usos de la obra cuando media ánimo de lucro) o la cláusula «sin obra derivada» (frecuente en obras artísticas). Ninguna licencia que contenga una cláusula no comercial o no derivada puede considerarse libre de acuerdo a ninguna de las definiciones de software libre existentes (la FSF, la OSI o la DFSG).

Sin embargo, nos podemos encontrar con ciertas cláusulas admisibles en las licencias libres, porque no están destinadas a restringir libertades, sino a preservarlas:

- Reconocimiento de autores (tal atribución no debe impedir el uso normal de la obra ni imponer condiciones imposible de cumplir a terceros).
- Transmisión de libertades (copyleft o *share-alike*).
- Protección de libertades (prohibición de añadir “medidas técnicas”, como DRM, que impidan ejercer algunas de las libertades a terceros).

## 2.2. Tipos de licencias libres

Todas las licencias libres comparten las cuatro libertades, pero pueden distinguirse dos grandes categorías en función de si obligan a no a que las obras derivadas se distribuyan con la misma licencia (o una equivalente).

- *Licencias tipo copyleft* (o robustas) el autor permite la modificación siempre que la nueva obra derivada conserve las mismas libertades que la original. También se las conoce como «licencias recíprocas», *share-alike* o «compartir-igual» (GPL, GFDL, CC-by-sa).
- *Licencias permisivas* (o minimalistas): el autor solo solicita que se mantenga su atribución, pero permite cualquier modificación sin condiciones, incluyendo que la obra derivada sea privativa (CC-by, \*BSD, Apache).

Las licencias de tipo copyleft no exigen ninguna condición mientras no se distribuya el software. Es decir, mientras no se modifique y se intenten distribuir copias del software modificado, la cláusula copyleft no entrará en acción. Por ejemplo, alguien puede introducir cambios en un programa GPL para uso privado y en modo alguno está obligado a distribuir los cambios. Por tanto, desde el punto de vista del usuario que hace un uso privado del código (no lo redistribuye), las licencias permisivas y las licencias copyleft se comportan exactamente igual.

Por último, existe el *dominio público*, que no es propiamente una licencia sino la situación en que queda una obra cuando el copyright ha expirado, o bien el autor ha renunciado de forma definitiva e irreversible a todo derecho de copyright presente o futuro (incluido el de sus herederos), por lo que se pueden usar, copiar y crear obras derivadas con cualquier licencia, libre o privativa. En el caso del software se necesitará además el código fuente para que pueda considerarse libre.

## 3. Licencias libres para software

### 3.1. La GPL

#### 3.1.1. Introducción

La **Licencia Pública General** (GPL, *General Public License*) de GNU es la licencia libre más popular y extendida: entre un 50% y un 70% del software libre está bajo la GPL, incluido el kernel Linux. También es la primera de todas ellas: sirvió de modelo a todas las que vinieron después y ha acompañado y sustentado al movimiento de software libre a lo largo de sus más de veinte años de historia.

La Free Software Foundation, creadora de esta licencia, consideró que no bastaba con conceder las cuatro libertades clásicas, sino que debía asegurarse que todo usuario que obtuviera una copia del programa recibiese estas mismas libertades. Por esta razón, además de las cuatro libertades comunes a todas las licencias libres, la GPL incorpora desde su primera versión una cláusula conocida como copyleft.

El copyleft requiere que la distribución de una versión modificada conserve idénticas condiciones que la obra original. De ese modo, se garantiza que toda obra derivada a partir de software protegido por la GPL conservará siempre las

mismas libertades. El copyleft no persigue restringir derechos al usuario, sino garantizar que el software libre bajo GPL no deje de ser libre. Por eso a esta cláusula se la conoce también como «licencia recíproca» o «compartir-igual» (*share-alike*). El copyleft es pues, en términos prácticos, una forma de defender activamente las libertades de los usuarios.

La importancia de la GPL va más allá de lo cuantitativo y de lo estrictamente jurídico:

- Plasma por primera vez el principio de copyleft.
- En ella se han basado muchas otras licencias libres, también fuera del software, por ejemplo la que se usa en Wikipedia.
- Sin la GPL, el copyleft solo habría sido una buena idea de tantas y la cultura de lo libre no existiría tal y como la conocemos.

### 3.1.2. Historia de la GPL

Durante la década de 1980, tras el nacimiento del Proyecto GNU (1983), las primeras licencias libres del Proyecto GNU estaban vinculadas a cada programa (por ejemplo, la licencia GNU Emacs o la licencia GNU GCC). Se extendió la costumbre de copiar la licencia y cambiarle el título, lo cual, pese a ser idénticas en todo salvo en la referencia al programa al que iban asociadas, creaba problemas de compatibilidad mutua. Por eso en 1989, Richard Stallman, el padre del software libre, depuró esas licencias particulares para crear una licencia pública de propósito general, la General Public License versión 1, a la que podía acogerse cualquier programador para licenciar su programa bajo los términos de dicha licencia.

Hasta 1992 en que aparecieron las primeras versiones de Linux, la GPL era usada generalmente por el propio Proyecto GNU. En 1991 se publicó la GPL versión 2 cuya mayor novedad era la inclusión de la cláusula que Stallman denomina «libertad o muerte» (*Liberty or Death*). Esta cláusula era una primera defensa contra imposiciones externas (por ejemplo, por infracción de patentes) que obligasen a distribuir el software limitando alguna de las libertades originales. Si se daba ese caso, la cláusula indicaba que el software no podría ser distribuido en modo alguno (“lo mataba”) para evitar males mayores (como que un propietario de patentes malicioso exigiese un pago por cada software libre distribuido). Esta cláusula fue un antecedente de otras que sería necesario incluir después en la versión 3, siempre con el objetivo de que se mantuviesen las libertades para las que fue concebida la licencia GPL.

## 3.2. La LGPL (Licencia Pública General Reducida de GNU)

La **LGPL** o **GNU Lesser General Public License** (anteriormente conocida como GNU *Library General Public License* o «Licencia Pública General para Bibliotecas de GNU») fue diseñada para establecer un compromiso estratégico entre el modelo copyleft estricto (tipo GPL) –que obliga a que cualquier



código enlazado o fusionado sea compatible con la GPL– y el modelo de licencias minimalistas (tipo BSD) –que permiten enlazarse y fusionarse con código privativo–. Escrita en 1991 por Richard Stallman y Eben Moglen (y actualizada en 1999 y 2007), fue pensada originalmente para librerías de software, de modo que software no GPL (incluyendo por ejemplo un ejecutable independiente privativo) pudiese enlazarse dinámicamente con librerías bajo la licencia LGPL, considerándose a aquel como una obra independiente que utiliza la librería. En ese caso, se aplicaría el párrafo 5 de la LGPL:

5. Un programa que no contiene derivado de ninguna porción de la Librería, pero está diseñado para trabajar con la Librería al ser compilado o enlazado con ella, se denomina un «trabajo que usa la Librería». Dicho trabajo, por separado, no es un trabajo derivado de la Librería, y por tanto cae fuera del ámbito de esta Licencia.

La idea de la FSF es que de esta forma se ayudase a popularizar las librerías libres (maximizando así la libertad de los usuarios), al no imponer ninguna restricción al programa que las utilizase. Posteriormente, cambió de nombre (*Lesser* o «menor» en lugar de *Library*) y ha sido utilizada por aplicaciones como Mozilla y OpenOffice.

La LGPL contiene las mismas restricciones que la licencia GPL para el código que se encuentra cubierto por ella. Sin embargo, estas restricciones no se aplican a otro programa cuando este último no-LGPL se limita a compilar o enlazar (dinámica o estáticamente) con el programa bajo LGPL. Por tanto, la principal diferencia entre la GPL y la LGPL es que esta última puede enlazarse contra (en el caso de una librería «ser utilizada por») un programa no-GPL, que puede ser software libre o software privativo. A este tipo de licencias se las denomina «copyleft débil», frente al copyleft estricto.

Una particularidad de la LGPL es que permite que software bajo sus términos sea convertido en software GPL. Esta característica es útil si se desea reutilizar código de una librería LGPL en aplicaciones o librerías GPL (por ejemplo, con el fin de impedir que sean utilizadas por software privativo).

### 3.3. La licencia Affero

La **Affero GPL** (o simplemente **AGPL**) es una licencia libre de tipo copyleft publicada por la Free Software Foundation. Es una licencia derivada a partir de la GPL que contiene una cláusula para obligar a distribuir el código fuente modificado en aplicaciones que están diseñadas para ejecutarse en servicios de red.

¿Por qué es necesaria dicha cláusula? Como se sabe, la GPL (tanto la v2 como la v3) solo obliga a proporcionar el código modificado a todos aquellos a quienes les proporcionemos binarios de la aplicación. En cambio, si se modifica código GPL pero no se distribuye, no es obligatorio proporcionar el código modificado a nadie. A juicio de la FSF, esto es así porque es esencial dar la libertad a los usuarios de modificar el código y usarlo de forma privada sin verse forzados a

distribuir esos cambios si no distribuyen copias del software modificado. Sin embargo, si ese software se ejecuta en un servidor, difícilmente podría considerarse que se trata de uso privado, pero técnicamente no puede considerarse distribución, por lo que tampoco estará obligado a proporcionar los cambios. La Affero GPL se ha diseñado para cubrir este resquicio legal y obligar a que el software modificado en servicios de red se ponga también a disposición de quienes usan esos sistemas.

Se entenderá mejor con un ejemplo: supongamos que tenemos un sitio web basado en un CMS con licencia GPL, como Drupal o WordPress. Supongamos también que hemos hecho cambios importantes en el código GPL del CMS. Pues bien, los visitantes de nuestra página web no podrán exigirnos dichos cambios invocando la GPL, ya que a ellos no les estamos distribuyendo el software, sino que se limitan a hacer uso de él por la red. Para estos casos especiales se diseñó la Affero GPL y por esta razón la FSF recomienda su uso para servicios web y aplicaciones online, de modo que si el código fuente de esta aplicación es modificado, deberá proporcionarse a los usuarios de ese servicio el código fuente de la versión modificada que utilice. Un sitio muy popular que utiliza esta licencia es Menéame (la versión 1, no la AGPLv3). De ese modo, cualquier sitio web que utilice una versión modificada del código fuente de Menéame está obligado por la Affero a poner a disposición de la comunidad esos cambios.

Conviene reseñar que la cláusula de la AGPL solo se aplica si el software cubierto por dicha licencia está expresamente diseñado para ofrecer servicios de red (por ejemplo, servidores web, servidores de correo o servidores de juego online). En cambio, si la aplicación solo hace un uso circunstancial de un servicio de red (como el uso a través de SSH o de una sesión X remota), no entraría en esta categoría y no impone la obligatoriedad de facilitar el código fuente simplemente por usar dicha aplicación (es decir, en ese caso se comportaría como una licencia GPL convencional).

### **3.3.1. Compatibilidad**

La AGPL es copyleft estricta, por lo que, como sucede con este tipo de licencias, son generalmente incompatibles con otras licencias copyleft. Por ejemplo, anteriores versiones de la Affero GPL resultaban incompatibles con la GPL porque ambas obligaban a cubrir cualquier fusión bajo sus propios términos. Sin embargo, esta incompatibilidad se ha resuelto con la versión GPLv3 y la AGPLv3, ya que cada una de ellas incluye cláusulas (sección 13) que explícitamente permiten combinar código bajo ambas licencias y distribuirlo. La versión final de la AGPLv3 fue publicado en noviembre de 2007.

## **3.4. El caso de la EUPL**

La EUPL (European Union Public Licence) es una licencia de tipo copyleft auspiciada y aprobada por la Comisión Europea. Pretende adaptar al marco normativo comunitario los mismos principios que definen a la General Public License (GPL). Fue concebida para usarse preferentemente por las Administra-

ciones públicas, aunque también con un propósito general. Su objetivo es estar perfectamente adaptada a las legislaciones de cada uno de los 27 países miembros de la Unión Europea, mediante traducciones oficiales de la misma. Ese objetivo presupone implícitamente que la GPL pueda en alguna forma no ser compatible con la legislación de la Unión Europea, algo que a juicio de los expertos en licencias libres carece de todo fundamento (la GPL está redactada de forma genérica sin hacer alusión a ningún marco legislativo concreto y definiendo los términos de forma auto-contenida para facilitar su interpretación en el marco jurídico y con la terminología jurídica que corresponda en cada caso). La EUPL además abunda en la proliferación de licencias, generando nuevas incompatibilidades. Para sortear (parcialmente) esta objeción, la EUPL declara explícitamente en un anexo su compatibilidad con las siguientes licencias:

- GNU General Public License (GPL) v. 2
- Open Software License (OSL) v. 2.1, v. 3.0
- Common Public License v. 1.0
- Eclipse Public License v. 1.0
- CeCILL v. 2.0

No incluye, por tanto, a la GPLv3 como licencia compatible.

Su versión 1.0 contiene un serio inconveniente y es su aparente carácter revocable mediante versiones posteriores de la licencia. Se trata de su sección 13, que hace que ninguno de los derechos que recoge se puedan considerar permanentes. En concreto, dicha sección exige que cuando se publique una nueva versión de la EUPL ya no se pueda distribuir el código bajo una versión antigua de la licencia (sino que deberá acogerse necesariamente a la nueva):

**13. Varios** [...] La Comisión Europea podrá aprobar traducciones o nuevas versiones vinculantes de la presente licencia en la medida en que resulte necesario y razonable. Las nuevas versiones de la licencia se publicarán con un número de versión único. El licenciatarario quedará obligado por la nueva versión de la licencia en cuanto tenga conocimiento de su publicación.

Esto hace que ninguna de las libertades que la EUPL concede sean fidedignas, pues podrían modificarse en versiones posteriores invalidando las que hubiese antes, incluyendo la declaración explícita de compatibilidad con la GPLv2. Esta eventual expiración de las condiciones es algo insólito en las licencias libres ya que debilita las garantías acerca del código protegido por esta licencia por lo que podría llegar a considerarse una licencia no-libre. Para evitar esto último, la FSF ha solicitado a la Comisión Europea que modifique este apartado, algo que sus promotores están considerando para una futura versión de la licencia.

## 4. Licencias libres para documentación

### 4.1. Licencia de documentación libre de GNU

La Licencia de Documentación Libre de GNU (*GNU Free Documentation License* o simplemente GFDL) es una licencia copyleft diseñada por la Free Software Foundation (FSF) para publicar contenido libre. Fue diseñada originalmente para manuales, libros de texto y, en general, «obras funcionales» (es decir, no artísticas ni literarias) que acompañaran al software GNU. Sin embargo puede utilizarse en cualquier clase de obra basada en texto, sea cual sea sea su contenido. Es, por ejemplo, la licencia de todos los artículos de Wikipedia.

La GFDL distingue entre copias «transparentes» del documento (similar al código fuente del software) y copias «opacas» (similar a los binarios). Por ejemplo, un PDF sería una copia opaca, mientras que el fichero en LaTeX que lo genera sería la copia transparente. Sin embargo, establecer la definición de copia transparente no siempre es algo pacífico.

De forma análoga a la GPL, esta licencia asegura que el material cubierto por ella pueda ser usado, copiado, modificado y redistribuido siempre y cuando la redistribución se mantenga bajo los términos de esta misma licencia (GNU FDL). Al ser una licencia orientada a la publicación de textos, impone algunos requisitos especiales, por ejemplo que, en caso de copiarse o distribuirse en una cantidad superior a 100 ejemplares, deberá distribuirse también junto a un formato transparente que garantice futuras ediciones. También incluye la posibilidad de añadir secciones invariantes, lo cual ha sido siempre objeto de polémica (por ejemplo, durante años el Proyecto Debian consideró que, a causa de la secciones invariantes, los textos bajo GFDL no podían considerarse libres de acuerdo a las directrices de software libre de Debian –las DFSG–). Otro asunto polémico es la inclusión de cláusulas anti-DRM, destinadas a impedir medidas técnicas que obstruyan la edición o modificación de los textos, pero que pueden impedir también usos legítimos (como el cifrado del canal o del documento, o incluso la modificación de los permisos del sistema). No es compatible con la GPL.

#### 4.1.1. Compatibilidad

Gracias a un acuerdo con Creative Commons (CC), la FSF ha publicado una nueva versión de la licencia GFDL (1.3) que contempla un periodo transitorio (desde el 1 de noviembre de 2008 hasta el 1 de agosto de 2009) que permite fusionar e incluso relicenciar contenidos cubiertos por la GFDL en wikis y similares («Massive Multiauthor Collaboration Sites» o MMCS) a la licencia copyleft de CC (CC-by-sa 3.0). Una vez pasada esa «ventana de oportunidad», el sitio web (MMC Site) deberá decidir con qué licencia deja esos contenidos.

### 4.2. Creative Commons

**Creative Commons** (CC) no es una licencia, sino un conjunto de licencias elaboradas por la organización sin ánimo de lucro del mismo nombre fundada

en 2003 por el jurista Lawrence Lessig con la idea de aplicar los conceptos de ingeniería jurídica del software libre a otro tipo de obras (textos, fotografías, obras audiovisuales, etc.). CC ofrece un abanico de opciones para confeccionar la licencia a medida, que van desde cláusulas semirrestrictivas (no comercial o no derivada) hasta licencias equivalentes a las del software libre (es decir, que incluyan las cuatro libertades), como el copyleft (denominado *share-alike* o «compartir igual»). Incluso ofrecen la posibilidad de dedicar una obras al dominio público.

Además de ser actualmente las licencias más extendidas, la importancia de Creative Commons se debe a que permitió aplicar masivamente el concepto de licencia libre fuera del software. Al igual que sucede en el software libre, se devuelve al autor la iniciativa y el control sobre su obra (generalmente en manos de los intermediarios, como los editores y productores). Una diferencia importante de resaltar respecto a las licencias libres de software es que Creative Commons ha optado por un modelo de transposición (o adaptación) de las licencias a las jurisdicciones locales. A finales de 2008, cerca de 50 jurisdicciones distintas contaban con adaptaciones de sus licencias a las legislaciones locales.

Todas las licencias CC están formadas por tres partes, destinadas a la lectura por parte de legos, de abogados o de máquinas (agentes de software):

- *Commons Deed*: Es un resumen del texto legal legible por no-abogados con los iconos relevantes.
- *Legal Code*: El código legal completo en el que se basa la licencia que has escogido.
- *Digital Code*: El código digital, que puede leer la máquina y que sirve para que los motores de búsqueda y otras aplicaciones identifiquen el trabajo y sus condiciones de uso.

#### 4.2.1. Licencias Creative Commons

Las diferentes licencias de Creative Commons se basan en combinar distintas propiedades o cláusulas. Estas propiedades son:

- Reconocimiento (*Attribution*): Obliga a citar las fuentes de esos contenidos. El autor debe figurar en los créditos.
- No Comercial (*Non commercial*): Obliga a que los contenidos no puedan usarse comercialmente sin autorización.
- Sin Obra Derivada (*No Derivate Works*): Obliga a que esa obra sea distribuida inalterada, literalmente, sin cambios.
- Compartir Igual (*share-alike*): Obliga a que todas las obras derivadas se distribuyan siempre bajo la misma licencia del trabajo original. Es el equivalente a la cláusula copyleft del software.

Las combinaciones dan como resultado estas seis licencias:

1. Reconocimiento (by)
2. Reconocimiento + No comercial (by-nc)
3. Reconocimiento + Sin Obra Derivada (by-nd)
4. Reconocimiento + Compartir Igual (by-sa)
5. Reconocimiento + No comercial + Sin Obra Derivada (by-nc-nd)
6. Reconocimiento + No comercial + Compartir Igual (by-nc-sa)

Además de estas seis licencias, hay algunas otras licencias especiales:

- *Dedicatoria al Dominio Público*: (DP) poner una obra en el dominio público de forma anticipada.
- *Founder's Copyright*: el llamado copyright de los fundadores (por ser el tiempo que establecieron los padres de la patria estadounidense), de 14 años a 28 años; luego la obra pasa directamente al DP.
- *Sampling Plus*: Partes de la obra pueden copiarse y modificarse para cualquier propósito. La obra completa puede copiarse solo con fines no comerciales.
- *Noncommercial Sampling Plus*: La obra completa o partes de ella pueden copiarse y modificarse con fines no comerciales.

Han quedado obsoletas: Developing Nations, Sampling.

También existen alguna licencias de software, aunque son casos especiales y es recomendable utilizar las específicas del ámbito del software libre.

### 4.3. 'Freedom Defined': obras culturales libres

**Freedom Defined** no es una nueva licencia, sino una iniciativa para fijar de forma clara y no ambigua una definición de libre fuera del software. Se definen las «obras culturales libres» (*Free Cultural Works*) como «obras o expresiones que pueden ser libremente estudiadas, aplicadas, copiadas y/o modificadas por cualquiera, para cualquier propósito». Se trata por tanto de una adaptación estricta de las cuatro libertades clásicas del software libre a otro tipo de obras. El propósito de esta definición es acabar con la ambigüedad actual en el movimiento de la cultura libre (en el que cada cual hacía valer su definición ad hoc de lo que es «libre») y que un usuario, al ver el logo de Freedom Defined, sepa de forma unívoca que esa licencia es libre de acuerdo a la definición que ha acuñado el movimiento de software libre («las cuatro libertades»).

La iniciativa la lanzó un desarrollador de Debian, Benjamin *Mako* Hill, consensuando el borrador con la comunidad, con la Free Software Foundation y con Creative Commons. La iniciativa no cuestiona el derecho de cada autor a decidir los términos en que comparte su obra, o a incluir las cláusulas restrictivas que

estime oportuno: se trata solamente de identificar, sin ningún género de dudas, qué licencias (y obras) se ajustan a la definición de libre y cuáles no.

Creative Commons ha asumido esta definición y por ello, al escoger la licencia, indica mediante una leyenda y un logo bien visibles en el resumen de la licencia si se ajusta a la definición de libre de Freedom Defined. Estas son: CC-by, CC-by-sa y PD (Dominio Público).

## 5. Recomendaciones

En términos generales, antes de escoger una licencia, es necesario estudiar si encaja con el modelo empresarial (o institucional) elegido. Una vez que se ha realizado este estudio previo, habrá que adaptar el código, escoger un modelo de licencia, establecer un modelo de desarrollo, elaborar un estudio de costes para la puesta en marcha, etc.

Lo primero que hay que tener claro es que existen licencias públicas específicas para software y licencias específicas para documentación. No debe usarse una licencia que no fue concebida para un determinado tipo de obra (por ejemplo, no es recomendable usar la GPL para documentación ni las licencias CC para software).

### 5.1. Escoger una licencia para el software

Existen tres instituciones que han fijado los criterios para determinar qué licencias son libres: la FSF, la OSI y las DFSG (Debian). Los tres coinciden en considerar libres (u *open source*) a aquellas licencias que otorgan al público las siguientes cuatro libertades: libre uso, copia, modificación y redistribución (incluida la venta). Deben darse las cuatro condiciones simultáneamente. Si por ejemplo una licencia restringe el uso comercial (pero permite el libre uso educativo) NO es una licencia libre bajo ninguna definición por mucho que use el término *open* o *shared* en su márketing, o regale el software (llamado *freeware*) o permita ver su código fuente.

#### 5.1.1. Copyleft ‘versus’ no-copyleft

En un primer momento, el enorme abanico de licencias libres (u *open source*) que cumplen dichos requisitos puede causar confusión. Por ello conviene saber (y distinguir) que en realidad existen dos grandes grupos de licencias libres: las licencias copyleft (o recíprocas) y las licencias permisivas (o sin restricciones).

Debemos ser conscientes que las consecuencias de escoger un modelo u otro son importantes: con el modelo copyleft, obligamos a que cualquier competidor juegue con las mismas reglas y, si modifica y distribuye nuestro software, deba hacerlo manteniendo la licencia. De ese modo, nosotros podremos también beneficiarnos de esas mejoras. En general, es algo ventajoso para modelos centrados en el desarrollo de software.

Sin embargo, puede haber circunstancias (y modelos de negocio) donde sea interesante un esquema legal lo más laxo posible, sin ninguna restricción sobre

la distribución, por ejemplo porque deseamos promocionar el uso de nuestra tecnología, con independencia de lo que se haga con ella. Si esa es la prioridad, desearíamos evitar todo inconveniente para que se adopte. Google, por ejemplo, ha optado por una licencia permisiva (BSD) para el motor de su navegador Chrome. Si la competencia adopta ese motor, es un modo de ayudar a extender (y estandarizar) su tecnología. Otro modelo exitoso de licencias permisivas o minimalistas fueron la pila de protocolos TCP/IP: el hecho de que estuviesen bajo una licencia BSD (no-copyleft) facilitó que se estandarizara, pues pudo incluirse en sistemas operativos propietarios, como Windows o MacOS. Con una licencia copyleft esto no habría sido posible. La propia FSF reconoce que, en ciertos casos, este esquema no-copyleft es interesante estratégicamente para facilitar que una tecnología libre se adopte: es el caso de la licencia LGPL, que permite usar (fusionar o enlazar) su código con software no libre. Richard Stallman recordó hace años que la GPL (y el copyleft) no es un fin en sí mismo, sino un dispositivo para garantizar la libertad de los usuarios y que, en ciertos casos (como en el del formato libre vorbis frente al patentado mp3) una licencia permisiva estilo BSD podría ser más beneficiosa para la libertad de los usuarios que una de tipo copyleft.

En términos generales, y salvo en esos casos donde el aspecto pragmático se imponga por las razones expuestas más arriba, para el autor de un programa es preferible utilizar una licencia GPL, ya que garantiza que el autor no perderá el control sobre las modificaciones y mejoras de su código. Pero ¿qué versión de la GPL usar? La GPLv3, recientemente actualizada, está preparada para afrontar todos los desafíos que el actual marco tecno-jurídico impone (como las patentes, el DRM y la *tivoización*) y que estaba dejando algunas licencias libres en papel mojado (al impedir ejercer con métodos tecnológicos algunas de las libertades que la GPL concede). La GPLv3 protege de forma consistente a los autores de software y a la comunidad de usuarios de todas esas amenazas.

La elección más adecuada puede variar también dependiendo de si somos creadores del software o vamos a utilizar, adaptar o modificar un software escrito por terceros. En el caso de que usemos software de terceros, la GPL nos impone algunas condiciones que pueden interesarnos o no, pero que debemos conocer (básicamente, la obligación de poner los cambios a disposición del público si optamos por redistribuirlo). Las licencias estilo BSD/Apache, en cambio, no nos imponen límite ninguno, lo cual podría ser interesante ante determinados requisitos del cliente (que se niegue, por ejemplo, a hacer públicas las modificaciones).

## 5.2. Escoger una licencia para documentación

Debe usarse siempre una licencia específica para documentación, no una de software. Otro punto al que hay que atender es a la definición de libre: no todas las licencias Creative Commons, ni todas las que se auto-proclaman *open content* u *open access* son libres de acuerdo a la definición de libre de Freedom defined (respaldada por la FSF y por la propia CC).



Entre las licencias libres de documentación disponemos también de licencias permisivas y licencias robistas (copyleft), que funcionan de forma análoga a las de software: mientras las licencias permisivas permiten distribuir versiones modificadas con cualquier licencia (incluida una privativa), las licencias copyleft obligan a mantener la licencia original en caso de que se decida redistribuir una copia modificada.

Entre las licencias de tipo copyleft tenemos dos opciones principales: la GNU FDL y la CC-by-sa. Ambas son similares en cuanto a su espíritu (mantener las libertades en las versiones derivadas), pero difieren en su plasmación jurídica. Además, son mutuamente incompatibles, esto es, no es posible fusionar material procedente de las dos licencias.

La GNU FDL está pensada para documentación y para obras impresas (pese a que se usa mucho en wikis, por influencia de Wikipedia) e incluye algunas restricciones problemáticas, como las cláusulas anti-DRM (que puede excluir usos legítimos) o la posibilidad de incluir cubiertas y partes invariantes (que no pueden modificarse). También exige que, junto a la obra, se distribuya el texto completo de la licencia. Su redacción es compleja y sujeta a interpretaciones, e incluso inconsistencias, como por ejemplo en lo referente a la definición de copia transparente. Su ventaja principal es que es usada por Wikipedia en sus artículos, lo que facilita la fusión con materiales procedentes de la enciclopedia libre por excelencia, caso de que el nuevo trabajo vaya a basarse en ella.

Por su parte, la CC-by-sa es una licencia más sencilla, de propósito general (no está escrita para ningún fin en particular), sin restricciones complicadas de interpretar o difíciles de cumplir, y que prevé la compatibilidad con otras licencias similares o equivalentes de Creative Commons. Tampoco exige incluir el texto completo de la licencia, sino solamente declarar bajo qué licencia está publicada la obra. Las licencias CC son mucho más conocidas y, gracias a su gran difusión, existe mucho material reutilizable bajo estas licencias, aunque habrá que estar atentos porque no todas las licencias de Creative Commons son compatibles con la CC-by-sa y esto puede causar confusiones.

Con la versión de la GFDL 1.3, publicada en octubre de 2008, se ha abierto un periodo transitorio hasta el 1 de agosto de 2009 para poder relicenciar material con licencia GFDL a una licencia CC-by-sa. Esto permitirá, por ejemplo que, si la comunidad de Wikipedia lo decide (o la de cualquier otro "Sitio de Colaboración Masiva Multiautor", o MMC Site, que usase la GFDL 1.2), se relicencien los artículos publicados en el sitio bajo GFDL con una licencia CC-by-sa.

### **5.2.1. Licencias permisivas para documentación**

En cuanto a las licencias permisivas de documentación, podemos destacar la CC-by y la FreeBSD Documentation License. La primera es la licencia libre permisiva (no-copyleft) de Creative Commons. Por su parte, la Licencia BSD de Documentación es análoga a la licencia BSD de software pero de propósito general para facilitar su uso con documentación técnica, tanto como código fuente ((SGML DocBook) como en forma compilada (transformado a otros DTDS, convertido a PDF, PostScript, RTF y otros formatos).

### 5.2.2. Problemas con las licencias CC

Las licencias de Creative Commons tiene serios problemas para cumplir con las directrices de Debian (DFSG). Los motivos se deben a una ambigua, excesiva (y algo descuidada) exigencia en relación al reconocimiento de la autoría. El autor original puede exigir la retirada de toda mención suya en las obras derivadas. O, al contrario, exigir su inclusión para toda la obra en obras derivadas, pese a que su participación sea parcial. En una interpretación pesimista, esto podría requerir situaciones absurdas o condiciones imposibles de cumplir. Por ejemplo, en una recopilación de artículos de diferentes autores, con la autoría reconocida en cada capítulo, el nombre del titular tendría que ser listado en cada capítulo, aunque no hubiese contribuido a él. Otro ejemplo es el caso de una autobiografía (de alguien llamada Alice) en la cual se incluyese un contenido cubierto por CC-by (por ejemplo, la letra de una canción de un tal Bob). En ese caso podría entenderse que el reconocimiento debe dársele a Bob sobre la obra entera (y la autobiografía pasaría ser «la autobiografía de Alice y Bob»). En las últimas versiones se ha intentado pulir estos problemas, refiriéndose a «usos razonables».

Otro problema con las Directrices de Debian es la cláusula anti-DRM que incluyen las licencias CC-by y CC-by-sa y que podría prohibir la distribución privada o mediante un servidor con control de acceso (un firewall o una capa criptográfica como SSL). Sin embargo, hay que tener en cuenta que este tipo de cláusula fue admitida en 2006 mediante votación para el caso de la GFDL. Por último, la estricta prohibición de los logos y marcas registradas de CC para cualquier otro uso que no sea indicar que la obra está licenciada con CC, podría evitar, en algunas interpretaciones, distribuir y modificar obras bajo estas licencias.

Todas estas razones hacen desaconsejable para Debian el uso de licencias CC para la documentación técnica. En su lugar, Debian aconseja el uso para la documentación de software de la GFDL (si se desea un esquema copyleft) o la FreeBSD Document License (si se desea un esquema permisivo no-copyleft).

## 6. Conclusiones

Es importante tener en cuenta varias cosas, a la hora de elegir una licencia. En primer lugar, usaremos licencias previstas para documentación o para código (según se trate de documentación o del código propiamente dicho).

Lo segundo que deberemos tener en cuenta es si deseamos un esquema de permisos de tipo copyleft (obliga a que si se distribuyen versiones modificadas, se haga siempre con la misma licencia) o permisivo (no se impone ningún requisito sobre las obras derivadas).

Lo tercero que debemos tener en cuenta es la compatibilidad, especialmente si estamos pensando combinar (o enlazar) software con distintas licencias libres. Esto puede afectar por ejemplo a librerías, módulos o plugins que se compilan o dependen de un *core* con distinta licencia. Si vamos a participar de forma lateral

en un gran proyecto de software libre que ya usa una licencia concreta (Apache, Mozilla, Perl, BSD, etc.), seguramente sea preferible mantener la licencia con la que se trabaja en el proyecto principal, para evitarnos sorpresas y asegurarnos de que no habrá incompatibilidades inesperadas entre licencias.

Por último, es recomendable evitar a toda costa la *proliferación de licencias*, ya que un mayor número de licencias no suele aportar nada nuevo pero incrementa exponencialmente las posibles combinaciones e interacciones, algunas de ellas tal vez incompatibles. Además de incompatibilidades, esta situación puede provocar inseguridad jurídica (por ejemplo, precisar del concurso de abogados para aclarar las interacciones) y favorecer el FUD.<sup>4</sup> La aparición de nuevas licencias vinculadas a un proyecto o empresa (llamadas *vanity licenses*, porque normalmente tienen fines únicamente publicitarios o no aportan nada que no recojan ya otras licencias) favorece esta proliferación e incluso la posibilidad de cometer errores en la redacción de las propias licencias. La EUPL de la Unión Europea (en casi todo análoga a la GPL) es un ejemplo de *vanity license*.

Copyright © 2008 Miquel Vidal – GSyC/Libresoft  
Este artículo se publica bajo la licencia Creative Commons  
Reconocimiento-Compartir bajo la misma licencia 3.0 España.

---

<sup>4</sup>*Fear, Uncertainty and Doubt*, «miedo, incertidumbre y duda». Por ejemplo, deducir que por el hecho de que no exista una licencia oficialmente traducida en un país, «no es válida» en ese país.