

# Licencias de software libre

Malcolm Bain  
Manuel Gallego  
Manuel Martínez Ribas  
Judit Rius

P08/M2114/00347



Universitat Oberta  
de Catalunya

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



# Índice

<b>Introducción.....</b>	<b>5</b>
<b>Objetivos.....</b>	<b>6</b>
<b>1. Aspectos generales de las licencias libres.....</b>	<b>7</b>
1.1. Libertad en el software .....	7
1.2. Software libre con <i>copyleft</i> .....	9
1.3. Software "de fuentes abiertas": la <i>Open Source Definition</i> .....	11
<b>2. Estudio particular de las licencias de software libre.....</b>	<b>16</b>
2.1. Las licencias permisivas: sin <i>copyleft</i> .....	17
2.1.1. La licencia Berkeley Software Distribution (BSD) y similares .....	17
2.1.2. Las licencias Apache (ASL) .....	20
2.1.3. Otras licencias permisivas .....	22
2.2. Las licencias con <i>copyleft</i> robusto .....	23
2.2.1. La Licencia Pública General GNU, versión 2.0 (GPLv2) .....	23
2.2.3. La Common Public License y la Eclipse Public License .....	37
2.2.4. Otras licencias con <i>copyleft</i> robusto .....	39
2.3. Las licencias con <i>copyleft</i> "suave" o "híbridas" .....	40
2.3.1. La Licencia Pública General Menor o de Bibliotecas GNU (LGPL) .....	40
2.3.2. La Mozilla Public License .....	43
2.3.3. La Open Source License (OSL) .....	48
2.3.4. Otras licencias con <i>copyleft</i> "suave" o "híbridas" .....	50
<b>3. Otras licencias de tipo "libre".....</b>	<b>52</b>
3.1. El auge y la caída de las licencias de software "pseudolibres" .....	52
3.1.1. La Sun Community Source License (SCSL) .....	52
3.1.2. Microsoft Shared Source Initiative (MSSI) .....	54
3.2. Las licencias de documentación libre .....	55
3.2.1. La Licencia de Documentación Libre de GNU (GFDL) .....	55
3.2.2. La iniciativa Creative Commons .....	56
3.3. Licencias de tipo <i>freeware</i> y <i>shareware</i> .....	59
<b>4. Conclusiones.....</b>	<b>60</b>



## Introducción

En este módulo de aprendizaje, analizaremos con detenimiento el elemento que constituye la base jurídica del movimiento de software libre y que diferencia a éste del software tradicional o propietario: **la licencia de software libre**.

Durante el estudio de los conceptos fundamentales, en los módulos anteriores ya hemos visto algunos aspectos relevantes y las cláusulas más importantes de las licencias de software en general. Aquí enfocaremos las modalidades principales de las licencias libres y sus efectos legales. Esto nos permitirá, en el módulo 7, comentar algunos otros temas importantes y correlativos que surgen como consecuencia de la aplicación de las mismas.

En el primer apartado, comentaremos el concepto de *libertad* relativo al software –la definición de software libre (*free software*) de la Free Software Foundation (FSF) la definición de Open Source Initiative (OSI)– tal como se plasma en una licencia y sus diferentes interpretaciones en el mundo del software libre.

Luego, en la segunda parte, analizaremos algunas de las licencias de software libre más frecuentemente usadas, para entender mejor su funcionamiento y sus características. Las clasificaremos en tres grupos: licencias permisivas, licencias con *copyleft* robusto y licencias híbridas o con *copyleft* "suave". Para cada una de ellas presentaremos una breve historia y haremos un análisis legal, destacando lo que permiten y lo que prohíben, sus condiciones y sus restricciones. Reseñaremos sus características particulares y su compatibilidad con otras licencias, especialmente con la GNU-GPL.

En el tercer apartado, comentaremos otras licencias relacionadas con el software libre, como las de documentación libre y otras que intentan acercarse a la categoría de libre, como la Sun Community y la Microsoft Shared Source. Cerraremos el módulo con una breve nota final sobre las licencias de tipo *free-ware* y *shareware*.

## Objetivos

Con el aprendizaje de este módulo, los estudiantes podréis alcanzar los siguientes objetivos:

1. Entender que hay un gran abanico de licencias libres, que va desde las licencias permisivas hasta las licencias con *copyleft* robusto.
2. Conocer las licencias de software libre de mayor uso, enfocando las licencias BSD, GPL, LGPL y MPL, y comprender sus particularidades.
3. Entender las diferencias principales entre las licencias libres, propietarias y otras licencias "casi" libres, como la Sun Community y la Microsoft Shared Source.
4. Conocer algunas licencias para otros recursos libres, como la documentación de software o las publicaciones en Internet.

## 1. Aspectos generales de las licencias libres

En los módulos anteriores, hemos presentado las licencias de software en general y su ajuste al marco legal de la propiedad intelectual e industrial. En este módulo, presentamos las licencias de software libre.

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

Tal y como hemos visto en anteriores, la diferencia entre el software libre y el software propietario reside en los derechos y obligaciones que se especifican en la licencia. Aquéllos otorgados por las licencias de software libre ofrecen una libertad amplia para explotar el software, en cuanto al uso, modificación y distribución del mismo, y suelen ser directamente opuestos a los otorgados y reservados por una licencia de software propietaria ("licencia propietaria"). En este módulo, analizaremos con detalle los derechos y las obligaciones de las licencias libres.

### Terminología de las licencias de software libre

La nomenclatura relativa a las licencias de software libre en sentido amplio que vamos a usar es la misma que hemos presentado en el módulo 1:

- **Software libre y licencia libre.** Seguiremos la práctica de la FSF, que usa el término *software libre* para cualquier software distribuido bajo una licencia que respete las cuatro libertades antes mencionadas.
- **Software abierto y licencia abierta.** Es el software cuya licencia cumple con las directrices de la definición de software de código fuente abierto (*open software definition*, OSD). Son licencias libres.
- **Software copyleft y "licencia con copyleft":** aplicaciones y licencias que se distribuyen con una cláusula de *copyleft* robusto, como la GPL, o con una cláusula de *copyleft* suave, como la LGPL o la MPL.
- **Software y licencia no-libre, propietario y privativa.** Son aplicaciones distribuidas bajo licencias que no son libres.

### 1.1. Libertad en el software

Como ya se ha comentado este módulo, la palabra *free* ('libre', en inglés) tiene dos sentidos: *libertad* y *gratuidad*. El uso de la palabra *free* en relación con el software no implica que el titular o el proveedor del software libre otorgue o

distribuya el software gratuitamente (aunque lo puede hacer), sino que **lo distribuye bajo una licencia que permite a los usuarios aprovecharlo libremente en cuanto a su uso, reproducción, modificación y distribución.**

Ya hemos visto en el módulo 1 la definición que nos da la Free Software Foundation (FSF) del concepto de *libertad del software*, adoptada y aceptada por toda la comunidad:

- la libertad de usar el programa con cualquier propósito (libertad 0);
- la libertad de estudiar el funcionamiento del programa y adaptarlo a las propias necesidades (libertad 1), para lo cual el acceso al código fuente es una condición previa;
- la libertad de distribuir copias (libertad 2);
- la libertad de mejorar el programa y publicar cualquier mejora, de modo que toda la comunidad se beneficie (libertad 3). El acceso al código fuente es un requisito previo para esto.

Recalquemos que estas libertades de uso corresponden a los derechos exclusivos de explotación reservados a los titulares de derechos de autor por las leyes de propiedad intelectual que hemos estudiado en el módulo 2:

- Libertad 0: el derecho de uso (no es un derecho exclusivo, pero la licencia libre permite un uso sin restricciones ni discriminación).
- Libertad 1: el derecho de modificación.
- Libertad 2: los derechos de copia y distribución.
- Libertad 3: los derechos de modificación y distribución/publicación de las obras derivadas.

Así, cualquier licencia de software libre debe ceder a los usuarios estos derechos.

### Ejemplo

Por ejemplo, la licencia MIT establece que "permission is hereby granted [...] to any person obtaining a copy of this software [...] to deal in the software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the software [...]", mientras que la licencia BSD dice que "redistribution and use in source and binary forms, with or without modification, are permitted [...]".

No obstante, es importante resaltar que **no todas las licencias libres son similares**. Parafraseando (sin abusar) una célebre expresión de George Orwell, aunque todas sean libres, algunas ofrecen "más libertad" que otras. Todas deben garantizar, al menos, las cuatro libertades antes mencionadas, pero las



disposiciones contenidas en estas licencias –las obligaciones asociadas con los derechos– pueden variar de manera significativa. El abanico de posibilidades se extiende desde las licencias permisivas, que contienen unas obligaciones mínimas (que obligan únicamente a mantener el aviso de autoría y la negación de garantías y de responsabilidad –el *disclaimer*), hasta el "máximo" (en cierto sentido) de las licencias con *copyleft* robusto –la cláusula *copyleft* de la GPL–, que obligan a distribuir cualquier modificación y obra derivada bajo la misma licencia libre.

El siguiente esquema ilustra este abanico de posibilidades:

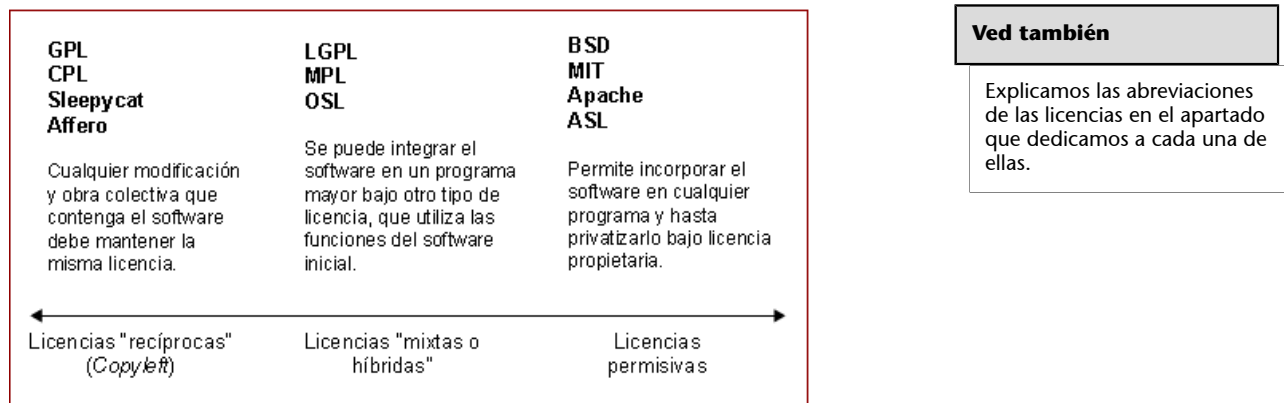


Figura 1. Abanico de licencias libres

Consideramos que las distintas licencias garantizan diferentes tipos de libertad:

- La BSD, por ejemplo, otorga más libertad a los desarrolladores, porque éstos pueden incorporar y distribuir implementaciones de "código BSD" bajo licencias tanto libres como propietarias.
- La GPL transmite más libertad a los usuarios finales, porque éstos siempre recibirán aplicaciones bajo una licencia libre y con código fuente disponible.
- La LGPL y la MPL buscan un equilibrio entre estas dos libertades, manteniendo el *copyleft* para el código inicial pero permitiendo su incorporación en obras bajo otras licencias.

## 1.2. Software libre con *copyleft*

Para efectuar este estudio, es primordial considerar el concepto de *libertad* de la Free Software Foundation y su implementación particular mediante la General Public License (GPL). No solamente porque ésta es la licencia más usada en el mundo del software libre (con un 70% de los proyectos libres en Sourceforge),

ni porque es la precursora de muchas otras licencias libres actuales (aunque no de todas; la BSD la precede), sino porque la filosofía de libertad de la FSF ha sido la base y uno de los elementos más destacados del movimiento libre.

El objetivo de R. Stallman y la FSF, muchas veces repetido, es **garantizar la libertad** de los usuarios del software y **mantener esa libertad** en relación con redistribuciones del software y de obras derivadas del software originalmente libre. Aunque la garantía de las cuatro libertades fundamentales enunciadas por la FSF es suficiente para que una licencia de software pueda considerarse "libre", su mero cumplimiento –que es lo que hacen, por ejemplo, las licencias BSD, Apache o MIT– no garantiza lograr estos objetivos específicos de la FSF. Por lo tanto, la licencia que ha redactado y que principalmente recomienda la FSF, la GPL (y de manera reducida, la LGPL), incluye una condición especial con tres elementos fundamentales:

- una obligación de usar la misma licencia para redistribuciones posteriores del software (y obras derivadas); y
- la obligación de proporcionar el código fuente del software en cualquier redistribución del programa (o por lo menos, acceso al mismo); y
- una prohibición de agregar cualquier restricción adicional (a las de la licencia original) sobre dichas redistribuciones.

Esta condición particular, que se llama *copyleft*, hace imposible legalmente "capturar" el software libre, modificarlo y privatizarlo. Por lo tanto, el *pool* o la cantidad de software con *copyleft* disponible no puede hacer más que aumentar a medida que los desarrolladores crean nuevas aplicaciones sobre la base del software con *copyleft*.

En el apartado 2.3, analizaremos con detalle el mecanismo legal por el que la licencia GNU GPL (que llamaremos simplemente "la GPL") implementa esta filosofía de manera "robusta" y por qué ha sido llamada "virus" por algunos y "herencia" o "reciprocidad" por otros. Este mecanismo también se usa en otras licencias, entre las cuales encontramos la Lesser GPL (LGPL), la Mozilla Public License (MPL), la Common Public License (CPL) y la Open Source License (OSL), que comentaremos más adelante. La mayoría de ellas se caracterizan por un *copyleft* "suave", ya que solamente afecta al software original (y a obras derivadas de éste), y no a las obras que usen o contengan el software con estas licencias.

Es importante observar que el *copyleft* no afecta a los derechos de uso del licenciatario original (por ejemplo, un simple usuario), sino que restringe las libertades relativas a la distribución posterior del software *copyleft*, con o sin modificaciones (o íntimamente incorporado en otras aplicaciones).

Comprender esto permite entender por qué una cláusula *copyleft* no afecta el uso comercial de aplicaciones con *copyleft* en las organizaciones privadas o públicas, porque dichas organizaciones son normalmente usuarios finales.

También hay que destacar que los impactos legales de las cláusulas de *copyleft* han provocado mucha preocupación en el mundo del software en general. Sobre todo, se ha temido que la vinculación o la incorporación de código con la GPL en otro programa afecte al uso o la distribución de la aplicación resultante, o que el uso de software con GPL (por ejemplo, GNU/Linux) impidiese el uso de otras aplicaciones propietarias. Estas dudas, que son más bien mitos, se considerarán en el módulo 7.

### 1.3. Software "de fuentes abiertas": la *Open Source Definition*

Como ya hemos señalado, hubo una cierta escisión en el movimiento de software libre en 1998, que en realidad fue la concreción de una división que venía produciéndose durante la década de 1990. La división se basaba no en los conceptos básicos del software libre –que son compartidos–, sino en la manera de presentarlos. Esta división se formalizó con la creación de la **Iniciativa de Código Fuente Abierto** (Open Source Initiative, OSI) por parte de Eric Raymond y Bruce Perens, entre otros, en 1998.

La OSI trata de reconciliar las libertades del software libre (en general) con las necesidades comerciales de las empresas implicadas en la creación, la distribución y el uso de software libre. De esta manera, el software de código abierto mantiene las libertades fundamentales del movimiento libre (reproducción, transformación, distribución, acceso al código fuente), pero no el nombre de "software libre". Éste es sustituido por "software de código fuente abierto" (*open source*), pues la OSI considera que el énfasis excesivo que hace la FSF en razones morales o éticas para la libertad del software puede provocar reacciones negativas en la mentalidad empresarial, y que es más beneficioso promocionar el software libre por sus méritos técnicos. Por otro lado, la FSF no está de acuerdo con el uso del término *open source* para referirse al software libre, justamente porque hace que pierda esta dimensión ética referida a la libertad.

Como resultado de esta iniciativa, se estableció la **Definición de Software de Fuentes Abiertas** (*Open Source Definition*), para la cual usaremos el acrónimo inglés OSD.

#### Ved también

Podéis ver en el módulo 1 un comentario sobre la historia de la FSF y la OSI.

## Los objetivos de la OSD)

La OSD fue diseñada para establecer una declaración abierta y comprensiva de los principios del movimiento de software abierto y un sistema de clasificar y "certificar" la multitud de licencias libres que existen. Se argumenta que, al establecer estándares de esta manera, la definición permite a desarrolladores, usuarios, organizaciones comerciales y a la Administración pública a entender mejor el movimiento de software libre y a respetar más sus principios.

Es importante considerar que la OSD no es una licencia, ni un modelo de licencia, sino que establece **directrices para la clasificación de licencias** relativas a aplicaciones y productos de software en sus diversas formas (componentes, programas, distribuciones completas). Es más, la OSI ha elaborado una marca de certificación, la marca "*OSI Certified*", que es una manera ostensible de indicar que una licencia cumple con la OSD. La marca sirve también para diferenciar el término general "*Open Source*", que no tiene un uso suficientemente definido para garantizar esta conformidad

### Ved también

Hay varios textos de la OSI en el apartado "OSI", en la bibliografía.

La OSD nos provee una segunda herramienta de análisis y clasificación de las licencias de software libre, y llamaremos también "abiertas" a aquellas licencias que cumplan sus directrices.

Es importante resaltar que las licencias abiertas son licencias libres, y viceversa. La diferencia entre la OSI y la FSF (como instituciones) de perspectiva (marketing, filosofía subyacente, etc.) y no de principios, que son compartidos entre las dos entidades. En realidad, las discrepancias no son legales sino de postura -la OSI enfatizando más la necesidad de acceso al código fuente, la FSF da más importancia a la ética o filosofía de "libertad". Es evidente que la GPLv2 y la GPLv3 son licencias "abiertas", que cumplen con la OSD: de hecho, están certificadas por la OSI.

## El origen textual

La OSD surge de las Directrices Debian de Software Libre (*Debian Free Software Guidelines*), adaptadas en 1998 –básicamente por la eliminación de las referencias a Debian.

La definición de la OSI enfatiza los cuatro elementos fundamentales del movimiento de software libre, expresados en las cuatro libertades enumeradas por la FSF. Asimismo, la disponibilidad y el acceso al código fuente es fundamental: la palabra *open* estaría mejor traducido por 'disponible', 'visible' o 'legible', y deberíamos hablar de licencias de software con código fuente disponible.

La OSD ha sido modificada varias veces desde su primera versión, la 1.0. La versión actual, que es la 1.9, tiene diez criterios.

A continuación, reproducimos la lista de directrices de la OSD en una versión no oficial en castellano, junto con algunos comentarios que indican la razón y los efectos de cada una de dichas directrices.

#### Web recomendada

Hallaréis la OSD en el sitio web de la OSI:  
[www.opensource.org/docs/osd](http://www.opensource.org/docs/osd).

	Directriz y comentario
1	<b>Redistribución libre.</b> La licencia no deberá impedir la venta o el ofrecimiento del software a ninguna parte como un componente de una distribución de software agregado que contenga programas de varias fuentes distintas. La licencia no deberá requerir el pago de los derechos de autor u otra tasa por dicha venta.
	Garantiza el derecho de distribución. Implica que un usuario puede copiar, vender o distribuir gratuitamente el software abierto. (Sin embargo, ¡no indica que el software <b>tiene que ser</b> distribuido gratuitamente!).
2	<b>Código fuente.</b> El programa ha de incluir el código fuente y ha de permitir la distribución tanto en código fuente como en forma compilada. Si alguna forma de un producto no es distribuida con el código fuente, tiene que haber un medio bien publicado de obtener el código fuente que no exceda de un coste razonable de reproducción, preferentemente una descarga por Internet sin cargo. El código fuente tiene que ser la forma preferida en la cual un programador modificaría el programa. No está permitido el código fuente deliberadamente escondido. Las formas intermedias, tales como la salida de un preprocesador o un traductor, tampoco están permitidas.
	El software debe incluir el código fuente y la licencia debe permitir que se realicen distribuciones de código binario, siempre y cuando la forma de obtener el código fuente esté indicada claramente. El código fuente es necesario para que los destinatarios tengan la libertad de modificar el programa.
3	<b>Obras derivadas.</b> La licencia tiene que permitir modificaciones y obras derivadas, así como su distribución en los mismos términos de la licencia del software original.
	Garantiza el derecho de modificación y de distribución de las modificaciones y las obras derivadas. Además, se debe <b>permitir</b> que estas modificaciones sean distribuidas bajo los mismos términos que la licencia original del software. Esto no implica que la obra derivada <b>deba</b> distribuirse bajo estos términos. La BSD, por ejemplo, permite modificar el software original y comercializar la modificación en formato binario solamente, a diferencia de la licencia GPL (que es compatible con esta directriz), que obliga a mantener las obras derivadas bajo la GPL.
4	<b>Integridad del código fuente del autor.</b> La licencia puede impedir que el código fuente sea distribuido en forma modificada <i>solamente</i> si permite la distribución de "archivos en forma de parche" con el código fuente con el objetivo de modificar el programa en el tiempo de construcción. La licencia tiene que permitir explícitamente la distribución del software construido a partir del código fuente modificado y puede requerir que las obras derivadas tengan un nombre o un número de versión distinto al del software original.
	Garantiza el derecho de distribución de las obras derivadas y permite mantener la integridad de cada componente de una aplicación original y proteger la autoría original. Una manera permitida de mantener esta separación es la de obligar a distribuir una obra modificada, en primer lugar, como la aplicación original, y en segundo lugar, como un "parche" que modifica el software original en el momento de la instalación o construcción ( <i>build time</i> ). Otro sistema para proteger la autoría es un control de la nomenclatura de las versiones. Se permiten estas restricciones particulares sobre la distribución de los programas derivados para que el autor original pueda proteger su reputación ante posibles problemas en el funcionamiento del software causados por una modificación o ante una diferencia de "calidad" en el software modificado.
5	<b>La no discriminación con respecto a las personas o grupos.</b> La licencia no debe discriminar a ninguna persona o grupo de personas.
	Garantiza un uso más amplio del software abierto en cuanto a las personas (usuarios). No se puede restringir el uso por motivos políticos, religiosos, etc. Asimismo, hay que notar que si el marco legal puede imponer restricciones de uso (como por ejemplo la criptografía en Estados Unidos), la licencia misma no puede incorporarlas.
6	<b>La no discriminación con respecto a los sectores de actividad.</b> La licencia no debe restringir a nadie que haga uso del programa en un sector de actividad específico. Por ejemplo, no puede impedir que el programa sea usado en un negocio o para una investigación genética.
	Garantiza un uso más amplio del software abierto en cuanto a las áreas de uso: no se pueden restringir los usos privados, comerciales, educativos o militares. Así pues, se amplían al máximo los usos del software.

	Directriz y comentario
7	<b>Distribución de la licencia. Los derechos adjuntos al programa se han de aplicar a todos aquellos que reciban el programa, sin que haya la necesidad de ejecutar una licencia adicional para estas partes.</b>
	La licencia abierta no debe obligar a los usuarios a firmar un "consentimiento", ni por la licencia ni por ninguna cláusula o pacto adicional (una carta de confidencialidad, por ejemplo). Asimismo, la falta de procedimientos adicionales es necesaria para permitir a licenciarios segundos y terceros aprovechar los derechos especificados en la licencia y quedar vinculados por las obligaciones correspondientes. Ya hemos comentado en los módulos anteriores que este mecanismo puede entrar en conflicto con el marco legal de los derechos de autor y de contrato en relación con la necesidad de prestar consentimiento por parte del licenciario.
8	<b>La licencia no tiene que ser específica de un producto. Los derechos adjuntos al programa no tienen que depender de que el programa forme parte de una distribución particular de software. Si el programa es extraído de esa distribución y es usado o distribuido de acuerdo con los términos de la licencia, todas las partes a las que el programa sea redistribuido deben tener los mismos derechos que son garantizados en conjunto con la distribución original del software.</b>
	Garantiza un uso más amplio del software abierto en cuanto a la forma de distribución utilizada. Los derechos que otorga la licencia no deben ser diferentes para un software incluido en una distribución original que para el mismo software redistribuido de manera diferente o separada. Es decir, no se puede restringir una versión de Linux a su uso con un paquete determinado de distribución. La versión de Linux queda abierta, incluso si se separa del paquete de distribución original.
9	<b>La licencia no debe limitar otro software. La licencia no debe imponer restricciones sobre otro software que se distribuya junto con el software licenciado. Por ejemplo, la licencia no tiene que insistir en que todos los otros programas distribuidos en el mismo medio tengan que ser software de código fuente abierto.</b>
	Garantiza una forma más libre de distribución: la licencia no debe poner límites sobre el software que se distribuya con el mismo. Por ejemplo, la licencia no debe obligar a que todos los programas distribuidos conjuntamente con el software en cuestión sean libres o abiertos. Como consecuencia, se puede distribuir software GPL, BSD y propietario en un mismo CD. Es importante distinguir la diferencia entre: 1) agregar aplicaciones lógicamente separadas sobre un mismo soporte –la flexibilidad de distribución garantizada por esta directriz–; y 2) "agregar" aplicaciones no lógicamente separadas, es decir, reunidas para crear una obra derivada. Esta directriz no trata de la creación de obras derivadas.
10	<b>La licencia debe ser neutra respecto de la tecnología. Ninguna disposición de la licencia debe predicar una tecnología o un tipo de interfaz particular.</b>
	La aceptación de la licencia no debe depender del uso de una tecnología o interfaz. Esta directriz se refiere a licencias que pueden obligar a usar determinados sistemas tecnológicos para prestar el consentimiento a la licencia (por ejemplo, el uso de licencias <i>click-wrap</i> ). Hay otras maneras de distribuir el código y otras interfaces posibles (FTP, interfaces no gráficas) que pueden ser independientes o incompatibles con la especificación de una tecnología en particular.

Para ilustrar estas directrices de manera simplificada, el siguiente diagrama esquemático (figura 2) resume los derechos que hay que garantizar al usuario-licenciario con una licencia abierta:

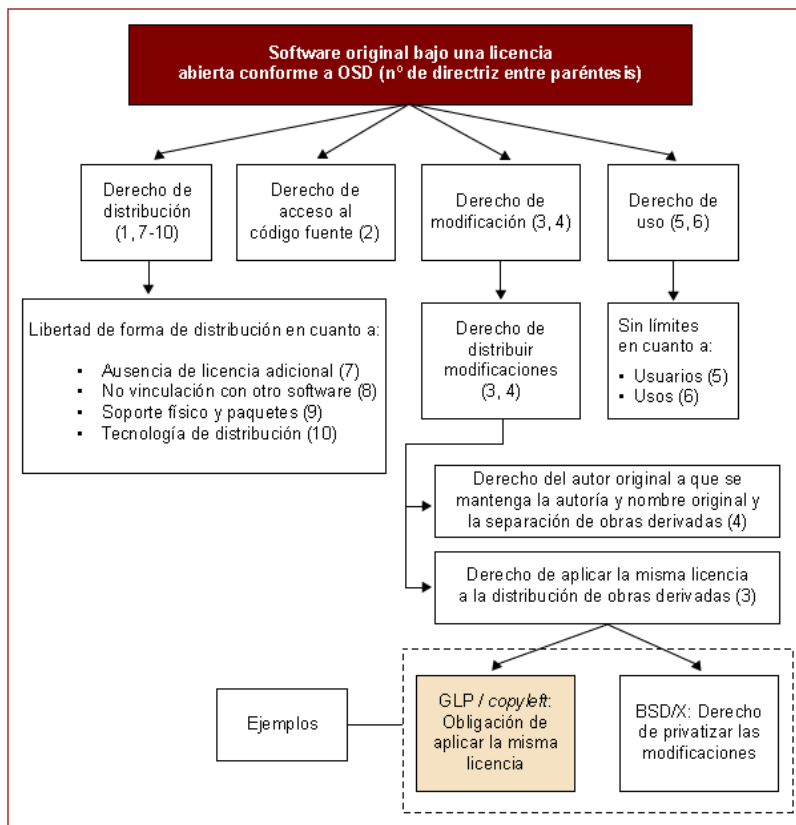


Figura 2. Diagrama esquemático de los derechos mínimos otorgados por una licencia abierta OSD

### Aplicación de la OSD

Un ejemplo interesante de la aplicación OSD surge en el caso de KDE, Qt y la empresa Troll tech. KDE es una interfaz gráfica de escritorio para Linux y depende de unas bibliotecas gráficas llamadas Qt, que pertenecen a Troll Tech. Sin embargo, la licencia de Qt no cumplía la OSD, dado que se requería una licencia especial para incorporar dichas bibliotecas en aplicaciones que no fueran X Windows System. (Qt obtenía ingresos por la cesión de licencias a Microsoft y Macintosh). Por lo tanto, la aplicación libre KDE tenía incorporados elementos que no se consideraban libres. Bajo la presión de la comunidad libre y de la OSI en particular, Troll Tech acordó, en un primer paso, crear una licencia especial para liberar el código Qt en el caso de la fusión o la quiebra de la empresa. Luego, con el inicio del desarrollo de GNOME, un producto abierto directamente competitivo con KDE, y con la creación de bibliotecas libres similares a Qt (como Harmony), Troll Tech modificó su licencia para que cumpliera la OSD.

## 2. Estudio particular de las licencias de software libre

Como hemos indicado, el abanico de licencias libres se extiende desde las licencias permisivas, que no imponen obligaciones mayores que la de adjuntar las condiciones y el *disclaimer*, hasta las licencias con *copyleft* robusto, que obligan a mantener la misma licencia para redistribuciones del software y de cualquier obra derivada.

Así pues, a los fines de nuestro estudio, hemos clasificado las licencias libres en tres categorías (más una), que examinaremos en este apartado. Estas tres categorías son las siguientes:

- 1) Las **licencias permisivas** de tipo BSD, que incluyen las licencias MIT y X (compatibles con la GPLv2), y la AFL o la ZPL (incompatibles con la GPLv2).
- 2) Las **licencias con *copyleft* robusto**: la GPLv2 y la GPLv3 en particular, y la CPL (de IBM) de manera accesoria.
- 3) Las **licencias híbridas o con *copyleft* suave**: la LGPLv1 y la LGPLv2, la MPL y la OSL.

La categoría adicional incorpora otras "licencias" que no son libres, aunque intenten aprovechar el modelo de desarrollo libre: las licencias Shared Source (por ejemplo, la Microsoft Shared Source Initiative), que estudiaremos en el apartado 3.1. Finalmente, como cualquier programa es casi inútil sin su documentación técnica correspondiente, consideraremos unas licencias de documentación libre, en el apartado 3.2.

Para cada licencia, organizaremos los comentarios de la manera siguiente:

- 1) Una introducción.
- 2) Los derechos otorgados y las obligaciones y restricciones impuestas.
- 3) Otros elementos importantes.
- 4) Algunos aspectos prácticos útiles para su comprensión y uso.

Para los elementos más importantes, indicamos las cláusulas relevantes, a fin de que podáis encontrar la disposición en la licencia. Recomendamos que descarguéis de Internet las licencias en cuestión y las tengáis a mano durante el estudio.



## 2.1. Las licencias permisivas: sin *copyleft*

En este apartado, presentamos algunas de las licencias libres más utilizadas en la comunidad de desarrollo libre, especialmente la licencia BSD, que ha sido un modelo para muchas otras licencias. La mayoría nacen del mundo académico (tal como indica su nombre: Berkeley Software Distribution, MIT, Education Community License...), y por ello se han llamado también "licencias académicas". Estas licencias se encuentran "en un extremo" del abanico de licencias libres, porque no contienen obligaciones de *copyleft* y permiten la privatización de obras derivadas, compuestas o colectivas, que incluyan software bajo las mismas.

La primera generación de estas licencias (BSD, MIT/X, Apache 1.0 y Apache 1.1) se caracteriza porque son muy cortas y no incluyen más obligaciones que las de mantener, a la hora de redistribuir el software, los avisos de autor en los ficheros fuente y la lista de condiciones (sobre todo el *disclaimer*). El objetivo principal de estas licencias es otorgar a los destinatarios todos los derechos de explotación sobre el software (derechos de reproducción, modificación, distribución y comunicación pública) para que los licenciarios puedan hacer "lo que quieran" con el código. No contienen las obligaciones de *copyleft* y permiten incorporar y combinar el software con cualquier tipo de obra, libre o propietaria. Por ejemplo, se dice que hay componentes de software BSD en los sistemas operativos Windows NT y OS-X de Macintosh.

Las generaciones siguientes (Apache 2.0 y AFL) incluyen una serie de nuevas condiciones relativas a las patentes, el derecho aplicable, etc., muy en la línea de la Mozilla Public License (que comentaremos más adelante), para modernizar y clarificar sus términos.

### 2.1.1. La licencia Berkeley Software Distribution (BSD) y similares

La licencia Berkeley Software Distribution (BSD) es quizás el modelo más simple de todas las licencias libres. Surge de las distribuciones de versiones de Unix de la Universidad de California, Berkeley, en las décadas de 1970 y 1980, en las raíces del movimiento de software libre. El principio subyacente en la licencia es que el software es fruto de las investigaciones y los trabajos universitarios financiados por el Gobierno de los Estados Unidos (y los impuestos del pueblo americano) y que, por lo tanto, debe ser de acceso libre, de manera que sólo proteja lo que aquí llamaríamos los "derechos morales" de los autores por la simple obligación de mantener los avisos de autoría (*copyright notice*).

La ficha siguiente resume las principales características de la licencia:

Aspecto	Contenido	Comentarios
Modelo	Original.	
Objeto de la licencia	No se indica.	Se presume el software que acompaña la licencia, y en la práctica, en el mismo software se indica el uso de la licencia BSD.
Derechos otorgados	Redistribución y uso, con o sin modificación. En forma de código fuente o código binario.	Implícitamente, se entiende que incluyen la copia, la modificación y la comunicación pública del software.
Atribución	Incluir en código fuente o documentación.	
Acceso al código fuente	No es obligatorio.	
Copyleft	No hay.	
Otras obligaciones	Mantener el aviso de <i>copyright</i> , el <i>disclaimer</i> y las condiciones. No usar el nombre del autor para promocionar el software.	Si se distribuye en forma de código objeto, el <i>disclaimer</i> debe estar en la documentación.
Garantías / responsabilidades	Excluidas.	
Versiones	No indica.	
Patentes/marcas	No indica.	
Jurisdicción / derecho aplicable	No indica.	
Otros	No indica.	

### Elementos esenciales

- **Derechos otorgados.** Permite el uso, la modificación, la copia y la redistribución sin restricción del software bajo BSD, en formato de código objeto (binario) o código fuente.
- **Obligaciones impuestas.** La distribución en forma de código fuente ha ir acompañada del aviso de *copyright*, la lista de condiciones y la negación de cualquier garantía y responsabilidad. Las redistribuciones en código binario deben reproducir lo mismo en la documentación. No se puede usar el nombre del autor o de los contribuyentes para fines de promoción de obras derivadas sin su permiso.
- **Otros términos.** No se otorga ninguna garantía sobre el correcto funcionamiento del programa y se niega cualquier responsabilidad.

Por lo tanto, se puede realizar casi cualquier acto con código bajo BSD, siempre que se respete la mención de autoría del programa inicial y se incluya la lista de condiciones en el código o la documentación. Además, no hace falta proveer al usuario final del código fuente.

## Comentarios

La primera versiones de la licencia imponía la obligación de atribuir cada componente a sus autores originales en cualquier publicación o material de promoción del programa o la obra derivada. Esta obligación causaba ciertas molestias, porque había que incluir atribuciones de autoría extensivas en toda la documentación y en el código fuente, relativas a cada autor que agregaba su nombre en una licencia. En un programa con cientos de contribuyentes, era muy difícil cumplir esta obligación. Además, ello hacía que código bajo la BSD fuera incompatible con código bajo la GPL. En julio de 1999, se eliminó esta obligación de la licencia BSD. De todos modos, actualmente, hay que fijarse bien en la versión de la licencia aplicada a código con BSD, a no ser que haya una versión anterior, y entonces asegurarse del correcto cumplimiento de los términos.

Por la libertad que se otorga a los desarrolladores para mezclar código bajo la BSD con código propietario, **la licencia BSD es más favorable para el mundo de los negocios** y los desarrollos comerciales y propietarios. Asimismo, permite una gran difusión del software y su uso como referencia o estándar (para protocolos, servicios, bibliotecas e incluso sistemas operativos completos, como el Unix BSD). Sin embargo, permite también lo que se llama la *bifurcación de código* (*forking*), porque cualquiera puede adaptar, modificar y extender el núcleo del programa y crear una versión "similar pero suficientemente diferente". Esto se nota, por ejemplo, en la multiplicación de sistemas operativos con licencia de tipo BSD, como el OpenBSD, el FreeBSD y el NetBSD.

Cualquier software con licencia BSD es compatible con software con GPL (y casi cualquier otra licencia de software libre), pero no al revés. Es decir, se puede incorporar código BSD en un programa bajo la GPL (con el resultado de una obra combinada bajo GPL), pero no se puede incorporar código GPL en un software BSD.

## Otras licencias similares a la BSD

La BSD ha sido modelo de muchas licencias parecidas, entre las cuales citaremos las licencias MIT y las de la familia X (X, XFree86, XOpen, X11), la licencia Apache 1.1 (que comentaremos a continuación), las licencias Cryptix, Python, W3C Software Notice, Zope Public License (ZPL), LDAP Public License, Phorum, etc., y las licencias OpenSSL y Sleepycat, que siguen el modelo simplificado de la licencia BSD, pero que incluyen cláusulas de *copyleft* (tal como comentaremos en el apartado sobre licencias con *copyleft*).

Las licencias X y MIT son similares a la BSD pero, por un lado, especifican más los usos permitidos: "el uso, la copia, la modificación, la fusión, la publicación, la distribución y/o la venta del software"; y por el otro, no distinguen entre distribuciones de código fuente y objeto.

### Lectura recomendada

Para un comentario interesante, podéis consultar E. Leibovitch, *License to FUD (comparing GPL and BSD)* (en la bibliografía).

Algunas de las licencias mencionadas llevan cláusulas adicionales que imponen mayores restricciones, haciéndolas incompatibles con las licencias con *copyleft* (la GPL en particular). Las comentaremos en la tabla al final de este apartado.

### 2.1.2. Las licencias Apache (ASL)

El servidor web Apache proviene de los laboratorios del National Center for Supercomputing Applications de la Universidad de Illinois, Estados Unidos, y ahora es "gestionado" por la Fundación Apache, tanto la técnica y organizativa como desde la perspectiva legal. La Fundación, ha redactado las licencias Apache Software License (ASL), cuyas versiones han sido la ASL 1.0, la ASL 1.1, y ahora, la ASL 2.0, puesto que desde enero de 2004 todo el software de la Fundación Apache se publica bajo la ASL 2.0. Dada la importancia que tienen la Fundación y el software Apache en general en la comunidad de software libre, en términos de calidad del software y su modelo de gestión, la ASL es una licencia que ha sido usada por muchos otros proyectos.

#### La Apache Software License 1.1

La ASL 1.1 es una variante de la BSD (por lo tanto, no ofreceremos de ella una ficha resumen), y agrega algunas obligaciones extras:

- Hay una obligación de mantener la publicidad acerca de los autores originales en la documentación o en el software de redistribuciones: "This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>)".
- Las obras derivadas no deben usar el nombre Apache sin la autorización de la Fundación Apache (para mantener la reputación de los autores originales).

Por estas obligaciones adicionales, la ASL 1.1 no es compatible con la GPLv2. Observamos que la primera versión de la licencia (ASL 1.0) tenía la misma obligación de publicidad que la BSD respecto de los materiales publicitarios que mencionan el producto.

#### Apache Software License 2.0

La licencia Apache 2.0 fue publicada en enero de 2004 y pertenece a la nueva generación de licencias libres. Es una licencia muy completa desde la perspectiva legal, que incorpora muchas de las modernizaciones que había aportado la Mozilla Public License en 1998 (que comentaremos a continuación): definiciones completas, una licencia de patente y un pacto de *patent peace*, una obligación de indicar las modificaciones, un fichero `notice.txt`, etc. Mantiene su grado de permisividad: no es una licencia *copyleft*.

Aspecto	Contenido	Comentarios
Modelo	BSD + elementos de MPL.	
Objeto de la licencia	Cualquier obra respecto de la que se indica que la licencia es la Apache 2.0.	La indicación puede estar en el código fuente o en la documentación adjunta.
Derechos otorgados	Reproducción, modificación, distribución y comunicación pública ( <i>display/perform</i> ), y sublicencia. En forma de código fuente o código binario.	Es una licencia completa desde la perspectiva de los derechos de explotación bajo la propiedad intelectual.
Atribución	Incluir en código fuente o documentación.	
Acceso al código fuente	No es obligatorio.	
Copyleft	No hay.	
Otras obligaciones	Incluir una copia de la licencia con la obra, indicar cualquier modificación y mantener los avisos de <i>copyright</i> , marcas o patentes sobre ella. Mantener cualquier fichero notice.txt con el código o la documentación. No usar el nombre del autor para publicar el software.	El fichero notice.txt se incluye para incluir menciones de autoría, modificaciones y cualquier otra mención legal.
Garantías / responsabilidades	Excluidas.	
Versiones nuevas	No indica.	
Patentes / marcas	Licencia de patente, con revocación en el caso del inicio de una demanda basada en patentes. Obligación de mantener cualquier aviso referente a las marcas, pero prohibición de uso de las marcas del licenciente.	
Jurisdicción / derecho aplicable	No indica.	
Otros		

## Elementos esenciales

- **Derechos otorgados.** Permite la reproducción, la modificación, la distribución y la comunicación pública (*performance* y *display*, en derecho americano), con derecho a sublicencia, del software bajo ASL en formato de código objeto (binario) o código fuente. Incluye el derecho explícito de usar otra licencia sobre las modificaciones o cualquier obra derivada "como un todo", siempre que ésta cumpla las condiciones de la licencia ASL 2.0.
- **Obligaciones impuestas.** La redistribución del software se ha de acompañar de la licencia, un aviso en caso de haber cambiado cualquier fichero, cualquier aviso original de *copyright*, patente o marca, cualquier fichero notice.txt (con menciones de autoría, modificaciones y cualquier otra mención legal). No se puede usar el nombre o las marcas del licenciente o de los contribuyentes.
- **Otros términos.** No se otorga ninguna garantía sobre el correcto funcionamiento del programa y se niega cualquier responsabilidad. Asimismo,

se incluye una licencia de patente (revocable en el caso de que se inicien acciones legales basadas en patentes contra **cualquier** persona respecto al software).

Comentaremos más adelante, en el apartado dedicado a la Mozilla Public License, los términos y los objetivos de la licencia de patente y el fichero notice.txt, ya que estos conceptos se crearon con esta licencia.

## Comentarios

De la misma manera que la Fundación Apache es un modelo para la gestión de comunidades y proyectos libres, su nueva licencia es un instrumento usado por muchos proyectos, sobre todo los que trabajan con tecnologías Java o las de la Fundación Apache (Tomcat, ANT, bibliotecas como Commons log4jar, etc.). Es incompatible con la GPLv2, según la FSF (por la licencia explícita de patente), y se considera que la ASL 2.0 es ahora compatible con la GPLv3. La importancia de esta licencia radica en el objetivo expreso de la FSF de crear una licencia GPLv3 compatible con ella.

### 2.1.3. Otras licencias permisivas

Basta con consultar la página de licencias de la OSI o de la FSF para ver que hay multitud de licencias permisivas. A continuación, presentamos una tabla que recoge las licencias más comunes de software libre de tipo "permisivo", junto con un breve comentario de cada una de ellas.

La tabla se divide en dos partes. En primer lugar, se mencionan las licencias que son compatibles con la GPLv2, en el sentido de que se puede integrar código de estos programas en un programa o su obra derivada bajo la GPLv2 (todavía queda pendiente de estudiar la compatibilidad con la GPLv3). En segundo lugar, se mencionan las licencias que son incompatibles con la GPLv2, licencias que incluyen obligaciones que son más restrictivas que la GPLv2. En muchos casos, la incompatibilidad deriva de una obligación de publicidad que viene de la primera versión de la BSD, pero también puede surgir de obligaciones sobre patentes, indemnizaciones u otros temas comentados en la tabla.

Licencias compatibles con la GPL	Comentarios
<b>Zope Public License 2.0</b>	Sigue el modelo BSD e incluye una cláusula que prohíbe expresamente el uso de las marcas, registradas o no ( <i>servicemarks</i> ), de Zope Corporation, excepto bajo un acuerdo paralelo específico. También se debe avisar de los cambios de los ficheros, con la fecha de modificación.
<b>Open LDAP License 2.7</b>	Sigue el modelo BSD e incluye una cláusula que permite al autor original revisar la licencia, similar a la disposición de versiones de la GPL.
<b>Artistic License 2.0</b>	Es una licencia modelada sobre la GPL pero sin <i>copyleft</i> . Es un poco larga y complicada de entender, a pesar de que divide los usos permitidos en párrafos separados (uso con y sin modificación, distribución con y sin modificación, lo mismo con y sin código fuente, etc.).

Licencias compatibles con la GPL	Comentarios
Perl	Es una licencia muy particular, mezcla de la GPL y la antigua licencia Artistic. Se puede elegir una u otra. Si se elige la GPL, su código es compatible con la GPL, por supuesto. La FSF recomienda las versiones 4 ó 5.
Licencias incompatibles con la GPL	Comentarios
Academic Free License 3.0	Licencia permisiva, "completa" (según el modelo de la MPL), redactada por Lawrence Rosen, ex asesor legal de la OSI. Es fruto de un esfuerzo por crear una licencia moderna que aporte certidumbre legal frente a varios temas que daban lugar a dudas (derecho aplicable, definición de <i>licenciante</i> y <i>licenciario</i> , etc.). Es incompatible con la GPL por los pactos adicionales respecto a las patentes, entre otros. Se adecua más al marco legal europeo (da garantías de título, por ejemplo).
Python 2.0.1, 2.1.1 y versiones siguientes	Una licencia similar a la BSD, pero que obliga a aplicar el derecho del estado de Virginia al programa (y potencialmente a sus obras derivadas). Como la GPL no tiene una cláusula de derecho aplicable, esto constituye una restricción adicional incompatible con ella.
PHP 3.0	Una licencia de la familia BSD, pero incluye la obligación de incorporar una cláusula de publicidad de PHP.
Q Public License (QPL) 1.0	Una licencia que obliga a distribuir cualquier modificación como un parche sobre el programa inicial. Además, hay que remitir al proveedor inicial (Trolltech) cualquier modificación que no esté a disposición del público.

## 2.2. Las licencias con *copyleft* robusto

En el primer apartado de este módulo hemos explicado el concepto de *copyleft* desde la perspectiva legal: la obligación de usar la misma licencia para las redistribuciones del software, con o sin modificaciones, o de un programa que contenga el software original. En este apartado explicaremos con detalle dos licencias con *copyleft* robusto: la GPL, que es una de las bases fundamentales del movimiento de software libre; y la CPL o Eclipse PL, que es una licencia moderna presentada por IBM y usada en su plataforma Eclipse. Con respecto a la GPL, comentaremos con bastante detalle tanto la versión 2 (GPLv2) como la reciente versión 3 (GPLv3). Luego, al final del apartado, nos referiremos a algunas otras licencias que tienen cláusulas de *copyleft*.

Vemos que casi todas las licencias con *copyleft* son incompatibles entre sí, puesto que todas obligan a usar la misma licencia para la redistribución, con lo que surge un conflicto respecto a qué licencia hay que aplicar para un programa que mezcle dos componentes bajo licencias *copyleft* diferentes.

### 2.2.1. La Licencia Pública General GNU, versión 2.0 (GPLv2)

Creada en 1989, la GPLv2 ha sido descrita como "una parte manifiesto político y otra parte licencia": en su preámbulo, contiene una enunciación de la filosofía del software libre y un resumen sencillo de la licencia; la parte principal especifica los derechos otorgados a los usuarios y las condiciones y las limitaciones impuestas a la explotación del software. Es importante resaltar que pese a su tono familiar y sencillo, la GPLv2 fue diseñada por Richard Stall-

man con sus asesores legales americanos, por lo que no contiene disposiciones cualesquiera, sino un mecanismo de transmisión y resolución de derechos deliberado y muy sutil.

Aspecto	Contenido	Comentarios
Modelo	Original, 1989.	El autor es la Free Software Foundation.
Objeto de la licencia	El "programa" y las obras derivadas y basadas en el programa.	Presenta una interpretación amplia por la comunidad de software libre.
Derechos otorgados	Reproducción, modificación y distribución.	La distribución ha de entenderse en el sentido americano, que incluiría la comunicación pública.
Atribución	Incluir en código fuente e indicar modificaciones.	
Acceso al código fuente	La forma preferida de la obra cuando se le hacen modificaciones. La definición incluye <i>scripts</i> para la compilación y la ejecución y las API. En caso de distribución de binario, el código fuente: debe distribuirse con el binario o estar a disposición por tres años para todos, gratis.	
<i>Copyleft</i>	Robusto: cubre el programa y cualquier obra derivada o que integre el programa (todo o parte de él), que deben redistribuirse bajo la GPL. Incluye obras colectivas.	La FSF argumenta que no permite vínculos dinámicos sin aplicar la GPL al "todo".
Otras obligaciones	No se pueden imponer mayores restricciones que las incluidas en la licencia.	Esto hace que muchas otras licencias sean incompatibles con la GPL.
Garantías/responsabilidades	Excluidas en la medida permitida por ley.	
Patentes/marcas	No indica.	Es una licencia implícita respecto del "uso" del programa.
Jurisdicción / derecho aplicable	No indica.	
Versiones	Permite su actualización por la Free Software Foundation.	De hecho, acaba de actualizarse a la GPLv3.
Otros		

A continuación presentamos con detalle, debido a su importancia, la GPLv2.

### Definiciones útiles

Aunque no sean explícitas, como en la MPL o la CPL, y ahora en la GPLv3, la licencia GPLv2 contiene varias definiciones que son de gran utilidad:

- **Programa:** cualquier programa u obra al cual se ha adjuntado la licencia. Técnicamente, podría aplicarse a un fichero de texto, de imagen u otro (cláusula 0).
- **Obra basada en el programa:** el programa original o cualquier obra derivada de él según la definición del derecho de *copyright*. Esto significaría



una obra que contuviera el programa o una parte del mismo, ya fuera una copia fiel o literal, o con modificaciones (cláusulas 0 y 2).

- **Código fuente:** la forma preferida de la obra para hacerle modificaciones. Respecto a un ejecutable, la obligación de proporcionar el código fuente incluye todos los módulos que aquél contenga, más los archivos con la configuración de la interfaz y los guiones (*scripts*) utilizados para controlar la compilación y la instalación. No incluye el código fuente de los módulos equivalentes del sistema operativo en el que el programa se ejecuta, salvo que dichos módulos acompañen el ejecutable (cláusula 3).

### Los elementos esenciales de la licencia

El primer elemento importante son los principales **derechos otorgados por la licencia**. Éstos garantizan las cuatro libertades fundamentales:

- el derecho de reproducción y de distribución del código fuente original (cláusula 1);
- el derecho de modificación del programa o de parte de él (cláusula 2);
- el derecho de distribución del código fuente de las eventuales modificaciones, siempre que se distribuyan con la misma licencia GPL y sin cobrar por ella (cláusula 2b –la cláusula *copyleft*–);
- el derecho de reproducción y de redistribución en forma de código objeto o ejecutable del programa (y de sus modificaciones), con la misma condición *copyleft* y siempre que se acompañe del código fuente o que éste se ponga a disposición de cualquier tercero, sin cobrar más que el coste de la entrega de dicho código fuente (cláusula 3).

El **acceso al código fuente** es el segundo aspecto fundamental de la licencia. Un programa puede distribuirse con la GPL en formato binario (código objeto), pero se debe acompañar con el código fuente o el ofrecimiento de proporcionarlo a cualquier tercero durante un plazo de tres años (cláusula 3).

El tercer elemento esencial se refiere a **las restricciones, las condiciones y los actos prohibidos**. La GPL mantiene cierto control sobre el programa, no con respecto a su uso, sino a su transmisión a terceros:

- cualquier distribución del programa o de una obra derivada de éste se debe acompañar de los avisos de autoría, una indicación de cualquier modificación que se hubiera hecho (y la fecha de la misma), el aviso de que no hay ninguna garantía y una copia de la licencia (cláusulas 1 y 2);

- no se permite copiar, modificar o distribuir el programa de una manera distinta de la expresamente permitida por la licencia, con menos libertades o mayores restricciones (cláusula 2b y cláusula 6);
- si se intenta realizar cualquier acto que viole la licencia, el licenciatarario perderá sus derechos originales (cláusula 4).

### Otros aspectos importantes de la licencia

a) **Ámbito de aplicación.** La licencia GPL se aplica a cualquier **programa u otra obra** que contenga un aviso del titular del derecho de autor en el que se indique que puede distribuirse bajo los términos de la misma. Respecto a los **actos cubiertos** por la licencia, son solamente los de reproducción, modificación y distribución del programa. Es decir, no se regula ningún otro acto, sobre todo el uso o la ejecución del programa. La licencia considera que éste no es un acto regulado por el derecho de autor y que los usuarios legítimos son, por lo tanto, libres de ejecutar el programa como quieran.

b) **Autoría.** Se respeta el reconocimiento del autor del software. Por ello:

- La licencia obliga a mantener de forma adecuada y bien visible el aviso de autoría sobre cualquier copia del programa (cláusula 1). En particular, en el apéndice de la GPLv2, se recomienda que el autor añada al principio de cada fichero fuente una línea de *copyright* en la forma clásica: "© año, autor".
- Cualquier modificación debe llevar avisos visibles, que indiquen la autoría (cláusula 2), de que se ha cambiado el programa original y la fecha del cambio (cláusula 2a). Si el programa modificado es interactivo, debe procurar mostrar el anuncio de *copyright* al iniciarse la ejecución en uso interactivo (cláusula 2c). Así quedará protegido el autor original en caso de degradación de calidad o de no funcionamiento del programa modificado.

c) **Aceptación.** La licencia considera que no es necesario "aceptar" la GPLv2 para usar el programa. Sin embargo, para modificarlo y distribuirlo, sí que se necesita esta aceptación. Ya hemos visto que en ausencia de una licencia otorgada por el autor –y, por lo tanto, en ausencia de esta aceptación de la licencia por parte del usuario–, las acciones de modificación y distribución están prohibidas por las leyes de derechos de autor. La GPLv2 entiende que, por el hecho de modificar o distribuir el software, el usuario está indicando que acepta la GPLv2 y todos sus términos y condiciones.

d) **Versiones.** La licencia permite a los autores-licenciantes a referirse a nuevas versiones de la misma (comentamos la versión 3.0 más adelante) agregando que la obra se publica "bajo la versión 2 y cualquier versión posterior" (cláusula 9). Linus Torvalds, por ejemplo, ha excluido las versiones posteriores para el núcleo (*kernel*) de Linux: se queda con la versión 2.0. Esta flexibilidad permite

que sus programas puedan mantener la compatibilidad con futuros programas bajo una versión posterior de la GPL –como la recientemente publicada GPLv3. En este caso, los licenciarios pueden elegir la versión aplicable.

**e) Garantías.** Las cláusulas 11 y 12 aclaran que no se ofrece ninguna garantía sobre el funcionamiento correcto del software cubierto por la licencia y niegan cualquier responsabilidad por daños y perjuicios. Sin embargo, hemos visto en el módulo 5 que la validez de estas cláusulas es dudosa (en jurisdicciones distintas a las de los Estados Unidos e incluso en los Estados Unidos en algunas circunstancias) por el derecho de protección del consumidor y la prohibición de las cláusulas abusivas en contratos de adhesión.

**f) Derecho aplicable.** La licencia GPLv2 no incluye ninguna cláusula que indique el derecho aplicable o los tribunales competentes para atender un conflicto que se refiera a ella. Por lo tanto, se aplicará el derecho relevante en los tribunales correspondientes bajo los principios del derecho internacional privado. En la mayoría de los casos, será el derecho del domicilio legal del licenciante, pero un consumidor, por ejemplo, puede elegir el derecho y los tribunales de su domicilio.

**g) Patentes.** El último párrafo del preámbulo resalta los peligros que presentan las patentes para el software libre. Sin embargo, la GPLv2 no incluye ninguna cláusula que restrinja las patentes eventuales sobre software bajo GPLv2 o que obligue a licenciarlas a favor de los demás usuarios (la GPLv3 sí que lo hace). Como consecuencia lógica de la obligación a distribuir el programa y cualquier obra derivada de él en términos iguales a los de la GPLv2 (cláusula 2b), cualquier licenciario que obtenga una patente sobre una obra de software bajo GPL deberá permitir el uso libre conforme a la GPLv2 por parte de todos los destinatarios posteriores –lo que podría considerarse como una licencia implícita de patente. Veremos que en la GPLv3 se especifican los términos de esta licencia de patente.

## Comentarios

Queremos comentar algunos problemas potenciales que pueden surgir en la aplicación de la licencia GPLv2 en particular, examinando el alcance del *copyleft*, las traducciones, su validez legal y la "compatibilidad" con ella.

**Obras derivadas y el efecto *copyleft*.** Un tema importante que hay que aclarar es la cuestión de las obras derivadas y la aplicación de la licencia a éstas por el *copyleft*. Es un concepto clave para entender la licencia GPLv2, porque define el alcance de la cláusula de *copyleft*, que es lo que más distingue a esta licencia de otras licencias libres. Este tema ha suscitado mucha polémica en el mundo del software libre y el software en general. Ya hemos dicho varias veces que no se puede "privatizar" un software bajo la GPLv2 ni las obras derivadas de ella. Por esto, algunos desarrolladores dudan de incorporar o vincular demasiado

íntimamente sus trabajos a un programa *copyleft*, pues temen verlos caer bajo la licencia GPLv2 en circunstancias en las que no puedan o no quieran permitirlo (como un desarrollo propietario o una licencia libre diferente).

¿En qué consiste una **obra derivada** o una **obra basada en el programa**, según los autores de la licencia? La definición citada anteriormente se refiere a la definición bajo el derecho de *copyright*: es una obra que contenga el programa o una porción del mismo.

Pero la palabra *contener*, en el ámbito de la programación, deja lugar a dudas: ¿se trata solamente de obras derivadas bajo la estricta interpretación legal del *copyright* o los derechos de autor?, ¿o también se aplica a "obras compuestas" o *collective works*, que incorporan el programa original? En el segundo caso, el alcance de la licencia puede ir más allá de lo que establece el marco legal de los derechos de autor, y por lo tanto, el licenciante deberá basar sus derechos en el derecho contractual.

### Interpretación de los GPLv2

La licencia y las preguntas más frecuentes (PMF o FAQ) sobre ésta nos aportan cierta ayuda.

**a) Lo que dice la licencia.** La cláusula 2b indica que el *copyleft* se aplica a cualquier obra que **contenga** o **sea derivada** del programa original, que se debe licenciar **como "un todo"** (y cada uno de sus componentes) bajo la GPLv2.

Los componentes de software pueden interrelacionarse de distintas maneras, por llamadas, vínculos o enlaces de distintos tipos. La compilación de un programa (para crear el ejecutable) puede incorporar diferentes componentes en un solo programa, o los diferentes componentes pueden interrelacionarse cuando se interpreta el programa en el momento de la ejecución. Cada una de estas interacciones puede tener efectos legales diferentes.

La cuestión debatida es si estas arquitecturas suponen o no que la obra resultante caiga total o parcialmente bajo la GPL. Este tema se ha complicado por la evolución de los métodos de programación (estructurados o por objetos) y los lenguajes informáticos (C, C++, Visual Basic, Java, PHP, etc.), muchos de los cuales no existían en el momento de la redacción de la licencia.

Observemos primero que la mera reunión o agregación de una obra (separable y no basada en un programa bajo GPL) en un mismo soporte o medio de almacenamiento con software con GPLv2, por ejemplo para su distribución, no implica que esta otra obra deba ser distribuida bajo la GPLv2. Asimismo, la licencia aclara que si partes identificables de una obra pueden ser razonablemente consideradas obras independientes y separadas por sí mismas, la licencia no se aplica a estas partes.

Frente a otros casos, la prudencia dicta que se deben evaluar los riesgos legales respecto de un desarrollo o arquitectura en particular, considerando el diseño y las consecuencias potenciales de caer bajo la GPL. Podemos decir con cierta seguridad lo siguiente:

- Si cuando se compila un desarrollo nuevo basado en un software con GPLv2, el ejecutable final incluye elementos del programa original (será el caso de componentes con vínculos estáticos entre ellos), entonces las modificaciones no se pueden considerar separables y, en consecuencia, la obra completa y cada una de sus partes tendrán que distribuirse bajo la GPL.
- Si el programa original GPLv2 y el desarrollo nuevo pueden coexistir de manera separada (incluso cuando se encuentren sobre un mismo soporte) y el desarrollo particular llama al programa GPLv2 en tiempo de ejecución (éste es el caso de un vínculo dinámico), desafortunadamente, la situación no es tan clara. Hay dos opiniones:
  - 1) R. Stallman, autor de la licencia, ha insistido muchas veces en que considera que los programas con vínculos dinámicos a código GPLv2 sí que quedan afectados por la GPLv2. Precisamente, sostiene que por ello se inventó la licencia LGPL,

### Lecturas recomendadas

**D. Ravicher.** *On open source legal issues.*

**M. Assay.** *A funny thing happened...*

**L. Rosen.** *The unreasonable fear of infection.*

### Web recomendada

Las preguntas más frecuentes (PMF o FAQ) están en [www.gnu.org/licenses/gpl-faq.html](http://www.gnu.org/licenses/gpl-faq.html).

para permitir este tipo de vínculos sin aplicación del *copyleft* de la GPLv2. Ésta es la interpretación y la definición que se incluye explícitamente en la GPLv3.

- 2) En la práctica y a lo largo del tiempo, muchas personas han tomado la posición contraria, argumentando que una obra que tenga vínculos dinámicos con código bajo GPLv2 no se verá afectada por la GPLv2.

**b) Las FAQs de la GPL.** Entre las "preguntas más frecuentes" sobre la licencia GPLv2, hay unas cuantas que exponen casos de modificaciones, vínculos y llamadas a código bajo GPL que la FSF "resuelve" ofreciendo su interpretación de la licencia y del derecho. Por ejemplo, aclara que un programa nuevo compilado por un compilador (*compiler*) bajo GPL no tendría que distribuirse bajo esta licencia, excepto si el ejecutable resultante después de la compilación incorporara elementos del compilador libre u otro programa bajo GPL.

Pero el tema no está resuelto del todo para la GPLv2 y, al final, queda a juicio de los creadores de modificaciones y obras derivadas considerar si éstas caen bajo la GPL (y cuándo consultar a sus asesores legales).

### **Linus Torvald y el *copyleft***

Linus Torvalds ha incluido expresamente, en la licencia GPL que cubre el núcleo Linux del SO GNU/Linux, una adenda para manifestar que él, como autor licenciante, no considera que los programas que tienen vínculos dinámicos al núcleo estén afectados por *copyleft*. Las aplicaciones de usuarios y otros elementos no centrales de un sistema operativo, como los controladores de dispositivos (*drivers*), interactúan de manera dinámica con los componentes y los módulos del núcleo del sistema. Por ello las aplicaciones y los controladores son específicos para una plataforma u otra. Hay una posibilidad de que esta interacción con un sistema operativo bajo la GPL afecte a estos programas y controladores. Sin esta aclaración, casi cualquier programa que se ejecutase sobre GNU/Linux y llamase a sus bibliotecas centrales podría considerarse, si se tomara la interpretación más estricta de la licencia, afectada por la GPL. Esto reduciría el uso y la difusión de GNU/Linux como sistema operativo a un entorno de programas compatibles con la GPL. Sin embargo, con el tiempo, L. Torvalds parece haber evolucionando hacia una interpretación más cercana a R. Stallman...

**Traducciones.** La GPLv2 no tiene traducciones oficiales. Es decir, la versión original en inglés será la que determine los términos de distribución cuando se aplique la GPL original a una obra. Hay traducciones oficiosas indicadas en las páginas de la FSF, que ésta no aprueba como oficialmente (jurídicamente) válidas. Observad que si un autor aplica una licencia GPL traducida a su programa, será la traducción de la licencia la que prevalezca, no la GPL original en inglés (salvo indicación contraria). Si hubiera un error de traducción, los resultados podrían ser no sólo desagradables, sino también espantosos para la comunidad del software libre. Habría versiones y modificaciones de software "casi-GPL" (en su matiz extranjero) mezcladas con programas realmente GPL (en su versión en inglés).

**Validez de la licencia como contrato.** Según se ha comentado, en muchos países, como España, las condiciones de una licencia son vinculantes si el licenciataria ha tenido la oportunidad de conocerlas antes de aceptarlas y ha expresado su consentimiento. El mecanismo de la GPL intenta asegurarse de ello:

- **Términos.** Las cláusulas 1 y 2 obligan a acompañar cualquier distribución del programa con una copia de la licencia. Por lo tanto, cualquier usuario debería tener la ocasión de ver sus condiciones.
- **Aceptación.** Ya hemos comentado que cualquier copia, modificación o distribución del programa significaría el consentimiento implícito del usuario.

Se quiere resaltar aquí que este tema no será de gran importancia en la práctica, porque la GPLv2 no intenta imponer obligaciones relativas al uso. Se limita a otorgar derechos a los usuarios, para lo cual no se necesita su consentimiento. La licencia tampoco intenta reservar o conceder al titular del software derechos mayores que los ya otorgados por las leyes de derechos de autor –lo que sí que requeriría el consentimiento expreso e informado del usuario-licenciatario.

### Derecho Anglosajón

En este aspecto, el derecho anglosajón puede ser más permisivo, ya que establece una diferencia entre una licencia de *copyright* (una autorización unilateral) y un contrato (un acuerdo mutuo). Aun si el licenciatario de código bajo GPLv2 no acepta expresamente los términos como "contrato", seguirá vinculado por las condiciones de uso como "licencia". Normalmente, estas condiciones son válidas siempre que se queden dentro del alcance de los derechos reservados a los autores por el derecho de *copyright* y no sean abusivas. Argumentamos que los ámbitos en los que la GPL impone obligaciones, relativas a la modificación y la distribución del código, están bien dentro de los derechos reservados por el *copyright* y, en consecuencia, que serían válidas y vinculantes sin la aceptación explícita por parte del licenciatario.

"[la ejecución de un software libre] es un derecho del cual todos los usuarios deben beneficiarse. Casi todos los que usan a diario software bajo la licencia GPL no necesitan licencia alguna, ni aceptan ninguna. La GPL impone obligaciones únicamente si uno quiere distribuir software derivado de código bajo GPL y solamente requiere el consentimiento cuando ocurra esta redistribución. Y, como no se puede redistribuir sin licencia, podemos suponer con seguridad que cualquier redistribuidor tiene la intención de aceptar la licencia. Después de todo, la GPL requiere que cada copia de software bajo GPL incluya la licencia; por lo tanto todos están informados".

E. Moglen, Free Software Matters Enforcing the GPL, 12/08/2001

**Compatibilidad de otras licencias con la GPLv2.** Un programa es compatible desde la perspectiva legal con software bajo GPLv2 cuando se distribuye en términos que son compatibles con los términos de esta licencia. No pueden ser más restrictivos (como los de cualquier licencia propietaria), pero sí que pueden ser más permisivos (como los de la licencia BSD modificada o los de la LGPL, que estudiaremos a continuación). La compatibilidad con la GPL tiene la doble ventaja de facilitar la integración de componentes libres en distribuciones y plataformas más complejas e integradas y de asegurar que su código puede integrarse sin miedo con el 75% de los programas de software libre disponibles en Internet. Vemos que la GPLv3 no es compatible con la GPLv2 (pero sí con software bajo la GPLv2 "y versiones anteriores"), lo que hace que sea necesario que los titulares de software bajo "solamente" la GPLv2 lo relicencien en los términos de la GPLv3 (o una licencia más permisiva) si quieren asegurar esta compatibilidad.

### Algunos ejemplos de incompatibilidad con la GPLv2

- La cláusula de la licencia BSD original y de la licencia Apache 1.0 que obligaba a incluir una mención de los autores originales en cualquier publicidad o material promocional del programa.
- Las cláusulas de reservas de derecho de la Netscape Public License, que permiten a Netscape beneficiarse de las modificaciones de terceros al Navigator e incorporarlas en nuevos productos de Netscape.
- La licencia explícita de patentes de la ASL 2.0 (según la opinión de la FSF).
- La obligación de obtener una "licencia de desarrollador" para poder integrar elementos de Qt en aplicaciones que no son Windows X System, en la licencia Qt.

### La GPL en su versión 3

El proceso de modernización de la GPLv2 empezó en 2005 y acabó el 29 de junio de 2007, cuando la FSF publicó la nueva GPLv3. Esta modernización responde a varias necesidades, entre las cuales las principales son las siguientes:

- La internacionalización de la licencia.
- Su flexibilización.
- La respuesta a los sistemas de gestión de derechos de autor (DRM) y su protección legal.
- La gestión de temas legales relacionados con las patentes de software.

A estos cuatro puntos, podríamos agregar uno más: la clarificación del alcance del *copyleft* frente a las nuevas tecnologías y arquitecturas, los vínculos dinámicos y el concepto de *código fuente*.

Ficha resumen (en cursiva, lo nuevo en relación con la GPLv2)

Aspecto	Contenido	Comentarios
Modelo	Original, 2007.	Autor Free Software Foundation.
Objeto de la licencia: " <i>obra cubierta</i> "	El "programa" con o sin modificaciones (obras "basadas" en el programa).	<i>Modificar = copiar o adaptar todo o parte del programa, excepto la copia integral.</i>
Derechos otorgados	<i>Propagación: cualquier acto que requiere el consentimiento del titular. Enlazar explícitamente con software bajo la licencia Affero.</i>	<i>Internacionalización de la licencia –la propagación incluirá en España la reproducción, la modificación, la distribución y la comunicación pública.</i>
Atribución	Incluir en el código fuente e indicar modificaciones. <i>Incluir: avisos de autor, licencia, ausencia de garantía.</i>	<i>Obligación adicional de asegurar que el usuario lo vea ("prominently visible feature").</i>

Aspecto	Contenido	Comentarios
<b>Código fuente "correspondiente" y acceso al mismo</b>	La forma preferida de la obra cuando se le hacen modificaciones. <i>Obligación de incluir todo el código fuente necesario para la generación, la instalación y la ejecución del software.</i> La definición incluye <i>scripts</i> para la compilación y la ejecución y las API, pero no elementos esenciales de sistemas operativos. <i>Explicita las maneras en las que se debe dar acceso al código fuente correspondiente.</i> Incluye códigos de acceso, para productos para consumidores.	En caso de distribución de binario, el código fuente: <ul style="list-style-type: none"> <li>• debe distribuirse con el binario o estar a disposición durante tres años / <i>mientras haya soporte.</i></li> <li>• estar disponible para todos los que tengan una copia del binario.</li> </ul> <b>Aclaración.</b> Los sistemas de acceso al código fuente incluyen la distribución por Internet desde un servidor o en redes de P2P. <i>DRM contra la Tivo-isación. La necesidad de tener una clave para ejecutar el código, una vez modificado un producto consumidor.</i>
<b>Copyleft</b>	Robusto: cubre el programa y cualquier obra derivada o que integre el programa (todo o parte de él), que deben redistribuirse bajo la GPL. Incluye obras colectivas.	<b>Clarificación.</b> Explícitamente cubre obras con vínculos dinámicos a software cubierto por la licencia.
<b>Otras obligaciones</b>	No se pueden imponer mayores restricciones que las incluidas en la licencia. <i>Permisos adicionales: se pueden agregar permisos adicionales (de tipo LGPL) y un licenciatario podrá eliminar estos permisos adicionales.</i> <i>Se pueden agregar materiales bajo licencias que varíen ligeramente de los términos de la licencia, respecto de:</i> <ol style="list-style-type: none"> <li>1) el alcance de las garantías,</li> <li>2) los avisos legales necesarios,</li> <li>3) las indicaciones de modificación,</li> <li>4) uso de nombres de terceros,</li> <li>5) la concesión de derechos de marca,</li> <li>6) las indemnizaciones a los autores.</li> </ol>	<i>Flexibilización de la licencia en casos de incorporación de software libre bajo otras licencias en un nuevo desarrollo. Son elementos incorporados en varias licencias permisivas, que las hacían incompatibles con la GPLv2 (como la Apache 2.0).</i>
<b>Garantías/responsabilidades</b>	Excluidas en la medida permitida por ley, o sustituidas por lo permitido por ley.	
<b>Patentes/marcas</b>	<i>Licencia de patente restringida.</i> <i>Obligación de extender una licencia de patente "recibida de" u "otorgada a" terceros, a los futuros licenciatarios del software.</i> <i>Prohibición de distribuir cualquier software bajo GPLv3 si se ha pactado con un tercero la protección de sus propios licenciatarios (y no a todos los licenciatarios) frente a demandas basadas en patentes.</i> <i>Prohibición de iniciar acciones legales basadas en patentes respecto al uso o la comercialización del software.</i>	<b>Patentes y flexibilización.</b> Compatible con la licencia Apache 2.0. Referencia al acuerdo de licencia cruzada de patentes entre Novell y Microsoft.
<b>Jurisdicción / derecho aplicable</b>	No indica.	
<b>Versiones</b>	Permite su actualización por la Free Software Foundation.	
<b>Otros</b>	<i>DRM: software GPLv3 no será considerado un sistema DRM.</i> <i>Resolución: Sesenta días de gracia para corregir errores que resolverían la licencia.</i> <i>Compatibilidad con la Affero GPL.</i>	<i>DRM. Impide que terceros demanden a los licenciatarios por elusión de sistemas de protección de derechos.</i> <i>Flexibilización de la licencia para eliminar obligaciones (eg. LGPLv3 = GPLv3 menos el copyleft para obras que usan la biblioteca).</i> <b>Flexibilización de la resolución de la licencia para errores involuntarios.</b>



A continuación, comentamos sobre todo las diferencias con respecto a la GPLv2.

**a) Definiciones.** En primer lugar, aparte de nuevas definiciones de *programa*, *Tú* (usuario) y *Modificar*, hay una nueva definición: *código fuente completo correspondiente* (cláusula 1) y dos términos nuevos: *propagación* y *transferir* (*convey*, en inglés) (cláusula 0).

- El alcance de la definición de **código fuente** es importante, dada la obligación de distribuir u ofrecer acceso al código fuente (bajo la GPL) de cualquier ejecutable distribuido sin él (GPLv2, cláusula 3). En la GPLv2 se define *código fuente* como "la forma preferida de la obra (programa) cuando se le hacen modificaciones" y la obligación de proveer el código fuente incluye cualquier "*script* necesario para compilar el programa". En la GPLv3, la definición de *código fuente* es la misma, pero la correspondiente obligación se refiere al "código fuente completo correspondiente", que es, *a priori*, mucho más amplio: incluye el "código necesario para generar, instalar, ejecutar y modificar (el programa)"; los *scripts* para realizar estas operaciones y definiciones de interfaz, y (explícitamente) el código fuente de bibliotecas compartidas o enlazadas dinámicamente que el programa está diseñado para usar.
- Los términos ***propagación*** y ***transferir*** se usan, conforme el objetivo de la internacionalización de la licencia, para cubrir todos los actos reservados por los derechos de autor bajo cualquier régimen legal, sin mencionar palabras como *distribuir* o *reproducir*, que podrían ser definidas legalmente de manera diferente en distintas jurisdicciones.
  - La ***propagación*** se utiliza para designar "cualquier actividad para la cual se requiera la autorización del titular del programa", excepto la ejecución del programa y las modificaciones privadas (es decir, las no destinadas a terceros). En derecho español, se hablaría de "acto de explotación" de la obra –copia, modificación, distribución y comunicación pública.
  - ***Transferir*** (*convey*) es un subgrupo de *propagar* a los efectos de las obligaciones de *copyleft* (que se activarán con la "transferencia"): significa realizar una operación de propagación que resulta en la creación u obtención de copias por terceros; por ejemplo, entregar una copia a un tercero, comunicar públicamente el software en Internet, compartirlo en redes P2P, etc.

**b) Derechos otorgados.** Mientras que la GPLv2 indica que no se necesita la autorización del titular para la ejecución del programa (considerando que el "uso" de un programa no está regulado por el *copyright*), la GPL3 concede explícitamente:

- El derecho de ejecutar y modificar para fines privados el programa sin restricciones.
- El derecho de *propagar* el programa sin restricciones siempre que no resulte en la *transferencia* del software. Así pues, esto incluye el derecho de reproducción, modificación y "distribuciones" internas. Asimismo, permite la entrega del software a terceros sin condiciones cuando se realiza bajo un contrato de consultoría según el cual el consultor hará modificaciones exclusivamente para el licenciataria (*work-for-hire*, en derecho americano).
- El derecho de *transferir* el software bajo condiciones de *copyleft*.

**c) Obligaciones.** Las obligaciones básicas respecto a la copia y la distribución del software son similares a las establecidas en la GPLv2: hay que mantener los avisos de autoría, la licencia, las notificaciones de cambios, etc. Se precisa que si el programa tiene una interfaz de usuario, debe contener un sistema para publicar los avisos de *copyright*, la ausencia de garantía y el acceso a la licencia –una obligación más fuerte que en la GPLv2.

Respecto al sistema de *copyleft*, la GPLv3 tampoco cambia mucho:

- Mantiene las obligaciones de *transferir* cualquier obra modificada "como un todo" "bajo la misma licencia" (en el pacto 2b de la GPLv2, ahora el 5c).
- Modifica ligeramente la obligación de acompañar cualquier distribución de código binario con el "código fuente completo correspondiente" u ofrecer acceso al mismo a cualquier tercero **que tenga el binario**. El plazo de este ofrecimiento es mayor de tres años o la duración de cualquier soporte u ofrecimiento de "correcciones". Además, se puede cobrar el coste de su distribución.
- Especifica cinco maneras de realizar esta distribución/ofrecimiento (como por ejemplo la distribución sobre CD, desde servidores de Internet o la compartición en redes P2P).

**d) DRM.** Ya hemos comentado en el módulo 2 el sistema legal de protección de los sistemas de gestión de derechos de autor (*Digital Rights Management* o DRM): es ilegal "eludir" (leer *crackear*) una medida tecnológica eficaz de protección de los derechos de autor. La GPLv3 tiene dos mecanismos contra estos sistemas, que considera una vulneración de la libertad de los usuarios (la FSF los llama *Digital Restrictions Management*):

- Por un lado, indica en su cláusula 3 que en ningún caso el software bajo GPLv3 se considerará parte de un "mecanismo de protección tecnológica efectiva" de derechos y que los titulares renuncian al derecho de demandar a terceros por cualquier acto de elusión consecuente al mero ejercicio de los derechos cedidos bajo la licencia. De esta manera indirecta, intenta

permitir que se modifique cualquier software bajo GPLv3 sin infringir estas nuevas reglas que prohibirían este tipo de "elusión". La consecuencia buscada es que será incompatible distribuir software GPLv3 en programas de DRM cuya licencia no permita el acceso, la modificación o la reingeniería. Queda pendiente de estudio ver si esto funciona legalmente, sobre todo dada la naturaleza imperativa del régimen legal de protección de estos sistemas DRM.

- Por otro lado, la GPLv3 incluye, en la definición de "código fuente completo correspondiente" de manera excepcional para **productos para consumidores**, las claves de acceso y descifrado y la información de instalación y ejecución de software modificado. Con la GPLv3, los fabricantes y los distribuidores de dispositivos "cerrados" para usuarios consumidores no podrían impedir el acceso al dispositivo o exigir la obtención y el correspondiente pago de una clave, por ejemplo, para "acceder a" o ejecutar el dispositivo o para modificar el código de su programa. Si lo hicieran, deberían entregar también las claves, los códigos y la información correspondientes.

**e) Patentes.** El sistema de protección contra patentes en la GPLv3 es complejo debido a las diferentes prácticas que han surgido en torno a las patentes de software. En la GPLv2, cualquier cesión de derechos de patente (sobre un proceso implementado por un software GPL) era implícita, con las consiguientes incertidumbres respecto de sus efectos legales. En la GPLv3, hay cuatro términos importantes (cláusula 11):

- La cesión de derechos de patente se hace explícita: si alguien tiene una patente sobre una contribución suya a un software distribuido bajo GPLv3, otorga una licencia de patente para usar, comercializar e importar el software contribuido a cualquiera que use esta contribución sin modificaciones;
- Cualquier licencia de patente explícita a un licenciataria se extenderá a todos los licenciarios;
- Asimismo, se intenta establecer un mecanismo de protección "en cascada": el que distribuya un software bajo GPLv3 beneficiándose de una licencia de patente de un tercero deberá extender su beneficio a todos los licenciarios, o renunciar al beneficio o asegurar que el "código fuente correspondiente" esté a disposición de todos bajo las condiciones de la GPLv3;
- Con referencia al pacto entre Microsoft y Novell de marzo de 2007 (que no será cubierto por la licencia), si una persona obtiene una protección específica respecto de software bajo la GPLv3 que, de manera discriminatoria, solamente lo protege a él y a sus licenciarios, no podrá transferir este software bajo la GPLv3.

**f) Servicios remotos o *Application Service Providers* (ASP).** Se pensaba que la nueva licencia restringiría el uso de software GPL por parte de los que ofrecen servicios comerciales a sus usuarios finales con base en software GPL sin distribuir sus programas y fuentes (Google y Yahoo! son ejemplos evidentes), o que les obligaría a entregar el código fuente de cualquier servicio ASP. Al final, se ha dejado este mecanismo para la licencia Affero GPL y se incluye un pacto de compatibilidad explícita con esta licencia.

**g) Permisos adicionales.** La GPLv3 permite agregar algunos permisos (pero no restricciones) adicionales, como excepciones a sus obligaciones. Se aplicarán a componentes identificados de software y podrán ser eliminados por los licenciarios a la hora de la redistribución. La LGPLv3 es un ejemplo de ello, puesto que consiste en la GPLv3 con el permiso adicional de enlazar con programas que "usan la biblioteca" bajo cualquier licencia (como veremos más adelante).

**h) Restricciones adicionales: compatibilidad de licencias.** La "compatibilidad legal" del software es fundamental en el desarrollo de software libre: significa poder mezclar dos programas con licencias libres diferentes, sin incumplir ninguna de las mismas en su redistribución. La GPLv2 prohíbe agregar cualquier restricción adicional que no esté en la misma licencia. Esto ha hecho que licencias con pactos sobre patentes, atribución de autores, uso de marcas, notificaciones y *disclaimers* con términos diferentes hayan sido declaradas "incompatibles" con la GPLv2 por la FSF (y por abogados que asesoran a sus clientes). La GPLv3 hace un esfuerzo para ampliar el conjunto de licencias libres que sean compatibles con ella con un nuevo mecanismo: permite agregar seis tipos de restricciones adicionales sobre programas o código agregado al código bajo GPLv3.

#### **Restricciones permitidas**

Estas restricciones son compatibles si se refieren a:

- 1) El mantenimiento de avisos de autor u otras formas de atribución (por ejemplo, avisos de *powered by* o ventanas *about*) y obligaciones sobre indicación de cualquier modificación en el código.
- 2) *Disclaimers* (exclusiones de garantías y limitaciones de responsabilidades) en términos diferentes de los de la GPLv3.
- 3) Cómo indicar las modificaciones.
- 4) Restricciones sobre el uso de nombres de autores para fines publicitarios (la antigua licencia BSD sigue siendo incompatible).
- 5) Concesiones de derechos o prohibiciones respecto a las marcas.
- 6) Indemnizaciones para los contribuidores.

Las licencias Apache 2.0, OSL y EclipsePL son ejemplos de licencias que podrían volver a ser compatibles.

### 2.2.3. La Common Public License y la Eclipse Public License

Las licencias CPL y EPL (y su predecesora, la IBM Public License) son nuevos instrumentos legales desarrollados por IBM, con un formato diferente de la GPL y la BSD, los dos modelos predominantes. La CPL, que comentaremos aquí, es más cercana a la Mozilla Public License, que analizaremos a continuación, ya que tiene una forma más "legalista" (con definiciones, derecho aplicable, etc.) y cubre temas como la indemnización entre contribuidores y las licencias de patentes.

Aspecto	Contenido	Comentario
Modelo	Original, con alguna influencia de la MPL.	Autor IBM.
Objeto de la licencia	Programa y obras derivadas bajo derecho de <i>copyright</i> .	
Derechos otorgados	Copia, modificación, distribución, ejecución en público ( <i>perform</i> ) y comunicación pública ( <i>display</i> ).	
Atribución	En el código fuente.	
<i>Copyleft</i>	Indirectamente fuerte, ya que cualquier obra derivada debe distribuirse: <ul style="list-style-type: none"> <li>• en código fuente: bajo la CPL;</li> <li>• en código objeto: bajo una licencia compatible con la CPL.</li> </ul>	Es <i>copyleft</i> robusto porque la única licencia compatible con la CPL es la CPL, y cubre tanto el programa original como cualquier obra derivada.
Acceso al código fuente	Debe acompañar la distribución del binario.	
Otras obligaciones	La licencia debe acompañar el programa en cualquier distribución. Indemnizaciones para los autores originales y otros contribuidores en caso de distribuciones comerciales.	
Garantías/responsabilidades	Excluidas y limitadas.	
Versiones	Permite su actualización por parte de IBM.	
Patentes/marcas	Licencia explícita y limitada de patente sobre el código original y las contribuciones. <i>Patent peace</i> en relación con demandas contra los contribuidores respecto a cualquier software, y contra cualquier persona relacionada con el software en cuestión.	La <i>patent peace</i> revoca las licencias de patente, no las de derecho de autor.
Jurisdicción / derecho aplicable	Tribunales de Nueva York / Derecho de los Estados Unidos.	
Otros		

### Definiciones

La CPL incluye una serie de definiciones útiles para su interpretación: *contribuidor*, "*patentes sujetas a licencia*", y el "*programa*". En particular, la CPL define las contribuciones como "modificaciones y componentes agregados al programa que son obras derivadas de él". Por lo tanto, la licencia solamente se aplica a software nuevo que cumple este doble test.

### Aspectos fundamentales

La CPL otorga amplios derechos de explotación sobre el programa: la reproducción, la modificación, la distribución y la comunicación pública (*perform y display*). Por otro lado, especifica una serie de obligaciones relevantes para el *copyleft*:

- La redistribución del código objeto se debe hacer bajo una licencia compatible con la original y debe indicar cualquier diferencia (cláusula 3) Se puede redistribuir el programa (y obras derivadas) en binario, solamente si el redistribuidor provee un mecanismo para que el destinatario puede recibir el código fuente.
- La redistribución en código fuente debe hacerse bajo la misma licencia.

### Aspectos particulares

La CPL es una licencia de segunda generación e incluye una serie de términos no recogidos en las licencias originales (GPL y BSD).

Por un lado, incluye una licencia específica de patente sobre las contribuciones y las combinaciones de una contribución con el programa original. Esta licencia de patente es revocada en caso de demandas basadas en patentes. Este tipo de condición se llama *patent peace*, y la encontramos en casi todas las nuevas licencias desde la MPL: la CDDL, la GPLv3 o la OSL, que vamos a ver a continuación. La *patent peace* de la CPL es particular a la situación de IBM como titular del mayor portafolio de patentes del mundo y determina que las licencias de patente de la CPL se resuelvan en dos situaciones:

- Una acción basada en patentes contra el contribuidor, respecto a **cualquier** software;
- Una acción basada en patentes contra **cualquier** persona, respecto al software cubierto por la licencia.

Por otro lado, es la única licencia de software libre que contiene una indemnización entre contribuidores: los redistribuidores **comerciales** deben indemnizar a cualquier otro contribuidor contra las pérdidas que puedan surgir a raíz de la distribución comercial (excepto en lo que se refiere a la propiedad intelectual). Se adecua más al marco legal de la protección del consumidor, bajo el cual las exclusiones de garantías y responsabilidades no son completa-

mente válidas. En este caso, una distribución comercial (a consumidores) podría exponer a los otros autores contribuidores a riesgos no previstos cuando hicieron su contribución.

## Comentarios

La licencia CPL es una licencia muy bien redactada desde la perspectiva legal y deja mucho menos lugar a dudas que la GPLv2, por ejemplo. Las definiciones son claras y el alcance de los derechos y las obligaciones, también. Nuestro comentario principal es que la licencia es incompatible con la GPLv2 por la obligación de licenciar cualquier patente de los contribuidores y de compensar a coautores contra las demandas de usuarios comerciales. *A priori*, entendemos que sigue siendo incompatible con la GPLv3, a pesar de que ésta tiene ahora una licencia de patente muy similar, por la indemnización comercial.

### 2.2.4. Otras licencias con *copyleft* robusto

En este último apartado sobre las licencias con *copyleft*, seguiremos con nuestra tabla analítica que nos ha ayudado a definir varias licencias de software libre que incluyen obligaciones de tipo *copyleft*. Algunas son compatibles con la GPL; por lo tanto, su código se puede mezclar con código bajo GPL y el resultado se puede distribuir sin problema. Otras, por las obligaciones adicionales que imponen, no son compatibles con la GPL. Las resumimos en la tabla siguiente:

#### Lecturas recomendadas

Hay varios análisis de licencias de software libre en Internet. Podéis consultar **R. Brooks**, *Open source licenses overview*; **Electronic Freedom Foundation**, *Guide to licenses*, o **S. Hackv  n**, *A quick survey of open source licenses* (en la bibliograf  a).

#### Licencias con *copyleft* compatibles con la GPL

eCos license 2.0 y Classpath	Es una licencia de la FSF sobre el <i>embedded configurable operating system</i> . B��sicamente, consiste en la GPL m��s una excepci��n que permite enlazar el programa con otros programas que no est��n bajo la GPL y con efectos muy similares a la LGPL. Aunque se integre por compilaci��n o enlace con un programa propietario distribuido en binario, se debe proporcionar o poner a disposici��n del usuario el c��digo fuente de eCos. Classpath tiene la misma excepci��n –y es interesante destacar que Sun ha publicado gran parte de la plataforma Java bajo la licencia GPL con la excepci��n Classpath.
Aladdin Free Public License (AFPL)	La Aladdin Free Public License (AFPL), relativa a Ghostscript, merece una menci��n especial, porque tiene un car��cter particular. No cumple la OSD, aunque se inspira directamente en la GPL. Lo interesante es que mientras que la ��ltima versi��n disponible de Ghostscript se distribuye bajo la AFPL y obliga a obtener una licencia propietaria para usos comerciales, la pen��ltima versi��n del software se libera bajo la GPL. Por lo tanto, se comercializa la "mejor" versi��n del programa y los desarrolladores libres pueden aprovechar el c��digo un poco m��s antiguo.
Sleepycat Software Product License (Berkeley Database)	Es una licencia que se aplica, sobre todo, a un motor de base de datos de la empresa Sleepycat (antiguamente Berkeley Database). Sigue el modelo simple de la licencia BSD, que comentamos a continuaci��n, y agrega una obligaci��n de distribuir o poner a disposici��n el c��digo fuente <b>del software</b> y de cualquier otro programa <b>que utilice el software</b> . Asimismo, dicho programa debe ser libremente redistribuible bajo t��rminos razonables (el <i>copyleft</i> ). Las licencias abiertas y libres son consideradas razonables, incluso la GPL.

#### Licencias incompatibles con la GPL

### Licencias con *copyleft* compatibles con la GPL

<b>Affero 1.0</b>	<p>Affero es un software para gestionar y extender las comunidades virtuales con funcionalidades de <i>rating</i> y comercio electrónico. La licencia es una variación de la GPLv2 redactada con la ayuda de la FSF. La licencia cubre el caso de la arquitectura de programas distribuidos en redes o servicios enlazados por la vía de servicios web. En este caso, el usuario licenciario no recibe el programa como distribución de software, sino como un servicio por medio de la web, y puede ofrecer el mismo servicio a terceros, evitando las obligaciones de <i>copyleft</i> de la cláusula 2b. La licencia Affero agrega a la GPLv2 una cláusula "2d" que dice que si en el caso de un servicio ofrecido por red el programa original tiene una función para proveer el código fuente también vía web, el licenciario no puede eliminar esa función y debe ofrecer acceso por web al código fuente de la obra derivada.</p> <p>Es incompatible con la GPLv2, porque esta adición crea una obligación más restrictiva que la GPL. Esta licencia ha sido criticada por restringir las comunicaciones de red al protocolo HTTP, cuando en el futuro puede haber otras formas de comunicación. La OSI también critica este aspecto, porque el acceso al código no debe estar vinculado a una tecnología en particular (directriz 10).</p>
<b>Affero GPLv3</b>	<p>La nueva licencia Affero GPLv3 es básicamente la GPL con un pacto adicional que cubrirá el mismo escenario que el mencionado respecto a la Affero 1.0. En este caso (ASP) se debe proporcionar al usuario de los servicios remotos acceso al código fuente. La GPLv3 es explícitamente compatible con la Affero GPLv3 y viceversa.</p>
La licencia OpenSSL / SSLeay	<p>Se aplica a programas de seguridad SSL. Es una combinación de las licencias Open SSL y SSLeay. Está modelada sobre la licencia BSD, que comentamos a continuación, y agrega al final de la licencia SSLeay una cláusula de <i>copyleft</i> en la que obliga a cualquier obra derivada a distribuirse en los mismos términos. Se prohíbe expresamente mezclar este código con código bajo la GPL. También es incompatible con la GPL porque tiene una cláusula con respecto a la publicidad y la atribución a los autores (que proviene de la versión anterior de la BSD y la Apache).</p>

## 2.3. Las licencias con *copyleft* "suave" o "híbridas"

En este apartado, comentamos las licencias libres que llamamos híbridas o con *copyleft* suave. Se diferencian del *copyleft* fuerte en lo que permiten su integración, uso y redistribución en programas bajo otras licencias, pero mantienen su propio código bajo *copyleft*.

### 2.3.1. La Licencia Pública General Menor o de Bibliotecas GNU (LGPL)

La Licencia Pública General Menor (o de Bibliotecas) GNU es la segunda licencia redactada por la Free Software Foundation. Inicialmente, esta licencia se llamó "Library GPL", puesto que fue diseñada expresamente para ser aplicada a bibliotecas informáticas. Luego, la FSF cambió su nombre a "Lesser GPL" ('GPL menor'), porque consideraba que garantiza menos libertad que su hermana mayor, la GPL. Su versión 2.1 es de febrero de 1999, y en junio de 2007 se publicó la versión 3.0, que es una variante de la GPLv3, comentada más arriba.

En los apartados anteriores hemos dicho que cuando un programa se enlaza con un componente de software, ya sea estáticamente o mediante un componente o una API compartidos de manera dinámica, la combinación se considera una obra "basada en" o "derivada del" software original. Si el software está bajo la GPL, muchos argumentan que este enlace obligará a distribuir todo el



programa final bajo la GPL. La LGPLv2 se creó específicamente para permitir que se enlazaran algunos componentes de software libre –las bibliotecas– con programas no libres, sin afectar al programa resultante. Por tanto, una biblioteca con LGPLv2 ofrece cierta comodidad o certeza para los desarrolladores de aplicaciones propietarias que quieran vincular sus programas con componentes bajo licencias libres, pero que teman el efecto *copyleft* de la GPL.

## Comentarios

La LGPLv2 deriva de la GPLv2 y la mayoría de sus términos son similares a los de ésta, por lo que nos remitimos al apartado sobre la GPL (tanto a la versión 2 como a la versión 3) y a sus fichas resumen. Aquí comentamos solamente los elementos diferenciadores.

## Definiciones útiles

Como la GPL, la LGPLv2 define *programa* y *código fuente*. Además, incluye tres definiciones nuevas:

- **Biblioteca:** consiste en un conjunto de componentes de software destinados a enlazarse con programas (que usan las funciones incorporadas en las bibliotecas) para crear un ejecutable.
- **Obra basada en una biblioteca:** recoge la definición de *programa* en la GPL y significa la biblioteca original o cualquier obra derivada de ella según la definición del derecho de *copyright*, es decir, una obra que la contenga o contenga una parte de la misma.
- **Obra que usa una biblioteca:** es una obra separada que no contiene ninguna parte u obra derivada de la biblioteca, sino que se destina a ejecutarse con la biblioteca por medio de la compilación o los enlaces.

## Elementos esenciales

Respecto a la misma biblioteca y sus modificaciones, las condiciones aplicables son las de la GPL. La principal diferencia con la GPL es que la LGPL permite distribuir **sin restricción** un ejecutable que consista en la compilación de, por un lado, obras que usan la biblioteca, y otro, la misma biblioteca (cláusula 6). Ésta es la excepción a la cláusula de *copyleft* habitual de la GPL. Sin embargo, se debe permitir al destinatario modificar el programa (incluso la obra "que usa la biblioteca") para su uso particular y para realizar operaciones de ingeniería inversa con el objetivo de corregir errores (por lo tanto, se argumenta que aunque no haya *copyleft*, igualmente hay que proporcionar el código fuente).

Hay que observar también que en las cláusulas 5 y 6 se da un tratamiento bastante minucioso de diferentes formas de enlazar un programa con las bibliotecas, para el cual remitimos al estudiante a la licencia misma.

Como condición adicional, se permite convertir la licencia LGPL aplicada a su biblioteca a la licencia GPL en cualquier momento (no se puede volver atrás) (cláusula 3).

## Comentarios

Por su vocabulario, la LGPL está destinada al uso para bibliotecas. Pero no está restringida a éstas, ya que hay otros programas que se distribuyen con esta licencia (por ejemplo, OpenOffice.org). Los autores de un software son libres de elegir la licencia que quieran, sea cual fuere su programa.

La FSF no recomienda el uso de la LGPL, excepto por razones estratégicas: la utilización de la LGPL permite la distribución y el uso más amplio de su código, y por lo tanto, favorece establecer un componente –una biblioteca, un módulo de programa, etc.– como un estándar en el sector. Sin embargo, la LGPL no favorece el desarrollo de aplicaciones libres, un objetivo fundamental para la FSF, y por ello no recibe su plena aprobación.

Como comentario práctico hemos de decir que, dentro de los límites de las cuestiones técnicas del tipo de enlace entre dos programas, se pueden combinar, integrar y distribuir bibliotecas bajo LGPL con software bajo cualquier otra licencia, incluso propietaria.

Un ejemplo de este tipo de software es la biblioteca de C (libgcc) que se distribuye con Linux, que se puede usar para desarrollar programas propietarios que se ejecutan sobre Linux.

## Uso estratégico

"El uso de la LGPL para la biblioteca C o para cualquier otra biblioteca es un tema de estrategia. La biblioteca C hace un trabajo genérico; todo sistema propietario o compilador viene con una biblioteca C. Por lo tanto, hacer que nuestra biblioteca estuviera sólo disponible para el software libre no le hubiera dado al software libre ninguna ventaja: sólo hubiera desalentado el uso de nuestra biblioteca. No hay ninguna razón ética para permitir aplicaciones propietarias en un sistema GNU, pero estratégicamente, parece que si no se permite, ello contribuirá más a desalentar el uso del sistema GNU que a alentar el desarrollo de aplicaciones libres".

R. Stallman, The GNU operating system and Free Software Movement, Open Sources, Voices from the Gen Source Revolution, O'Reilly, 1999.

Hacemos notar también que la licencia LGPLv2 es compatible con la GPLv2, pero no con la GPLv3 ni con la LGPLv3.

## LGPLv3

La LGPLv3 es una variante explícita sobre la GPLv3, es decir, es la GPLv3 con permisos adicionales. Estos permisos autorizan el uso de la biblioteca en cuestión por un programa tercero y la licencia "del todo" bajo una licencia que no sea la LGPL. Además, no se aplica la cláusula 3 sobre sistemas DRM.

### Combinaciones

La combinación de los programas puede efectuarse por:

- usar los datos de cabecera de la biblioteca –en este caso, sólo hace falta indicar la existencia de la biblioteca–;
- combinar los programas juntos –en este caso, hay una serie de obligaciones que permiten al destinatario tener el código fuente de la biblioteca (bajo la LGPL) y el código fuente de la aplicación (no sujeto a la LGPL) y reinstalarlo todo después de cualquier modificación–;
- poner juntas las funciones de la biblioteca original y otras funciones (nuevas) en una "biblioteca combinada" –en este caso hay que proporcionar la biblioteca original.

En todos los casos, hay que avisar de la existencia y el régimen de licencia (LGPLv3) de la biblioteca y proporcionar una copia de la misma.

### 2.3.2. La Mozilla Public License

La Mozilla Public License (MPL) se desarrolló junto con la Netscape Public License en 1998, cuando Netscape "abrió" (como software abierto) el código de su navegador de Internet, Netscape Navigator. El desarrollo de la licencia fue un proceso colaborativo entre varios de los "gurús" del movimiento abierto, como Linus Torvalds, Bruce Perens y Eric Raymond. Éstos intentaron, en un principio, persuadir a Netscape para que empleara la GPLv2, pero ante la negativa de Netscape y la necesidad de respetar la propiedad intelectual de terceros, acabaron distribuyendo el código bajo la NPL.

#### Consultando la comunidad

Antes de abrir su código fuente al público, Netscape distribuyó un borrador de la licencia propuesta en un foro de noticias (*newsgroup*) especialmente creado para opinar sobre ésta ([netscape.public.mozilla.license](http://netscape.public.mozilla.license)). El proceso de desarrollo abierto para el software fue trasladado al mundo legal y despertó gran entusiasmo y... críticas. Hubo varias propuestas para modificar algunos términos de la NPL, sobre todo el que permitía a Netscape utilizar el mismo código en otros productos que no estaban bajo la NPL. Este proceso lo ha seguido la Free Software Foundation para la redacción de la GPLv3.

Al final, buscando un equilibrio entre los objetivos comerciales y los de desarrollo libre de Netscape y la comunidad libre, se resolvió emitir dos licencias: la NPL y la MPL. La primera se aplicó al código inicial de Navigator y a las *modificaciones* hechas a éste, y no se usa más. La segunda se aplicó a cualquier software *agregado* al código y a cualquier programa totalmente nuevo que quisiera usar esta licencia. Ahora se usa la licencia MPL para varios programas, entre los cuales encontramos el navegador Firefox y otros programas de Mozilla.org. Las dos licencias son idénticas, excepto por unos derechos reservados por Netscape en la NPL para su código inicial, con valor "histórico" solamente.

#### Lectura recomendada

La historia de Mozilla está en C. DiBona y otros (ed.) (2001), *Open sources: voices*.

Aspecto	Contenido	Comentarios
Modelo original	Licencia original, de tipo "empresarial".	Es la primera licencia libre desarrollada por una empresa comercial (Netscape).
Objeto de la licencia	"Código cubierto": ficheros originales y sus modificaciones.	No incluye ficheros agregados.
Derechos otorgados	Copia, modificación, distribución.	
Atribución	Incluir en código fuente.	
Copyleft	Parcial: obligación de aplicar la MPL únicamente a cualquier "código cubierto", no a programas mayores que consisten en programas que "usan" o que "incluyen" los ficheros originales.	Tiene un efecto similar a la LGPL y permite la distribución bajo cualquier licencia de "obras que usan software bajo MPL".
Código fuente	Incluye "scripts para creación de ejecutables", API, etc. Debe "seguir" cualquier distribución de binario (sin obligación de distribuir a cualquier tercero).	
Otras obligaciones	Incluir legal.txt con comentarios sobre derechos, reclamaciones o demandas relativos al código.	Es un fichero muy útil para identificar cualquier problema legal.
Garantías/responsabilidades	Excluidas/limitadas en la medida permitida por ley.	
Versiones	Permite actualizarse (actualización que controla la Mozilla Foundation).	
Patentes	Licencia explícita sobre código original y contribuciones <i>patent peace</i> amplia: revocación de la licencia en caso de cualquier reclamo basado en derecho de patentes contra procesos implementados por <b>cualquier</b> software de los titulares originales (no solamente el código cubierto).	
Jurisdicción / derecho aplicable	Tribunales de Santa Clara y derecho aplicable de California.	
Otros	Permite especificar partes del código distribuido bajo MPL y otras licencias.	

## Componentes esenciales de la MPL

a) **Definiciones.** La MPL tiene una estructura de licencia de software clásica y empieza con definiciones importantes que permiten, entre otras cosas, diferenciar entre lo que es código original y lo que es código agregado.

- **Desarrollador inicial:** en el caso de la NPL, Netscape; en código bajo MPL, el autor inicial indicado en el anexo de la licencia y cualquier autor de contribuciones.
- **Código inicial:** código distribuido por los desarrolladores iniciales.
- **Modificación:** cualquier modificación al código cubierto que **no** incluya una simple adición de un fichero nuevo separado o código nuevo que interactúe con el código original sin modificarlo (por ejemplo, por medio de una API –aunque la API misma podría ser una modificación, si está integrada en el código cubierto. La palabra *modificación* no se refiere tampoco a toda la obra modificada (tal como pasa en la GPL), que también podría ser una "obra derivada" en el derecho, sino únicamente a la parte modificada.
- **Código cubierto (por la licencia):** código inicial más las modificaciones.

- **Contribuidor:** cualquier tercero que modifique código cubierto.
- **Obra mayor:** una obra separada del código cubierto pero que lo puede incorporar o con el que se puede enlazar, sin modificarlo (cláusula 3.7).

La definición completa de los diferentes tipos de código permite distinguir entre los ficheros que están sujetos a la MPL ("código cubierto") de aquellos que se pueden mantener separados y eventualmente privatizar. El sentido de "modificación", resumido aquí, explicita muchas cosas que la GPLv2 no dejaba claras –sobre todo la cuestión de ficheros nuevos adicionales que no modifican ninguna parte del código inicial en el momento del desarrollo. Por lo tanto, permite a un desarrollador agregar ficheros y programas separados (propietarios o libres) y distribuirlos separados del código cubierto, pero parte de un programa más grande (potencialmente propietario).

## b) Los derechos otorgados

- El desarrollador inicial otorga, en primer lugar, una licencia para usar, reproducir, modificar y distribuir libremente el código y, en segundo lugar, una licencia de patente suficiente como para permitir el uso del programa y las modificaciones (cláusula 2.1).
- Cada contribuidor otorga licencias similares relativas a su contribución o modificación (cláusula 2.2).
- Se puede distribuir el código en binario bajo una licencia compatible con la MPL, siempre que se respeten las obligaciones contenidas en la licencia, por ejemplo, el acceso al código fuente (cláusula 3.6).
- Se puede incorporar el código cubierto en una "obra mayor" (que lo incluya, pero no lo modifique) bajo cualquier licencia, siempre que se respeten las obligaciones relativas a la parte de código cubierto (cláusula 3.7), por ejemplo, el acceso al código fuente.

## c) Obligaciones

- El código fuente del código inicial y de cualquier modificación (código cubierto) debe distribuirse bajo la MPL, sin cláusulas más restrictivas (*copyleft* para el código cubierto, cláusula 3.1).
- Si se distribuye el código cubierto en binario, se debe ofrecer acceso a su código fuente al destinatario de la distribución durante por lo menos doce meses (cláusula 3.2).
- Hay que acompañar cualquier modificación con una copia de la licencia y una indicación de las modificaciones y sus autores, así como una indi-

cación de cualquier reclamo conocido sobre el código (legal.txt) (cláusula 3.3-3.5).

#### d) Otros elementos relevantes

La licencia MPL es una licencia completa –imitada en cierta manera por la CDDL, la CPL, la OSL, y ahora la GPLv3.

##### Cláusulas de los MPL

- el cumplimiento en caso de inaplicabilidad de parte de la licencia (cláusula 4),
- las versiones y el renombramiento de variantes de la licencia (cláusula 6),
- la ausencia de garantías (cláusula 7) y la limitación de responsabilidad (cláusula 9),
- la resolución de la licencia en caso de incumplimiento o litigio con otro contribuyente relativo a patentes (cláusula 8.2),
- el derecho aplicable y la resolución de conflictos (cláusula 11),
- una declaración de responsabilidad mutua de los coautores (cláusula 12).

Todavía más importante, la versión 1.1 de la licencia MPL incluye una cláusula adicional (la 13) que permite al autor inicial licenciar el código cubierto o las partes designadas de éste, bajo **licencia múltiple** (al principio, estaban previstas la GPLv2 y la LGPL). Esto permite el uso comercial, pero además algo más interesante: la posibilidad de distribuir el código bajo la GPL. Así, la nueva versión de la licencia es compatible con la GPL en relación con las partes del software designadas de esta manera.

#### Comentarios

La MPL es una licencia mucho más compleja y completa que la GPLv2 y, obviamente, que la BSD. Fue redactada con y por abogados en el contexto de una empresa comercial, por lo que incluye definiciones específicas y recoge cuestiones tradicionales relacionadas con las licencias, como la jurisdicción competente, el derecho aplicable, etc. Aunque su efecto puede parecer más cercano a la BSD que a la GPL, hay varios puntos importantes que hemos de considerar y que comentamos en este apartado:

**a) Persistencia y *copyleft*.** La MPL tiene un efecto de persistencia parcial, como la LGPL: el código cubierto (incluyendo cualquier modificación) debe mantenerse bajo la licencia MPL, mientras que cualquier extensión (una obra mayor) puede ser propietaria. Además, es muy fácil crear un archivo adicional propietario que llame al código original bajo MPL y distribuirlo todo bajo una licencia propietaria. Esto sigue la filosofía de la licencia BSD. Sin embargo, en todos los casos, el código fuente de la parte libre original debe distribuirse u ofrecerse al destinatario. Todo esto queda ilustrado a continuación:

**b) Compatibilidad con la GPLv2 y la GPLv3.** Cualquier software bajo la licencia MPL 1.0 (y la MPL 1.1 sin licencia alternativa) es incompatible con la GPLv2 o la GPLv3; fundamentalmente, porque tiene demasiadas restricciones adicionales relativas a las patentes (aunque la GPLv3 se acerca en ese aspecto) y la posibilidad de enlazarse con programas propietarios, entre otras cosas. La posibilidad de licencias múltiples ofrecida por la versión 1.1 facilita la compatibilidad si se elige la GPL como una licencia alternativa, (por ejemplo el código fuente de los programas de Mozilla.org).

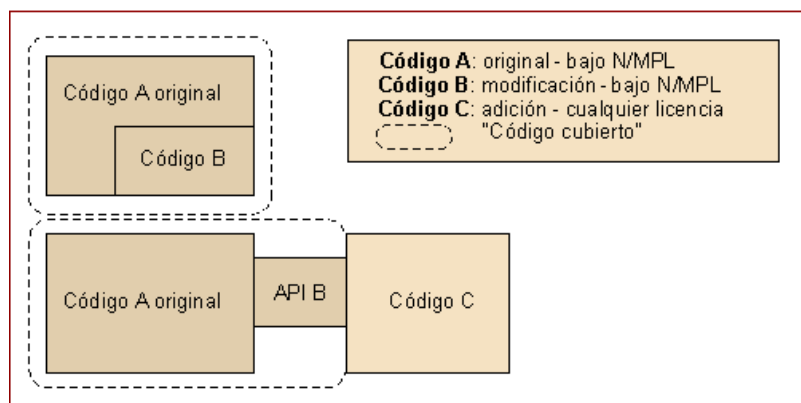


Figura 3. Ilustración de la persistencia en la licencia MPL

#### Contenido complementario

Cualquier código bajo la licencia MPL 1.0 es incompatible con la GPL.

**c) Las cláusulas de patentes.** Como ya hemos visto en relación con la GPLv3 y la CPL, la cláusula resolutoria (aquí, la 8), combinada con la licencia de patentes (cláusula 2.1), es parte de una nueva generación de cláusulas en las licencias libres para crear un entorno de trabajo libre de patentes y libre del riesgo de patentes. Constituye lo que se llaman "las licencias cruzadas de patentes". Los desarrolladores no pueden impedir que una persona solicite y obtenga (en Estados Unidos) una patente sobre un proceso que puede ser parte de una modificación del programa inicial. El riesgo es que el uso o una modificación posterior del software podrían violar una patente si el usuario no tiene una licencia de patente adecuada. Por lo tanto, estas cláusulas intentan hacer dos cosas:

- Por un lado, la persona "patentadora" debe otorgar a todos los otros licenciarios (usuarios y desarrolladores) una licencia de patente respecto al proceso o código patentado incluido en su contribución.
- Por el otro lado, se rescindirán las licencias de derechos de autor (y de patente, si las hay) otorgadas a esta persona "patentadora" en el caso de cualquier litigio o intento de impedir la explotación libre de la modificación.

**d) El equilibrio comercial.** Los conceptos de *modificación* y de *obra mayor* han sido cuidadosamente elaborados para encontrar un equilibrio entre la libertad de la BSD, que permite un uso ilimitado del código, y la libertad de la GPL, que obliga a mantener todo el código y las obras derivadas libres, es decir, entre la promoción del desarrollo de software libre por empresas comerciales y la protección del trabajo de desarrolladores "libres". Este justo medio ha sido

definido por la diferencia entre una **modificación** y una **adición**. Recordemos que la GPL, en contraste, afecta a las adiciones que se vinculan de manera íntima con software bajo GPL.

e) **El legal.txt**. Es un fichero donde los contribuidores deben consignar cualquier aviso de reclamo, litigio o restricción sobre una parte del código. Demuestra un conocimiento evidente del proceso de desarrollo libre, en el que el riesgo de denuncias relativas a la propiedad intelectual e industrial es alto y la información transparente es primordial. Un desarrollador posterior debería utilizar este fichero para estudiar las limitaciones legales de un código provisto por terceros, quizás en relación con un litigio de patente, quizás por las limitaciones de ciertas partes de código que puedan estar bajo una licencia compatible pero diferente de la MPL...

### Ejemplo de la creación de licencias libres

Las labores de Netscape, y ahora la Mozilla Foundation, nos pueden enseñar varias cosas en relación con la creación de licencias libres.

Demuestran una tendencia de los participantes comerciales en el movimiento libre a redactar sus propias licencias. IBM tenía la IBM Public License, ahora transformada en CPL. Sun, Apple y Microsoft también han intentado formular unas variantes más comerciales (que comentaremos más adelante). A veces se trata de "variaciones" para mejorar aspectos inciertos, como el tema de obra derivada en la GPL o las patentes, otras veces regulan temas en particular. Actualmente, hay un movimiento tanto en contra de las licencias "patrocinadas" (y por lo tanto, a favor de licencias neutras o template, como la CDDL o la CPL) y en general contra cualquier licencia nueva.

El proceso de creación de la MPL y el rechazo de los derechos reservados por Netscape demuestran que hay realmente un "efecto de comunidad" en el movimiento libre. Como consecuencia, una licencia inadecuada será rápidamente rechazada. La historia de las licencias Shared Source de Microsoft y Sun lo confirma.

Finalmente, la MPL es casi un modelo de licencia libre por excelencia, por sus orígenes mixtos (empresa comercial, movimiento libre, expertos legales) y por sus objetivos de desarrollo y explotación. Se adecua a muchos contratos y licencias comerciales, por lo que las empresas se sienten más cómodas con ella.

### 2.3.3. La Open Source License (OSL)

La Open Source License (OSL, ahora en su versión 3.0) es una licencia con *copyleft* suave redactado de manera neutra por el asesor legal de la OSI, Lawrence Rosen. Es una licencia completa (definiciones, licencia explícita de los diferentes derechos, etc.) y se adecua más que otras al marco legal de la propiedad intelectual en Europa.

Ficha resumen

Aspecto	Contenido	Comentario
Modelo	CPL/MPL (con elementos originales).	El autor es Lawrence Rosen y la licencia responde a un esfuerzo por crear una licencia <i>copyleft</i> genérica.
Objeto de la licencia	"Programa" y "obras derivadas" definidas por ley (derecho de autor/ <i>copyright</i> ).	



Aspecto	Contenido	Comentario
Derechos otorgados	Copia, modificación, distribución, comunicación pública ( <i>perform</i> y <i>display</i> ).	
Atribución	En código fuente.	
<i>Copyleft</i>	Débil: debe aplicarse la OSL a cualquier distribución del programa original y obras derivadas (definidas por ley).	No se aplica a obras que usan el programa (efecto similar a la LGPL y a la MPL).
Código fuente	Código fuente y documentación sobre cómo modificar el programa. Sigue el binario al destinatario (no en general).	
Otras obligaciones	Ninguna en particular.	
Garantías/responsabilidades	Garantía de título; las demás garantías están excluidas. Responsabilidades limitadas.	Limitaciones más legales desde la perspectiva europea (no excluye muerte o daños físicos).
Versiones	Modificable por cualquiera, eliminando el título y el autor (L. Rosen).	
Patentes/marcas	Licencia explícita sobre código original y obras derivadas. <i>Patent peace</i> limitada a cualquier reclamo relativo a procesos patentados en relación con el programa (no cualquier programa de los autores).	
Jurisdicción / derecho aplicable	Flexible: jurisdicción y derecho de residencia del titular.	Principal principio de derecho internacional privado.
Otros	Licenciatario = persona / empresa / grupo de empresas (no incluye transferencias dentro de un grupo como una "distribución" para el <i>copyleft</i> ). Cubre distribuciones en modo ASP (debe ofrecer el código fuente al usuario).	

## Comentarios

La OSL es una licencia libre moderna y bien redactada, desde la perspectiva legal, que se acerca al marco legal europeo en cuanto al derecho de la propiedad intelectual y a las limitaciones respecto de las garantías y responsabilidades. A tal punto que un asesor legal de la Comisión Europea la recomendó como "base" para crear una licencia pública europea para el software *copyleft* de la Administración pública europea.

***Copyleft.*** La OSL 3.0 limita su efecto *copyleft* a las obras derivadas según la definición del derecho de la propiedad intelectual que se aplique en cada caso. El autor de la licencia argumenta que la GPLv2 trata de extenderse más allá de lo permitido por el mero derecho de autor (la reproducción, la modificación, la comunicación pública y la distribución) y podría verse limitada por una interpretación estricta del derecho. Por lo tanto, el alcance del *copyleft* de la OSL se encuentra estrictamente dentro del ámbito de derechos exclusivos de los autores bajo la propiedad intelectual. Esto permitiría, por ejemplo, enlazar

software bajo la OSL 3.0, como bibliotecas o con vínculos dinámicos, y la licencia no afectaría ("infectaría") al software que usara estas bibliotecas, en la medida que no fueran "obras derivadas" del software original.

**Otros elementos.** La definición de *derecho aplicable* y *foro competente* (a favor del licenciante) es más a favor de los autores y los distribuidores del software. Asimismo, con una garantía expresa de título sobre el software y cobertura en cuanto a dolo y daños personales, las limitaciones de garantías y responsabilidades serán más válidas en Europa. Finalmente, la distribución entre un grupo de empresas no se consideraría una distribución a efectos de las obligaciones de *copyleft*, pero sí la distribución de los servicios provistos por el software (en modo ASP o "SaaS") en cuyo caso habría que proporcionar al destinatario de los servicios una copia del código fuente.

### 2.3.4. Otras licencias con *copyleft* "suave" o "híbridas"

Presentamos en la tabla a continuación otras licencias libres que siguen el modelo o la filosofía de *copyleft* suave de la LGPL y la MPL. Es importante entender que estas licencias no son "intercambiables", aunque tengan un efecto *copyleft* bastante similar. Cada licencia debe entenderse y seleccionarse según su propia redacción y méritos, aplicados al caso concreto.

Licencias	Comentarios
<b>Apple Public Source License v. 2</b>	<p>Es una variación de la MPL creada por Apple, con nuevos elementos, como el derecho aplicable (California), y para cubrir la posibilidad de ofrecer servicios por Internet (<i>externally deployable</i>), similar a la Affero.</p> <p>Aunque las licencias iniciales de la Apple Public Source License no eran libres, se ha modificado la versión 2.0 para que lo sean: se ha eliminado la obligación de devolver a Apple cualquier modificación y la posibilidad de revocar en cualquier momento la licencia inicial (podéis ver más adelante un comentario breve sobre ello).</p> <p>Permite enlazar código bajo APSL 2.0 con programas no libres; por lo tanto, no es <i>copyleft</i>, y por la obligación de licenciar las patentes, es incompatible con la GPLv2.</p>
<b>CDDL</b>	<p>Se trata explícitamente de una versión genérica de la MPL creada por Sun Microsystems con algunas modificaciones y sin el nombre comercial Mozilla. Se usa para OpenSolaris, entre otros programas. Las principales diferencias son:</p> <ul style="list-style-type: none"> <li>• No incluye "<i>scripts</i> para creación de ejecutables" ni API etc, en la definición de código fuente.</li> <li>• En caso de distribución del binario, el código fuente debe publicarse generalmente (no limitado a destinatarios de distribución).</li> <li>• El legal.txt de la MPL ha sido eliminado.</li> <li>• La <i>patent peace</i> es limitada: la licencia de patente es revocada en caso de demandas basadas en patentes con respecto a procesos implementados por el código cubierto.</li> <li>• El derecho aplicable es flexible, definido por los titulares originales.</li> <li>• El <i>copyleft</i> incluye distribuciones de servicios del programa a clientes en modo ASP (hay que ofrecer las fuentes al destinatario del servicio).</li> </ul>

Licencias	Comentarios
EUPL	<p>La Licencia Pública de la Unión Europea es una nueva licencia (de enero de 2007), expresamente redactada para la liberación de software de la Administración pública europea y de los países miembros de la Unión Europea. El alcance del <i>copyright</i> es similar al de la OSL, tiene una licencia de patente y las limitaciones de garantías y responsabilidades son válidas dentro del marco general de la protección del consumidor y los contratos de adhesión de la Unión Europea.</p> <p>Con el propósito de establecer una compatibilidad expresa con otras licencias <i>copyleft</i>, tiene un pacto de compatibilidad con otras licencias incluidas en un anexo (actualmente la GPLv2, la LGPLv2, la OSL, la CPL y la CeCILL, un licencia <i>copyleft</i> francesa): en caso de mezclar software bajo la EUPL con software bajo estas licencias, se podrá distribuir el software bajo la licencia nueva.</p> <p>La Comisión Europea ha publicado traducciones oficiales en las lenguas de la Unión Europea.</p>

### 3. Otras licencias de tipo "libre"

En el apartado anterior, hemos estudiado en profundidad las principales licencias libres y hemos comentado sus rasgos particulares, sus compatibilidades y sus consecuencias. En este nuevo apartado, queremos completar nuestro análisis de las licencias "libres". Comentaremos en orden las siguientes:

- 1) las licencias que llamaremos "pseudolibres", que tratan de emular a las libres pero que contienen alguna restricción que no cumple las libertades de la FSF o las directrices de la OSD;
- 2) las licencias de documentación libre;
- 3) las licencias de tipo *freeware* y *shareware*, que no son para nada "libres".

#### 3.1. El auge y la caída de las licencias de software "pseudolibres"

Aunque en este módulo enfocamos las licencias de software libre, es interesante hacer un breve análisis de las licencias creadas por empresas comerciales que intentan beneficiarse del modelo de desarrollo libre –sin pagar todos sus "costes". En primer lugar, ello indica el abanico de posibilidades entre lo libre y lo propietario. Asimismo, permite elucidar la posición de dichas empresas al respecto e indicar algunas estrategias que hay que evitar desde el punto de vista de las licencias libres. Observamos que el papel de las licencias Shared Source se ha visto reducido, por la confianza y la popularidad que han ganado las licencias de software verdaderamente libres, y las críticas que recibieron aquéllas en su momento.

##### 3.1.1. La Sun Community Source License (SCSL)

La licencia SCSL era una tentativa de ofrecer acceso al código y a entornos de programación de Sun Microsystems Inc., por ejemplo Java o Jini, y establecerlo como un estándar. En esto, ha tenido mucho éxito, sobre todo en relación con Java. Los "componentes" incluidos en la Sun Community License eran el J2EE, el Java Developers Kit (JDK), el Personal Java y el Embedded Java, entre otros. Sin embargo, la época *shared source* de Sun casi ha terminado, ya que en noviembre de 2006 Sun Microsystems liberó bajo la GPL (con la excepción de Classpath) la mayoría de los programas que componen el entorno tecnológico de Java.

La SCSL era, sobre todo, una licencia para desarrolladores. Se "abre" principalmente a los fines de investigación y desarrollo, pero permitía a Sun mantener un control muy fuerte sobre la evolución del programa y los entornos de programación.

Conceptualmente, es una licencia a medio camino entre la MPL y una licencia propietaria: permite correcciones, modificaciones y extensiones, pero cualquiera de ellas debe ser devuelta a Sun.

Los licenciarios bajo la SCSL tienen derechos en tres categorías:

- 1) **para la investigación**, el uso es libre, pero hay que enviar y licenciar a Sun cualquier corrección de error y las API para extensiones;
- 2) **para el uso interno**, cualquier distribución de código debe ser compatible con los criterios técnicos de Java (hay que comprar una licencia para el uso del logotipo de Sun y pasar los exámenes técnicos de Sun), y
- 3) **para el uso comercial** (para desarrollar software para clientes), hay que cumplir con el punto 2 y concluir un contrato de soporte con Sun. Sin embargo, se puede comercializar el código objeto bajo cualquier licencia. La licencia incluye un acceso a las especificaciones técnicas de las tecnologías Sun, pero no se permite hacer una reingeniería inversa (bajo amenaza de patente).

Como vemos, debido a las obligaciones adicionales (sobre todo, la de devolver a Sun cualquier modificación y cumplir las especificaciones técnicas de Sun), esta licencia no es libre. Es más, ha suscitado una serie de críticas por ser "falsa", por tratar de venderse como una plataforma libre y por permitir que Sun se aproveche de las modificaciones, los arreglos y las correcciones de errores (*bug fixes*) de terceros desarrolladores. Por otro lado, la obligación de cumplir los criterios de Sun permite a ésta mantener un cierto control de calidad sobre el uso de su tecnología.

En 2006-2007, bajo la presión de la comunidad libre, el surgimiento de nuevos proyectos libres para crear tecnologías Java para reemplazar el software de Sun y la aceptación dentro de Sun de los beneficios del software libre, la empresa comienza a adoptar una posición más favorable al software libre. Primero, libera Opensolaris, su sistema operativo, bajo la licencia CDDL y crea un proyecto y una comunidad alrededor del software. Luego, renuncia a usar su parte de portafolio de patentes (*patent pledge*) contra los usuarios de Opensolaris. Finalmente, en noviembre de 2006 libera sus tecnologías Java bajo licencia GPLv2, con la excepción de Classpath (que permite usar las bibliotecas sin efecto *copyleft*).

#### Lectura recomendada

Podéis ver en la bibliografía: Gabriel, Richard y Joy (1999), *Sun community source license principles*, y S. Hackv n (1998), *Not quite open source, but closer*.

### 3.1.2. Microsoft Shared Source Initiative (MSSI)

Microsoft también creó una serie de más de diez licencias "semilibres" para una parte de sus programas. Se aplicaban al sistema operativo CE para dispositivos portátiles, CLI (Common Language Infrastructure) y las especificaciones de C#, y también incluyeron elementos de Windows 2000 y XP. Este "gesto" permitía sobre todo el estudio académico de las tecnologías en cuestión y, para las empresas comerciales que crean productos que se ejecutan sobre estas plataformas, integrar mejor sus programas con los de Microsoft. Asimismo, permitía a Microsoft revelar el código fuente de varias aplicaciones a organizaciones gubernamentales bajo condiciones muy estrictas de secreto.

Había varios tipos de licencia dentro de su iniciativa Shared Source. El modelo básico, por ejemplo la licencia Shared Source de CE, abría el código a investigadores y estudiantes: se podía descargar y estudiar el código fuente, usar, modificar y distribuir las modificaciones del código solamente para cualquier uso no comercial, siempre que se mantuviera la misma licencia. Luego, con el Windows CE Shared Source Premium Licensing Program, los fabricantes de aparatos OEM tenían acceso al código fuente de Windows CE y el derecho de modificarlo y distribuir las modificaciones de manera comercial. Sin embargo, debían licenciar cualquier modificación a Microsoft gratuitamente, lo que permitía a ésta incorporar dichas modificaciones en versiones posteriores del software después de un período de seis meses.

Otras licencias de MSSI tienen variaciones sobre estos derechos otorgados y reservados. La licencia para ASP.net, por ejemplo, permite cualquier uso comercial y no comercial, pero prohíbe combinar o distribuir el programa ASP.net con programas libres y sobre todo bajo condiciones de *copyleft*.

En octubre de 2005, Microsoft redujo sus licencias Shared Source a cinco: tres licencias básicas y dos variantes limitadas a la plataforma Windows. Las tres licencias básicas son:

- **Microsoft Public License (Ms-PL).** Es una licencia permisiva, *copyleft* para distribuciones en formato de código fuente pero permisiva para distribuciones en formato de binario. Tiene una variante limitada a tecnologías para Windows. Aprobada por la OSI, y compatible con la GPLv3.
- **Microsoft Reciprocal License (Ms-CL).** Es una licencia recíproca o *copyleft*, con efecto similar a la licencia Mozilla: el efecto *copyleft* se define en función de los ficheros originales y se pueden distribuir los ficheros de una "obra mayor" (que usa los ficheros originales) bajo cualquier licencia. También tiene una variante limitada a tecnologías para Windows. Aprobada por la OSI, no compatible con la GPL.

#### Web recomendada

Para más información sobre Shared Source, podéis consultar <http://www.microsoft.com/resources/sharedsource/default.aspx>.

- **Microsoft Reference License (Ms-RL).** Es una licencia similar a las antiguas licencias Shared Source que permite copiar el programa para usos internos, pero no modificarlo ni distribuirlo.

Las dos primeras licencias incluyen una licencia de todos los derechos bajo la propiedad intelectual y además una licencia de patente.

### **3.2. Las licencias de documentación libre**

Las licencias libres se aplican en su gran mayoría, pero no solamente, al software. Se han creado una serie de licencias libres para documentación, sobre todo, porque el software va acompañado de una documentación técnica a menudo necesaria para su uso. No tendría sentido distribuir el software libre sin distribuir la documentación correspondiente bajo términos similares. Por ello, la FSF creó la General Free Document License para acompañar sus programas.

Además, siguiendo la tendencia a la apertura del conocimiento, se han creado otras licencias sobre documentación y materiales, sobre todo, académicos. Presentaremos un ejemplo: la iniciativa Creative Commons.

#### **3.2.1. La Licencia de Documentación Libre de GNU (GFDL)**

La GFDL se destina a la documentación técnica, los manuales de usuario y otros textos relevantes para el software libre. Se modela sobre la GPL, pero cambia sus condiciones para adecuarse a un texto escrito en lugar de al software. La licencia busca el equilibrio entre permitir las modificaciones (sobre todo, aquellas que son necesarias para documentar una modificación del software), mantener la autoría de la obra inicial y respetar las ideas y las opiniones de los autores originales.

#### **Componentes esenciales de la GFDL**

La licencia define varios elementos de un documento para establecer los derechos y las obligaciones correspondientes a cada uno de esos, por ejemplo "secciones secundarias" (avisos legales, dedicatorias, reconocimientos, etc.) y "secciones invariables" (secciones secundarias que no se podrán modificar).

La GFDL otorga varios derechos relativos a la copia, la distribución, la modificación, la agregación y combinación, la colección y la traducción del documento original, que están, en su mayoría, permitidas bajo condiciones de respeto de la autoría original, el mantenimiento de las partes invariables y la provisión de acceso a una versión transparente del documento (una copia legible y modificable por un tercero, por programas no propietarios o genéricos, como ASCII, XML con DTD público, HTML etc., similar al código fuente de un programa).

La modificación implica una serie de obligaciones: cualquier obra derivada debe cambiar de título en la portada, detallar a los autores originales y las modificaciones, indicar dónde se puede encontrar la versión original y mantener los avisos de *copyright* y la licencia. Asimismo, se deben mantener las secciones invariables y el tono y el contenido general de las secciones secundarias. Hay que eliminar de las obras derivadas cualquier indicación de patrocinio (*endorsements*).

### Otros aspectos relevantes y comentarios

Como la GPL, la GFDL mantiene el *copyleft* de los documentos: hay que distribuir cualquier modificación bajo la misma licencia y no se puede combinar con texto que provenga de una obra bajo cualquier licencia más restrictiva.

La licencia no solamente se aplica a documentación técnica para software. También se puede usar sobre cualquier texto, específicamente cualquier obra "literaria" que se desarrolle a la manera del software libre: en obras en colaboración. De hecho, Wikipedia (en [www.wikipedia.org](http://www.wikipedia.org)) se publica bajo la GFDL.

La GFDL no es la única licencia de documentación libre. En parte debido a la polémica sobre ésta, muchos proyectos de software libre crearon sus propias licencias: la FreeBSD Documentation License, la Apple Common Documentation License o la Open Publication License, y la OR Magazine License (de O'Reilly).

### 3.2.2. La iniciativa Creative Commons

La iniciativa Creative Commons (cuyas posibles traducciones en castellano serían 'espacio público creativo' o 'comunidad creativa'), abreviada como CC, es un proyecto de la Universidad de Stanford, California, creado por una serie de expertos en derechos de autor, entre otros Lawrence Lessig. Intenta ayudar a los autores y los creadores a distribuir libremente sus obras para uso del público, ampliando por lo tanto el número de obras creativas accesibles a todos. Se dirige, sobre todo, a las creaciones literarias y artísticas y no al software, y recomienda expresamente la GFDL para cualquier documentación informática. Además, la CC propone un sistema privado, bajo derecho americano, para limitar la duración de la protección de *copyright* a catorce años en vez del plazo acordado por ley (generalmente, la vida del autor más setenta años) sobre la base de una declaración pública. Finalmente, permite dedicar obras al dominio público, también bajo las condiciones del derecho de autor de los Estados Unidos.

#### Web recomendada

La Creative Commons se encuentra en <http://creativecommons.org/>.



### ***Some rights reserved***

La iniciativa Creative Commons actúa bajo un lema que es un juego de palabras sobre la reserva habitual de derechos de autor *all rights reserved*. El lema es "Some rights reserved" ('Algunos derechos reservados'), similar al de la FSE, que es "All rights reversed" ('Todos los derechos invertidos'). La licencia más libre de CC permitiría incluso usar la expresión "No rights reserved" ('Ningún derecho reservado').

Además de una versión genérica de la licencia, que intenta enmarcarse dentro de los convenios internacionales sobre derechos de autor, hay versiones adaptadas a los marcos legales de cada país: España, Perú, Inglaterra, Japón, etc. (y versiones lingüísticas, por ejemplo en catalán). La última versión genérica, la 3.0, tiene un pacto de compatibilidad para permitir la equivalencia de licencias entre estas versiones "locales".

La estrategia de la CC ha sido crear una serie de licencias modulares que establecen qué derechos se conceden a los licenciarios. Los autores pueden elegir los derechos que se reservan y se otorgan en la licencia en función de tres criterios: usos comerciales, obras derivadas y reciprocidad (*copyleft*).

### **Licencia CC**

Como consecuencia, una licencia CC se crea a partir de dos preguntas:

#### **1) Permitir usos comerciales o no**

- ***Commercial (comercial)***. Se permite cualquier tipo de uso, incluso el comercial.
- ***Non commercial (no comercial)***. Se permite cualquier acto de explotación y la derivación, siempre que sea para fines no comerciales.

#### **2) Permitir la creación de obras derivadas o no**

- ***No derivative works (ninguna obra derivada)***. No se permite la modificación para crear obras derivadas.
- ***Share alike (compartir de manera igual)***. Se permite la redistribución de la obra y de obras derivadas solamente bajo términos iguales a la licencia original (*copyleft*).

Además, las licencias contienen un núcleo de términos comunes a todas las variantes:

- Se obliga a mantener los avisos de autoría y *copyright*;
- Se permite establecer vínculos de Internet en las obras publicadas en ese medio;
- No se permite modificar la licencia;
- No se permite usar medios tecnológicos para restringir los usos legítimos de la obra (es decir, tecnologías de DRM);
- Se aplica a todos los países del mundo;

- Es irrevocable y dura el plazo de la protección de *copyright*;
- Ofrece una garantía de titularidad y de no violación de derechos de terceros (para aumentar la confianza en la reutilización y la redistribución de la obra);
- Se permite al autor o al titular de derechos distribuir la obra bajo una licencia diferente;
- Contiene una excepción especial que permite compartir ficheros (P2P *file-sharing*), lo cual no es considerado como una actividad comercial, siempre que no tenga fines de lucro.

### Ejemplos

Algunos ejemplos de las licencias potenciales incluyen:

- La licencia Attribution-NonCommercial-ShareAlike permite la modificación, obliga a mantener la misma licencia en obras derivadas y prohíbe los usos comerciales. La licencia MIT OpenCourseWare es de este tipo (en <http://ocw.mit.edu/OcwWeb/Global/terms-of-use.htm>). Otro texto con esta licencia es "HOWTO: Installing Web Services with [free software]", en <http://www.linuxjava.net/howto/webapp/>.
- La licencia Attribution-Noncommercial obliga a dar crédito y restringe los usos comerciales. La Electronic Freedom Foundation, en [www.eff.org](http://www.eff.org), usa esta licencia.
- La licencia más libre (sin ser de dominio público) es la Attribution, que obliga a dar crédito y permite todo lo demás.

El sitio [www.creativecommons.org](http://www.creativecommons.org) contiene una herramienta automatizada para crear la licencia a partir de las respuestas a preguntas sobre dichos criterios. La herramienta crea y ofrece al usuario el texto de la licencia. Las licencias vienen en tres formatos:

- una versión fácil de leer: un resumen muy fácil de entender ("Commons deed" o "Human code"), con iconos que mencionamos más abajo;
- una versión legal para abogados: la versión completa de la licencia ("Legal code");
- una versión legible por ordenadores: una expresión en metadatos RDF y XML para que un proceso informático automatizado pueda entender la licencia en el contexto de la web semántica ("Digital code").



Atribución



Usos no comerciales



Sin modificar



Compartir / *copyleft*

### 3.3. Licencias de tipo *freeware* y *shareware*

Únicamente queremos comentar aquí que las licencias de tipo *shareware* y *freeware* no son licencias libres. Aunque los programas correspondientes puedan distribuirse gratuitamente, no proporcionan acceso al código fuente y, en su gran mayoría, no respetan las condiciones mínimas para ser libres o abiertas: las cuatro libertades de la FSF o las directrices de la OSD. Por lo tanto, no las incluimos en este estudio.

#### Lectura recomendada

Para un comentario breve sobre dichas licencias, podéis ver: L. P. Deutsch (1997). *Tipos de licencias para software redistribuible libremente*.

## 4. Conclusiones

Actualmente, hay "muchas" licencias libres (unas setenta aprobadas por la OSI como "de fuentes abiertas")... hasta el punto que la OSI ha establecido un comité contra la proliferación de licencias, algo que comentaremos en el módulo 7. En su sitio web, la OSI ha clasificado (arbitrariamente, según algunos) las licencias disponibles en "más populares", "de uso frecuente", y "menores". Observamos con interés en Sourceforge, el mayor repositorio de software libre, la evolución del uso de las diferentes licencias, con la GPLv2 que sigue a la cabeza con un 70% de los proyectos (lo que no significa necesariamente el 70% del código).

Por otro lado, el software y las licencias no se hallan en un entorno estable e invariable, sino que viven en un medio cambiante que evoluciona rápidamente, tanto técnica como legalmente. Por ejemplo, los enlaces dinámicos y los métodos de Java no existían cuando se redactó la licencia GPL v.2.0, ni existía la legislación sobre la protección de la información de identificación de derechos y de las medidas de protección tecnológica de los mismos. En consecuencia, para mantener la libertad del código, se requiere una adaptación continua de la tecnología al marco legal y viceversa, es decir, adaptar las licencias a los nuevos desarrollos tecnológicos (SaaS, ASP, *web-services*) y legales (DRM). Hemos visto que se ha redactado la nueva GPLv3 teniendo en cuenta esta evolución.

Sin embargo, se argumenta que las licencias no son suficientes para mantener la libertad del código en un mundo en rápida evolución. La FSF propone otro modelo interesante, que agrega una estructura y unos procesos globales para proteger el software libre. Predican la centralización de los derechos de autor en un solo titular (por ejemplo, la FSF) que pueda gestionar estas licencias y su evolución frente al derecho y la tecnología: un fiduciario a escala internacional que reaccione ante los cambios legales y tecnológicos y que tome una postura activa en defensa de las libertades. Es cierto que hasta hoy la FSF ha sido un modelo muy eficaz en la protección de código bajo la GPL contra el abuso y la privatización.

Finalmente, queremos resaltar la importancia que tienen las licencias, su selección y su respeto en relación con el uso, la distribución y la comercialización de software libre o de productos basados en software libre:

- Para los usuarios, es fundamental el estudio de las licencias que se aplican a los programas –y sus diferencias y compatibilidades, en entornos de desarrollo y ejecución cada vez más complejos, como los entornos distribuidos (*web-services*, ASP en línea, etc.) o la programación por componentes.

### Web recomendada

Para más información, podéis consultar <http://fsfeuroWpe.org/projects/fla/>.

- Asimismo, para los desarrolladores de software es también fundamental realizar la gestión de la propiedad intelectual de los contribuidores al código y de los usuarios intermedios y finales. Es importante decidir de antemano la licencia que se va a usar y conocer los aspectos relevantes –ventajas, inconvenientes, consecuencias– de cada licencia... ¡Sobre todo antes de pensar en crear una nueva!

