

## Licencias incompatibles de *software* libre

### *Incompatible Free Software Licenses*

NAYARETH GUERRA GACITÚA

*Egresada de Derecho, Chile*

**RESUMEN** Este artículo indaga acerca de las licencias de uso libres incompatibles. El objetivo principal es presentar un estudio respecto de la incompatibilidad de ciertas licencias libres con la GNU General Public License (GPL) tomando como referente las razones que esgrime la Free Software Foundation al momento de justificar la incompatibilidad. La incompatibilidad de licencias es relevante cuando un desarrollador de *software* decide utilizar código GPL en conjunto con código bajo una licencia incompatible: el desarrollador estará infringiendo los términos de uso de una u otra licencia, lo que, finalmente, implicaría un incumplimiento contractual y responsabilidad. De ahí surge la importancia de comprender qué condiciones de la GPL entran en conflicto con una licencia incompatible. La primera sección de este artículo introduce el concepto de *software* y su regulación. La segunda sección revisa la legislación nacional aplicable. La tercera sección delinea qué se conoce como *software* libre, así como también un concepto con el cual es usualmente confundido: *software* de código abierto. La cuarta sección se centra en las licencias de uso de *software* libre, para luego presentar la licencia libre por excelencia: la GNU General Public License. Finalmente, el artículo presenta las principales licencias consideradas por la FSF como incompatibles.

**PALABRAS CLAVE** Programa computacional, *software* libre, código abierto, licencias libres.

**ABSTRACT** This article analyses incompatibility of free licensing. Therefore, the main purpose is presenting a study on incompatibility of free licensing with the GNU General Public License (GPL), by having in consideration reasons argued by the Free Software Foundation when it justifies incompatibility. Licensing incompatibility is relevant when software developer decides to use code under GPL with code under an incompatible license: the developer will infringe the terms of use from one or another license and, eventually, will imply a breach of the contract and liability. That explains the importance of understanding the terms of the GPL and its potential incompatibility with other licenses. The first section of this article introduces the concept of software and its regulation. Then, the second section analyses applicable domestic law. The third section explains what is known as free software and another concept with which the latter is, usually, confused: open source software. The fourth section focuses in licensing of free software and, after that, introduces the ultimate free license: the GNU General Public License. Finally, this article introduces the main licenses that the FSF considers incompatible.

**KEYWORDS** Computer program, free software, open source software, open licensing.

## INTRODUCCIÓN

Previo al estudio de las licencias de uso de *software*, es necesario analizar qué se entiende por *software* y de qué forma es amparado por el derecho.

En nuestro país es la Ley sobre Propiedad Intelectual (en adelante, «LPI») la que otorga un concepto de programa computacional o *software*, señalando que es un «conjunto de instrucciones para ser usadas directa o indirectamente en un computador, a fin de efectuar u obtener un determinado proceso o resultado, contenidas en un cassette, diskette, cinta magnética u otro soporte material».<sup>1</sup> Además, la LPI comple-

---

1. Ley 17.336 sobre Propiedad Intelectual, art. 5.º, letra t).

menta la anterior definición señalando que la protección del derecho de autor abarca «los programas computacionales, cualquier sea el modo o forma de expresión, como programa fuente o programa objeto, e incluso la documentación preparatoria, su descripción técnica y manuales de uso».<sup>2</sup>

Observamos de inmediato que es el derecho de autor la rama del derecho llamada a proteger al *software*; sin embargo, la propiedad industrial (a través de las patentes<sup>3</sup> y marcas<sup>4</sup>) también podría hacerlo, con no poca polémica. Sin entrar en detalle, cabe señalar que este debate fue zanjado concluyéndose que «el programa de ordenador constituye una obra del intelecto resultante de un proceso creativo similar al que demanda una obra literaria [...] que, cuando es original en su composición y en su expresión, está protegida por el derecho de autor» (Lipszyc, 1993: 107). Además en el *software*, al igual que en una obra literaria, «se escriben caracteres sobre un soporte, se emplea un lenguaje y su resultado puede ser reproducido e impreso» (Lipszyc, 1993: 107); por ende, el programa

---

2. Ley 17.336, art. 3.º núm. 16.

3. Si consideramos que «el programa en su última etapa, código máquina, se convierte en una combinación de señales electromagnéticas que actúan físicamente sobre el ordenador, produciendo un resultado» (Fernández, 1996: 13), resultado que es nuevo y se obtiene por un medio físico, es posible concluir que el *software* podría ser protegido mediante el sistema de patentes. Sin embargo, la idea de amparar al *software* por medio de patentes ha sido «rechazada con carácter general» (Aparicio, 2004: 46), por considerarse al *software* «como el resultado de una actividad intelectual sin aplicación industrial directa, al tiempo que era (y es) requisito esencial de las patentes en todos los ordenamientos la novedad entendida como altura inventiva, requerimiento que no cumplen todos los programas de ordenador que, sin embargo, sí resultan ser ‘originales’ a los efectos del derecho de autor» (Aparicio, 2004: 46). En este sentido, el profesor Hardings (2003) identifica varias razones por las cuales sería perjudicial someter a los programas al régimen de las patentes.

4. En principio, la protección marcaría es eficaz para resguardar al *software* en el sentido de amparar el nombre o signo distintivo con que se comercializa un programa. Pero se sostiene que «la protección que ofrece la marca al *software* es de orden periférico o indirecto ya que, aunque tiene carácter *erga omnes*, su protección queda reducida a la del nombre con el que se comercializa o a la del signo distintivo que acompaña su comercialización» (Hernando, 1995: 118). En resumen, y siguiendo a Aparicio (2004: 46), la tutela que otorgan las marcas al *software* «es de carácter mínimo, y no persigue tanto la protección del programa en sí como la de sus elementos identificativos o la posición de la empresa perjudicada en el mercado».

computacional se encuentra sometido al derecho autoral como una obra literaria.<sup>5</sup>

La vía de protección a través del derecho de autor constituye hoy la posición dominante. Como dice D. Karjala, existe unánime acuerdo en la justificación de proteger legalmente la copia literal de los programas, y también hay acuerdo casi universal en que una ley como las de derecho de autor es el camino más apropiado para llevar a cabo tal protección (Fernández, 1996: 26).

Así, la protección que el régimen autoral (en general) otorga sobre los programas conlleva una serie de beneficios, entre los cuales encontramos: un mayor plazo de protección,<sup>6</sup> el nacimiento automático de la protección<sup>7</sup> y pocas obligaciones para el titular.<sup>8</sup>

---

5. En relación a los argumentos a favor y en contra de la protección de derecho de autor, véase Lipszyc (1993: 104 y ss). En el plano internacional se reafirma la consideración del *software* como obra literaria en varios tratados. Entre éstos destaca el Acuerdo sobre los Aspectos de los Derechos de Propiedad Intelectual Relacionados con el Comercio (Acuerdo ADPIC), que establece que «los programas de ordenador, sean programas fuente o programas objeto, serán protegidos como obras literarias en virtud del Convenio de Berna (1971)». A pesar de la referencia al Convenio de Berna, éste no menciona expresamente a los programas de computador como obra literaria; sin embargo la OMPI, al revisarlo, ha expresado que «en general se considera que los abarca [al *software*] ya que el Convenio de Berna establece que se aplica a ‘todas las producciones en el campo literario, científico y artístico, cualquiera que sea el modo o forma de expresión’ (artículo 2.1) y que un programa de ordenador es una producción del campo literario» (OMPI, 1996: 17). Otro convenio internacional que reafirma que el *software* es una obra literaria es el Tratado de la OMPI sobre Derecho de Autor, que complementa la Convención de Berna: establece que «los programas de ordenador están protegidos como obras literarias en el marco de los dispuesto en el artículo 2 del Convenio de Berna. Dicha protección se aplica a los programas de ordenador, cualquier que sea su forma o modo de expresión».

6. Este plazo «establecido en las distintas legislaciones, es un plazo más largo que el de los derechos de propiedad industrial» (Davara, 2005: 129).

7. Dicha protección «nace desde el mismo instante en que la idea es expresada en un soporte, sin necesidad de someterla a ningún formalismo (al contrario de lo que sucede en el ámbito de la propiedad industrial) para que ya sea objeto de esa protección» (Davara, 2005: 130). Así también Lipszyc, quien señala —a propósito de la protección de los programas de ordenador como obras literarias— que «los programas de ordenador se benefician automáticamente de la misma protección de que gozan las obras literarias» (1993: 47).

8. En correlato con lo anterior, «el titular de los derechos no necesita [...] realizar ninguna acción o cumplir alguna obligación para ser protegido» (Davara, 2005: 130).

## ESTATUTO LEGAL DEL SOFTWARE

Como ya hemos adelantado, el *software* se encuentra amparado jurídicamente por el derecho autoral. Por ende, se encuentra regulado por la LPI, la cual reglamenta específicamente<sup>9</sup> dos aspectos: la titularidad de los derechos sobre el programa desarrollado por encargo y las excepciones y limitaciones al derecho de autor en relación con el *software*. Revisemos, someramente, ambos temas.

En general, se entiende que es autor y titular de una obra la persona que la crea. Esto es, quien «directamente realiza una actividad tendiente a elaborar una obra intelectual, una creación completa e independiente, que revela una personalidad, pues pone en ella su talento artístico y su esfuerzo creador» (Satanowsky, 1954: 265). En este sentido, Lipszyc estima que, dado que la obra nace del intelecto, «ésta sólo puede ser realizada por las personas físicas, la consecuencia natural es que la titularidad originaria corresponda a la persona física que crea la obra» (1993: 123). Existe, sin embargo, un caso en que el titular originario del derecho de autor es directamente una persona jurídica: «tratándose de programas computacionales, serán titulares del derecho de autor respectivo las personas naturales o jurídicas cuyos dependientes, en el desempeño de sus funciones laborales, los hubiesen producido, salvo estipulación escrita en contrario» (LPI, artículo 8, inciso segundo). Es decir, por regla general, cuando un *software* es desarrollado por un trabajador (dentro de su función laboral) el titular de dicha obra será su empleador, a menos que exista una estipulación escrita en que se establezca que los derechos recaerán en el dependiente.

Aun más, mediante una reciente modificación a la LPI se estableció que «respecto de los programas computacionales producidos por encargo de un tercero, se reputarán cedidos a éste los derechos de su autor, salvo estipulación escrita en contrario» (artículo 8, inciso tercero). Esta modificación implica que al desarrollar un *software* por encargo, los derechos sobre éste le corresponderán al tercero que encomendó la creación del programa, mas no al desarrollador de él.

El segundo aspecto a destacar de la regulación que la LPI hace del

---

9. En lo no regulado específicamente se aplican las reglas relativas a las obras literarias de la LPI al *software*.

*software* es la reciente modificación en relación con las excepciones y limitaciones del derecho de autor. Al respecto encontramos el artículo 71Ñ de la LPI, donde se establece que serán permitidas tres actividades sobre el *software* sin necesidad de autorización del titular del derecho de autor. Revisemos dichas actividades.

En primer lugar, se permite «la adaptación o copia de un programa computacional efectuada por su tenedor», pero para permitir lo anterior se exige que «la adaptación o copia sea esencial para su uso, o para fines de archivo o respaldo y no se utilice para otros fines» (artículo 71Ñ, letra a). En definitiva, se permite copiar (reproducir<sup>10</sup>) o adaptar<sup>11</sup> (modificar) un *software* pero sólo si es esencial para su uso o respaldo y que no sea empleado para otros fines.

Además, señala la LPI, «las adaptaciones obtenidas en la forma señalada no podrán ser transferidas bajo ningún título, sin que medie autorización previa del titular del derecho de autor respectivo» (artículo 71Ñ, letra a). Lo anterior significa que el *software* modificado sólo podrá ser transferido mediante el otorgamiento de una licencia por parte del titular del *software* original. Asimismo «las copias obtenidas [...] no podrán ser transferidas bajo ningún título, salvo que lo sean conjuntamente con el programa computacional que les sirvió de matriz» (artículo 71, letra a). Entendemos que esta exigencia implica que el *software* modificado sólo podrá ser transferido en conjunto con el programa original.

En segundo lugar, la reciente modificación a la LPI incorporó como excepción al derecho de autor operaciones de ingeniería inversa<sup>12</sup> sobre una copia legal de *software* pero «siempre que se realicen con el único

---

10. La reproducción es «la fijación permanente o temporal de la obra en un medio que permita su comunicación o la obtención de copias de toda o parte de ella, por cualquier medio o procedimiento» (LPI, artículo 5, letra u). A su vez, fijación es la incorporación de sonidos, o la representación de éstos, a partir de los cuales puedan percibirse, reproducirse o comunicarse mediante un dispositivo» (LPI, artículo 5, letra x).

11. Consiste en «utilizarla [la obra] en cualquier forma que entrañe una variación, adaptación o transformación de la obra originaria» (LPI, artículo 18, letra c).

12. En pocas palabras, la ingeniería inversa «se basa en tomar aplicaciones y procesos que ya existen y se encuentren en producción para, a partir de ellos, extraer los datos, modelos de datos y relaciones entre los mismos» (Ramírez, 2005: 38). En relación con este tema, véase Baro (2007).

propósito de lograr la compatibilidad operativa entre programas computacionales o para fines de investigación y desarrollo» (artículo 71Ñ, letra b).

En tercer lugar, la reforma del año 2010 admitió la realización de actividades sobre una copia legal de *software* «con el único propósito de probar, investigar o corregir su funcionamiento o la seguridad del mismo u otros programas, de la red o del computador sobre el que se aplica» (artículo 71Ñ, letra c), siendo tales actividades lícitas pero prohibiéndose la utilización de la información obtenida para otros fines.

Finalmente, y mediante la mencionada reforma, la LPI contempló como una excepción al derecho de autor la «reproducción provisional de una obra [...] [que] deberá ser transitoria o accesoria; formar parte integrante y esencial de un proceso tecnológico, y tener como única finalidad la transmisión lícita en una red entre terceros por parte de un intermediario, o el uso lícito de una obra u otra materia protegida» (artículo 71O). La importancia de la incorporación de esta excepción radica en que, desde el momento en que un *software* es descargado o instalado, está siendo ejecutado. Dichos actos (en principio) parecen estar restringidos por cuanto la reproducción de un programa es un derecho patrimonial exclusivo del titular del derecho de autor, como ya vimos. Sin embargo, gracias a la presente excepción, se permiten los anteriores actos indispensables, ya que se trata de una reproducción provisional y necesaria para ejecutar (o usar) un programa lícitamente adquirido, amparado por la LPI.

## SOFTWARE LIBRE Y SOFTWARE DE CÓDIGO ABIERTO

En los albores de la informática el *software* se distribuía junto con el *hardware*. En este escenario existía el principio tácito comunitario de que el *software* «no se compraba ni se vendía, simplemente se entregaba» (Hardings y Fuentes, 2003). Este panorama de libertad cambió radicalmente cuando la empresa IBM «anunció que a comienzos de 1970 comenzaría a vender parte de su *software* por separado» (González, Seoane y Robles, 2004), debido al proceso por monopolio en su contra. En ese momento se abandonó el principio tácito de compartir el código fuente, situación que «inauguró el período del llamado, dentro del mundo de la informática, ‘modelo del *software* propietario’» (Carranza,

2004: 73).<sup>13</sup> Es así como la pérdida de la capacidad de estudiar, mejorar y compartir el código fuente, y la repentina necesidad de efectuar pagos por el uso del *software*, comenzó a ser rechazada por algunos desarrolladores, situación que provoca el nacimiento del proyecto GNU,<sup>14</sup> la Free Software Foundation (en adelante, «FSF») y el *software* libre.

Stallman define *software* libre como «cualquier programa cuyos usuarios gocen de ciertas libertades fundamentales» (2004: 45); pero más allá de un concepto en específico, lo que caracteriza al *software* libre son las cuatro libertades básicas que se le conceden al usuario:

- Libertad 0. Libertad de ejecutar el programa para cualquier propósito.
- Libertad 1. Libertad de estudiar cómo trabaja el programa y cambiarlo para que el usuario haga lo que quiera. Esta libertad permite que los usuarios puedan «modificar el programa según sus necesidades o conveniencias y utilizar esas modificaciones de manera privada ‘sin siquiera tener que anunciar que dichas modificaciones existen’». Para lograr lo anterior «el acceso al código fuente es una condición necesaria».<sup>15</sup>

---

13. *Software* propietario es, según la Free Software Foundation, «otro nombre para el *software* no libre», es decir, aquel *software* cuyo «uso, redistribución o modificación está prohibida, o requiere que usted solicite autorización o está tan restringida que no pueda hacerla libre de un modo efectivo». Dicho concepto no se debe confundir con *software* privativo, ya que éste último es aquel «*software* desarrollado por un usuario (generalmente una organización o compañía). Dicho usuario lo tiene en su poder y lo utiliza, y no lo libera al público ni como código fuente ni como binario». Sin embargo, «un programa privativo es *software* libre en un sentido trivial si su único usuario tiene plenos derechos sobre él» (véase «Categories of free and non software», disponible en <<http://www.gnu.org/philosophy/categories.en.html>>). En relación con las diversas categorías de *software*, véase Carranza (2004: 73 y ss).

14. Podemos señalar dos acepciones de GNU. La primera se refiere a las siglas de la idea detrás del movimiento «GNU is Not Unix». La otra corresponde a la similitud con el vocablo inglés *gnu* (ñu, en español), un animal herbívoro africano. El proyecto GNU tenía por objetivo crear un sistema operativo compatible con Unix. En 1983 Richard Stallman anunció la creación del GNU, siendo un elemento esencial de dicho proyecto el que se llevaría a cabo bajo «el espíritu cooperativo que prevalecía en la comunidad computacional en sus primeros días» (FSF, «Visión general del sistema GNU», disponible en <<http://www.gnu.org/gnu/gnu-history.es.html>>). A pesar de la intención inicial, el proyecto comenzó a abarcar no sólo un sistema base, sino también diversos programas de aplicación.

15. Entendemos esta fuerte exigencia de otorgar acceso al código fuente debido a que se

- Libertad 2. Libertad para redistribuir copias para que pueda ayudar al prójimo. Esta libertad «debe incluir las formas binarias o ejecutables del programa, así como el código fuente; tanto para las versiones modificadas como para las que no lo están».<sup>16</sup>
- Libertad 3. Libertad de distribuir copias de sus versiones modificadas a terceros.

Así las cosas, para 1994 surgieron voces contrarias a la ideología del *software* libre, lo que en definitiva significó la generación de una nueva forma de desarrollo de *software* denominada ‘código abierto’ u ‘open source’, la cual se aglutinó en la Open Source Initiative (en adelante, «OSI»).

Al igual que el *software* libre, no existe una definición única de código abierto; más bien se trata de «una especificación de qué es permitido en una licencia de *software* para que ese *software* sea calificado como de código abierto» (Perens, 1999). Es decir, la OSI considerará a un *software* como de código abierto si cumple con ciertos criterios (a semejanza de las libertades exigidas por la FSF).

Los criterios o requisitos que componen la definición de «código abierto» son los siguientes:

1. Redistribución libre. Señala la OSI que «la licencia no deberá impedir la venta o el ofrecimiento del *software* como un componente de una distribución de *software* que contiene programas de diferentes fuentes. La licencia no debe exigir el pago de una regalía o tarifa por dicha venta».<sup>17</sup>

---

requiere de dicho código para realizar alguna modificación al *software*; si sólo se otorgara el código objeto (código ya compilado) habría que pasar por un complejo proceso para revertirlo a código fuente (en relación con este tema, véase Hardings y Fuentes, 2004: 2). Lo mismo sucede con la distribución del *software*: si sólo se entregara el código objeto de un programa, y siendo dificultoso volverlo a código fuente, se generarían obstáculos para modificar dicho programa. Lo anterior redundaría en una forma encubierta de incumplir con las solicitadas libertades y, eventualmente, con el incumplimiento de las condiciones exigidas en las licencias que gobiernan dichos programas, situación que analizaremos más adelante. En correlato con lo anterior, la FSF establece expresamente que el código ofuscado «no es código fuente real, y no cuenta como código fuente» («La definición de *software* libre», disponible en <<http://www.gnu.org/philosophy/free-sw.es.html>>).

16. FSF, «La definición de *software* libre».

17. Open Source Initiative, «The Open Source Definition», disponible en <<http://www>.

2. Código fuente. La OSI exige que «el programa debe incluir el código fuente, y debe permitir la distribución en código fuente como en forma compilada».
3. Obras derivadas. La definición de código abierto requiere que las licencias permitan «modificaciones y obras derivadas, y debe permitir que ellas sean distribuidas bajo los mismos términos de la licencia del *software* original».
4. Integridad del código fuente del autor. La OSI acepta que la licencia impida que el código fuente sea distribuido en forma modificada pero «sólo si la licencia permite la distribución de ‘archivos parches’<sup>18</sup> con el código fuente con el objetivo de modificar el programa en el tiempo de construcción».
5. No discriminación contra personas o grupos de personas. Esto porque «con el fin de obtener el máximo beneficio [...], es necesario que la máxima diversidad de personas y grupos tengan igualmente derecho a aportar al código abierto».
6. No discriminación contra campos laborales. «La licencia no puede restringir a nadie que haga uso del programa en un campo laboral específico. Por ejemplo, no se puede impedir que el programa sea utilizado en un negocio, o que se utilice para la investigación genética».
7. Distribución de la licencia. La OSI exige que «los derechos vinculados al programa deben aplicarse a todos aquellos a quienes se redistribuya el programa, sin necesidad de pedir una licencia adicional».
8. La licencia no debe ser específica a un producto. Este criterio implica que «los derechos vinculados al programa no deben depender de que el programa forme parte de una distribución de *software* particular».
9. No restricción de la licencia en relación a otro *software*. Esto significa que una licencia de código abierto «no debe imponer restric-

---

opensource.org/docs/osd>. Todas las citas que siguen y definen «código abierto» provienen de este documento.

18. Un *parche* o *archivo parche* es un archivo que actualiza el archivo original.

ciones sobre otro *software* que es distribuido junto con el *software* licenciado».

10. Neutralidad tecnológica de la licencia. Lo que la OSI exige es que «ninguna disposición de la licencia puede basarse en una tecnología determinada o estilo de interfaz».

Si una licencia cumple con los anteriores requisitos, la OSI le otorga una especie de certificación de marca ('OSI Certified') que «implica que el *software* está siendo distribuido bajo una licencia que ha sido aprobada como conforme a la Open Source Definition» (McJohn, 2000, citado por Carraza, 2004: 94).

### LICENCIAS DE USO DE SOFTWARE LIBRE

Uno de los aspectos de los programas computacionales que el derecho informático abarca es el ámbito contractual. En general, se ha entendido doctrinariamente<sup>19</sup> que los contratos informáticos son todas «aquellas operaciones jurídicas por las cuales se crean, modifican o extinguen relaciones obligacionales sobre bienes o servicios informáticos» (Carrascosa, Pozo y Rodríguez de Castro, 2000: 116). Pues bien, dentro de esta categoría encontramos los contratos relativos al *software*, uno de los cuales es la licencia de uso, materia central de este artículo.

El término licencia proviene del latín *licentia*, que significaba permiso o autorización para hacer algo, y actualmente se utiliza dicha palabra, dentro de la cultura jurídica continental, «por recepción del vocablo inglés (*licence* o, puesto que proviene del derecho estadounidense, *license*)» (Aparicio, 2004: 72). Respecto a la locución 'de uso', ésta tiene por finalidad «enfaticar el carácter del contrato (proveer un programa para su utilización) frente a otros que son calificables 'de explotación'» (Aparicio, 2004: 75), como la licencia de distribución.<sup>20</sup> A pesar de lo

19. En dicho sentido, para Davara la contratación informática es «aquella cuyo objeto sea un bien o servicio informático —o ambos— o que una de las prestaciones de las partes tenga por objeto ese bien o servicio informático» (2005: 255).

20. Se trata de un «contrato por el que el titular de los derechos de explotación [el titular de los derechos patrimoniales, en nuestro derecho] cede los derechos de distribución del *software*, con el alcance y las limitaciones que el acuerdo contractual determine» (Carrascosa, Pozo y Rodríguez de Castro, 2000: 166).

anterior, utilizar el término ‘licencia’, desde un punto de vista estrictamente jurídico y a la luz de la LPI, parece impropio porque no existe la figura de licencia de uso propiamente tal; entonces ¿a qué figura jurídica corresponde el fenómeno conocido como ‘licencia’ en el derecho comparado? En los siguientes párrafos intentaremos dar una respuesta.

Primero, debemos recordar que el derecho de autor confiere derechos de orden patrimonial (LPI, artículo 17) que se traducen en tres facultades: utilizar directa y personalmente la obra, transferir (total o parcialmente) sus derechos patrimoniales sobre la obra y autorizar la utilización de la obra por parte de terceros. Pues bien, esta última facultad corresponde —jurídicamente— al contrato de licencia de uso de *software*.

En definitiva, licencia de uso es, según la LPI, «el permiso otorgado por el titular del derecho de autor, en cualquier forma contractual, para utilizar la obra en alguno de los modos [...] que esta ley establece» (artículo 20, inciso primero). Es decir, mediante una licencia de *software* el titular del derecho de autor permite a un tercero realizar algunas de las actividades del artículo 18 de la LPI: adaptar (modificar), reproducir (copiar) y distribuir (sea mediante comunicación pública o distribución al público) el programa.

A la luz de este concepto podemos advertir que en el contrato de licencia existen dos partes: el licenciante, titular de los derechos de autor y que autoriza la utilización del *software*, y el licenciatarario, adquirente o usuario, quien recibe dicha autorización por la cual —por regla general— paga algún tipo de retribución.

Estas licencias de uso se pueden clasificar desde distintos puntos de vista; sin embargo, para este artículo interesa la clasificación entre licencias propietarias y las licencias F/OSS.<sup>21</sup> Las licencias propietarias son aquellas en que el usuario «necesita la autorización del titular, de forma que si expresamente no se concede, se consideran prohibidas» (Aparicio, 2004: 86). En oposición a ellas encontramos las licencias F/OSS, las cuales «conceden al usuario amplias facultades de copia, disposición [...] sobre el ejemplar adquirido» (Aparicio, 2004: 87). Observamos,

---

21. Hemos de utilizar el término Free/Open Source Software (F/OSS) para abarcar tanto el *software* libre como los programas de código abierto u *open source*, a pesar de que, como ya hemos revisado, existen diferencias entre ambos conceptos. Hemos tomado la expresión F/OSS de Carranza (2004: 24).

de inmediato, la diferencia esencial entre ambas categorías: las licencias propietarias establecen claramente lo que los usuarios pueden realizar respecto del programa, así como bajo qué condiciones, mientras que las licencias F/OSS conceden una serie de derechos (o libertades) al usuario, siempre y cuando éste cumpla con las condiciones establecidas en la licencia.

Ahora, dentro de la categoría de las licencias F/OSS distinguimos las licencias de *software* libre, por una parte, y las licencias de código abierto, por otra. Una licencia es de *software* libre cuando ésta «permite a los usuarios aprovecharlo libremente en cuanto a su uso, reproducción, modificación y distribución» (Bain, Gallego, Martínez Ribas y Rius, 2004: 8), es decir, aquella licencia que refleja las cuatro libertades básicas del *software* libre. Por otra parte, una licencia de código abierto es aquella que sigue las directrices de la definición de código abierto. En definitiva, lo que determinará si estamos frente a una licencia de *software* libre o una licencia de código abierto es si la licencia cumple o no con los criterios determinados en dichas definiciones.<sup>22</sup>

Analizaremos la licencia libre por excelencia, la GNU General Public License (en adelante, «GNU» «GPL» o «GPL»), que es determinante a la hora de establecer si una licencia de *software* es compatible o incompatible, como veremos más adelante.

Durante la década de 1980, recién comenzado el Proyecto GNU, sus licencias «estaban vinculadas a cada programa (por ejemplo, la licencia GNU Emacs o la licencia GNU GCC). Se extendió la costumbre de copiar la licencia y cambiarle el título, lo cual [...] creaba problemas de compatibilidad mutua» (Vidal, 2008: 8). Ante esta situación, «Stallman depuró esas licencias particulares para crear una licencia pública de propósito general, la General Public License versión 1, a la que podía acogerse cualquier programador para licenciar su programa» (Vidal, 2008: 8). En 1991, la FSF publica la versión 2 de la GPL, versión a partir de la cual la «utilización [de la GPL] ha ido más allá, hasta convertirse en la licencia más utilizada (por ejemplo, más del 70% de los proyectos anunciados en

---

22. A pesar de la distinción que presentamos, debemos advertir que ella no es excluyente, toda vez que es posible encontrar con cierta frecuencia licencias que sean libres y de código abierto a la vez. Entre estas licencias 'propriadamente F/OSS' (en el sentido de cumplir con ambos criterios a la vez) encontramos, entre otras, a la IBM Public License 1.0.

FreshMeat<sup>23</sup> están licenciados bajo la GPL)» (González, Seoane y Robles, 2004: 43). Como observamos, la GPL es la licencia de mayor trascendencia: «en primer lugar, porque la mayoría del F/OSS es licenciado bajo sus términos. Segundo, porque es la licencia oficial<sup>24</sup> del movimiento *free software* [...]. Finalmente, porque los propios promotores del movimiento *open source* recomiendan su empleo» (Carranza, 2004: 137).

A continuación realizaremos un somero esbozo de las cláusulas más importantes de la GNU GPL versión 2, centrándonos específicamente en dicha versión debido a su importancia y amplia utilización.

En la GPL los derechos otorgados reflejan, obviamente, las cuatro libertades fundamentales del *software* libre: en su cláusula primera la GNU GPL permite copiar y distribuir copias del código fuente original, sin modificaciones; lo que se condice con la libertad 2 del *software* libre. También se permite modificar el programa o parte de él; es decir, la libertad 1 se encuentra presente en la GNU GPL. Además, la licencia autoriza la distribución de dichas modificaciones, lo que corresponde a la libertad 3. Finalmente, la GNU GPL otorga el derecho a copiar y distribuir el programa, con o sin modificaciones, como código objeto o ejecutable, derecho que corresponde a las libertades 2 y 3 del *software* libre.

Como correlato a la concesión de los anteriores derechos, la GNU GPL impone ciertas condiciones o restricciones. En primer lugar, y respecto de la posibilidad de reproducir y distribuir copias sin modificar del programa, la licencia antedicha presenta una serie de exigencias, siendo la más interesante la obligación de otorgar a los destinatarios del programa una copia de la licencia GPL junto con el *software* distribuido. Lo anterior es, según Gómez Padilla (2011: 70):

Una de las formas concebidas para conservar la característica ‘viral’ de esta licencia, pues asegura que con la copia del programa se copie también su forma de licenciamiento, e incurriéndose en una infracción al derecho de autor en caso de obviarse esta obligación, ya que se distribuiría el programa en términos diferentes a los establecidos por el autor en la licencia.

---

23. Un sitio web popular entre desarrolladores de *software*, que sirve como medio de información de las últimas actualizaciones y versiones de programas —principalmente— libres.

24. De hecho la administración de la licencia GNU GPL se encuentra en manos de la FSF.

En segundo lugar, la GPL establece los requisitos a cumplir para copiar y distribuir copias del programa con modificaciones. La más importante de ellas es la exigencia de que la obra derivada distribuida debe ser licenciada como un todo<sup>25</sup> y sólo bajo la misma GNU GPL, cláusula en donde se vuelve a establecer el *copyleft*<sup>26</sup> o viralidad de la GPL.

En tercer lugar, y para copiar y distribuir el programa (u obras derivadas de él) en formato ejecutable u objeto, la licencia GPL requiere que (además de cumplir con los requisitos anteriores) se entregue el código fuente del programa en alguna de las tres formas alternativas que presenta.

Ya revisada la licencia GPL podemos adentrarnos en la distinción entre licencias compatibles e incompatibles que realiza la FSF.

Dicha institución entiende que una licencia de *software* libre es compatible con la GPL cuando permite «combinar el código publicado bajo otra licencia con el código publicado bajo la GNU GPL dentro de un programa más amplio»;<sup>27</sup> al contrario de las licencias anteriores, las licencias incompatibles no permiten la creación de obras derivadas a par-

---

25. Debemos advertir que la GNU GPL reconoce que, cuando en una obra derivada sea posible identificar partes que no sean derivadas del programa original, la GPL no se aplica a dichas partes cuando sean distribuidas en obras separadas. Sin embargo, cuando dichas partes se distribuyen como un todo (en conjunto con la obra original) estaremos ante una obra derivada cuya distribución —completa— debe ser bajo los términos de la GPL. Ahora, si simplemente se reúne una obra no basada en el programa con el programa licenciado bajo la GPL, la antedicha obra no se encuentra en el ámbito cubierto por la GPL.

26. El *copyleft* es considerado un elemento central dentro del movimiento del *software* libre, ya que «autoriza la ejecución del programa, su copia, modificación y distribución de versiones modificadas, siempre que no se añada ninguna clase de restricción a posteriori» (Stallman, 2004: 22). Ahora, desde un punto de vista estrictamente jurídico, el *copyleft* sería una condición resolutoria, «un suceso (condición) que, de darse, produce determinado efecto en los derechos» (Nonius, 2002), como es la resolución de éstos. En otras palabras, si se infringen los términos del *copyleft* (si no se respetan las libertades que envuelven al código) estamos frente a una contravención del contrato que desencadena su resolución. Así también lo considera Carranza (2004: 132), para quien se trata de una renuncia bajo condición resolutoria: «el *copyleft* implica que el titular de los derechos de autor renuncia, respecto de potenciales usuarios, al monopolio de explotación que las leyes le reconocen, pero esta renuncia está sujeta a la condición resolutoria de que estos potenciales usuarios agreguen restricciones al momento de distribuir el *software*».

27. Free Software Foundation, «Preguntas frecuentes acerca de las licencias de GNU». Disponible en <<http://www.gnu.org/licenses/gpl-faq.es.html>>.

tir de código licenciado bajo la GNU GPL. Es decir, dos licencias son compatibles cuando es posible combinar el código publicado bajo una licencia con el código licenciado bajo otra distinta, sin infringir cualquiera de dichas licencias. Así también lo entienden Bain, Gallego, Martínez Ribas y Rius (2004: 16), para quienes un programa es compatible con otro:

Si se puede mezclar o eventualmente interrelacionar sus códigos para crear una obra derivada compuesta de elementos de cada uno de ellos y distribuir el resultado de su integración sin infringir las licencias de uno y de otro, de manera que se pueda cumplir con las condiciones de ambas licencias a la hora de redistribuir el resultado.

En definitiva, entendemos que dos licencias serán compatibles cuando es posible combinar códigos gobernados por cada licencia sin violar o infringir los términos de la otra. *A contrario sensu*, dos licencias serán incompatibles<sup>28</sup> cuando los términos de ellas sean imposibles de conciliar.

## LICENCIAS INCOMPATIBLES DE SOFTWARE LIBRE

En los párrafos anteriores nos abocamos a presentar las licencias de uso de *software* F/OSS, lo que incluyó la presentación de la compatibilidad/incompatibilidad. Pues bien, en el presente apartado nos centraremos en las principales licencias libres incompatibles con la GNU GPL.

### ORIGINAL BSD LICENSE (OBL)<sup>29</sup>

La OBL, también conocida como ‘BSD Antigua’ o ‘BSD de 4 cláusulas’, es una licencia redactada por la Universidad de California en Berkeley, utilizándose por primera vez en 1980 «para la [distribución de] *Berkeley Software Distribution* (BSD), también conocida como BSD Unix, una

28. Existe otro tipo de incompatibilidad conocida como incompatibilidad de códigos. Dicho problema surge al vincular códigos (situación distinta a la modificación o combinación de códigos) gobernados por licencias distintas.

29. Xfree86.org, «Other Copyrights», disponible en <<http://www.xfree86.org/3.3.6/COPYRIGHT2.html#6>>. Nota: la Original BSD License se encuentra en la sección UCB/LBL.

versión mejorada del original sistema operativo UNIX». <sup>30</sup> Se trata de una licencia de redacción muy breve en comparación con las demás. El principio detrás de la OBL 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 [...] los ‘derechos morales’ de los autores por la simple obligación de mantener los avisos de autoría (*copyright notice*)» (Bain, Gallego, Martínez Ribas y Rius, 2004: 17).

Luego de examinar la anterior licencia, la Free Software Foundation ha concluido que la OBL es «una licencia simple y permisiva de *software* libre, sin *copyleft*». <sup>31</sup> Claramente se trata de una licencia sin *copyleft* ya que no exige que las obras derivadas, creadas a partir de código licenciado bajo la OBL, se licencien bajo las mismas condiciones de dicha licencia.

Continúa la FSF expresando que la OBL posee «un serio defecto: la irritante cláusula de publicidad». <sup>32</sup> Dicha cláusula expresa que: «todo el material de publicidad en el cual se mencionan características o la utilización de este *software* debe mostrar el siguiente reconocimiento: Este producto incluye *software* desarrollado por la Universidad de California, Berkeley y sus contribuyentes».

Al comienzo esta cláusula no causaba ningún problema en particular; sin embargo, «otros desarrolladores que utilizan licencias del estilo BSD no copiaron literalmente la cláusula [...] la cambiaron, y sustituyeron ‘Universidad de California’ con su propia institución o nombre». <sup>33</sup> El resultado de esta práctica fue que, finalmente, «había que incluir atribuciones de autorías extensivas en toda la documentación y en el código fuente, relativas a cada autor que agregaba su nombre en una licencia» (Bain, Gallego, Martínez Ribas y Rius, 2004: 19); entonces «si un sistema de *software* requiriera 75 frases distintas, cada una de las cuales cita

---

30. The Linux Information Project, «BSD License Definition», disponible en <<http://www.linfo.org/bsdlicense.html>>.

31. FSF, «Various Licenses and Comments about Them», disponible en <<http://www.gnu.org/licenses/license-list.en.html#GPLIncompatibleLicenses>>.

32. FSF, «Various Licenses and Comments about Them».

33. FSF, «Various Licenses and Comments about Them».

a un autor o grupo de autores diferente [...] necesitará toda una página para el aviso».<sup>34</sup> Finalmente, el problema fue resuelto cuando Williams Hoskins, director de la Oficina de Licenciamiento de Tecnologías de la Universidad de California, anunció en julio de 1999 que «los licenciarios y distribuidores ya no están obligados a incluir el reconocimiento dentro de los materiales de publicidad».<sup>35</sup> Sin embargo, para la FSF lo anunciado por Hoskins «no elimina la herencia de la cláusula de propaganda: aún existen cláusulas similares en las licencias de muchos paquetes que no son parte de BSD», es decir, «el cambio en la licencia BSD no tiene efecto en los otros paquetes que imitaron la vieja licencia BSD; solamente los desarrolladores que los escribieron pueden cambiarlos».<sup>36</sup>

En definitiva, para la FSF «el defecto no es fatal, es decir, no vuelve al *software* como no libre. Pero sí causa problemas prácticos, incluyendo la incompatibilidad con la GNU GPL».<sup>37</sup> Esta incompatibilidad se produce específicamente con la sección sexta de la GNU GPL v2, la cual expresa: «no puede imponer al receptor [del programa] ninguna restricción más sobre el ejercicio de los derechos aquí garantizados». Es decir, desde nuestro punto de vista, la exigencia de incluir ciertos avisos en los materiales publicitarios que mencionen características del programa licenciado, consiste en una restricción adicional impuesta a los receptores; por ende, infringe la sección ya presentada, conllevando a su vez la incompatibilidad entre ambas licencias.

Sin perjuicio del elemento de incompatibilidad comentado, consideramos que existe otra posible incompatibilidad entre la GNU GPL y la Original BSD License. La primera licencia estipula que «para un trabajo ejecutable, el código fuente completo significa todo el código fuente de todos los módulos que contiene»; es decir, la GPL exige que, al distribuir el *software* licenciado bajo sus términos en formato ejecutable, se entregue el código fuente completo de dicho ejecutable.

Por su parte la OBL, al establecer las condiciones de redistribución

---

34. FSF, «El problema de la Licencia BSD», disponible en <<http://www.gnu.org/philosophy/bsd.es.html>>.

35. FTP. William Hoskins, disponible en <<ftp://ftp.cs.berkeley.edu/pub/4bsd/README.Impt.License.Change>>.

36. FSF, «El problema de la Licencia BSD».

37. FSF, «Various Licenses and Comments about Them».

del código binario,<sup>38</sup> señala que: «la redistribución en formato binario debe reproducir el anterior aviso de *copyright*, esta lista de condiciones y la siguiente renuncia en la documentación y/u otros materiales provistos con la distribución».

Observamos que la OBL no exige, para esta distribución, que se acompañe la versión en código fuente. ¿Estamos ante un posible conflicto entre ambas licencias que desencadene su incompatibilidad? Aunque en apariencia la respuesta pareciera ser afirmativa, en definitiva creemos que no genera incompatibilidad alguna, ya que es posible cumplir con ambas licencias simplemente entregando voluntariamente el código fuente de la distribución binaria del código BSD (toda vez que no sea un acto prohibido por ésta) para cumplir, a su vez, con la exigencia de la GNU GPL.

IBM PUBLIC LICENSE, VERSIÓN 1.0 (IPL)<sup>39</sup>

La IBM Public License (IPL) fue la primera licencia de código abierto de IBM. La idea detrás de la redacción de la IPL fue «apoyar y fomentar el desarrollo colaborativo en código abierto del código base, al tiempo que maximiza la capacidad del código para ser utilizado e integrado con el *software* licenciado bajo otras licencias, incluyendo muchas licencias comerciales».<sup>40</sup>

La Free Software Foundation, luego de analizar la IPL, ha concluido que se trata de una licencia libre pero que «desafortunadamente contiene una cláusula de elección de ley que la hace incompatible con la GNU GPL»;<sup>41</sup> cláusula que encontramos en el último párrafo de la sección

---

38. El código binario «consta de dos únicos símbolos, 0 y 1, denominados *bits*» (Soler, 2004: 46-47). Sin embargo, dicho lenguaje cayó en desuso «por su dificultad y complicación», siendo reemplazado por nuevos lenguajes de alto nivel o evolucionados (como Fortran, Cobol, Pascal, C). Así «el programa escrito en lenguaje de alto nivel recibe el nombre de ‘código fuente’, mientras que su conversión en lenguaje de máquina se denomina ‘código objeto’» (Soler, 2004: 46-47), «también llamado código ejecutable» (Hardings y Fuentes, 2003: 1).

39. IBM, «IBM Public License», disponible en <<http://www.research.ibm.com/dx/src-Download/license.html>>.

40. IBM, «Common Public License (CPL): Frequently asked questions».

41. FSE, «Various Licenses and Comments about Them».

séptima de la IPL, que estipula: «este acuerdo se rige por las leyes del Estado de Nueva York y las leyes sobre propiedad intelectual de Estados Unidos». Esta cláusula establece un requisito que va más allá de lo exigido por la GNU GPL, siendo incompatible específicamente con la sección sexta de la GPLv2, sección que prohíbe imponer a los receptores del programa licenciado mayores restricciones en el ejercicio de sus derechos que los contemplados por la antedicha licencia.

No obstante lo anterior, estamos de acuerdo con ciertos autores que señalan que la IPL además «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» (Bain, Gallego, Martínez Ribas y Rius, 2004: 39).

El primer requerimiento lo encontramos en la cláusula de distribución comercial (cláusula cuarta), la cual —en términos generales— permite distribuir el programa (licenciado bajo la IPL) en un producto comercial, pero en tal caso el colaborador que lo distribuye (que pasa a llamarse colaborador comercial) debe evitar crear cualquier tipo de responsabilidad a los demás colaboradores. Ahora, señala la IPL, en caso de que se reclame efectivamente dicha responsabilidad por un tercero, el colaborador comercial debe defender e indemnizar cualquier pérdida al colaborador afectado. Algunos autores han señalado, al revisar esta cláusula de indemnización a colaboradores, que «se adecúa más al marco legal de la protección del consumidor, bajo las cuales las exclusiones de garantías y responsabilidades no son completamente válidas» (Bain, Gallego, Martínez Ribas y Rius, 2004: 38), conclusión a la cual adherimos. Observamos que, claramente, se trata de una exigencia impuesta por la IPL que va más allá del umbral de requerimientos permitidos en la GPL.

El segundo requerimiento incompatible es la obligación de cada colaborador de otorgar una licencia amplia sobre las patentes (cláusula 2b); exigencia no contemplada por la GPLv2 y que, por ende, genera incompatibilidad con la sección sexta de dicha licencia. Sin embargo, en relación con la GPLv3, esta incompatibilidad desaparece, ya que en dicha versión se incorpora una cláusula de patentes similar; mas continúa la incompatibilidad en relación con la posible obligación de indemnizar contemplada en la cláusula sobre distribución comercial.

Además de lo anterior, hemos distinguido otro elemento generador de incompatibilidad en la IPL. Por una parte, la licencia en comento expresa

lo siguiente: «las contribuciones no incluyen adiciones al programa que: i) sean módulos separados de *software* distribuidos en conjunto con el programa bajo su propio acuerdo de licencia». Es decir, la IPL reconoce que código independiente puede ser distribuido (bajo su propia licencia) en conjunto con el programa licenciado bajo la IPL.

Sin embargo, observamos que la GPL impone lo siguiente: «estos requisitos se aplican a la obra modificada como un todo. [...] Cuando usted distribuya las mismas secciones [partes no derivadas del programa] como parte de un todo que es una obra basada en el programa, la distribución del todo debe ser, según los términos de esta licencia [...] con independencia de quien lo escribió». La anterior cláusula establece que si el código independiente es distribuido en conjunto con el programa licenciado bajo la GPL, ambos pasan a constituir ‘un todo’ gobernado por esta última licencia.

En definitiva, el código no-GPL se convierte (por el solo hecho de distribuirse en conjunto con código-GPL) en código licenciado bajo la GPL; situación totalmente contraria a lo dispuesto por la IPL, en donde el código no-IPL distribuido en conjunto con código IPL continúa bajo su propia licencia. Todo lo anterior conlleva a la imposibilidad de cumplir con ambas licencias al mismo tiempo y, por ende, no se permite la combinación de código IPL con código GPL, generando en definitiva la incompatibilidad entre ambas licencias.

#### MOZILLA PUBLIC LICENSE 1.1 (MPL)<sup>42</sup>

La historia detrás de esta licencia comienza cuando la empresa Netscape decide redactar dos licencias: la Netscape Public License y la Mozilla Public License; «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» (Bain, Gallego, Martínez Ribas y Rius, 2004: 43).

Luego de analizar la licencia, la Free Software Foundation ha concluido que estamos ante una licencia de *software* libre la cual no tiene un

---

42. Mozilla, «Mozilla Public License», disponible en <<http://www.mozilla.org/MPL/1.1/index.txt>>.

*copyleft* fuerte, porque la MPL exige que la distribución del código cubierto por ella (tanto en formato fuente como ejecutable) deba realizarse bajo la misma licencia. Sin embargo, debemos tener presente la posibilidad de que el código cubierto sea distribuido dentro de una ‘obra mayor’<sup>43</sup> (la cual puede distribuirse bajo cualquier otra licencia), situación que lleva a concluir que la MPL presente un *copyleft* más bien débil.<sup>44</sup> En resumen, «la MPL tiene un *copyleft* limitado; es más *copyleft* que la familia de licencias BSD [...] pero menos que la LGPL o la GPL».<sup>45</sup>

Además, la FSF concluye que la MPL «tiene algunas restricciones complejas que la hace incompatible con la GNU GPL».<sup>46</sup> Revisemos cuáles son, desde nuestro punto de vista, aquellas ‘restricciones complejas’.

En primer lugar, la MPL finaliza la sección 3.1 con la siguiente advertencia: «usted no podrá ofrecer o imponer ningún término sobre la versión en código fuente que altere o restrinja [...] los derechos de los receptores concedidos a continuación». Dentro de esos derechos ‘concedidos a continuación’ por la MPL se incluye la posibilidad de incluir el código cubierto con cualquier otro código dentro de una ‘obra mayor’. Este derecho no lo contempla la GPL para sus receptores. Por ende, al aplicar la GPL, se elimina el derecho concedido por la MPL, lo que constituye una violación a sus términos. Es decir, ambas licencias son incompatibles.

En segundo lugar, observamos que la GPL dispone lo siguiente: «cada vez que redistribuya el programa (o cualquier trabajo basado en el programa), el receptor recibe automáticamente una licencia del licenciataro original para copiar, distribuir y modificar el programa, de forma sujeta a estos términos y condiciones». Es decir, la GPL establece que al redistribuir el programa (o cualquier obra derivada de ella), se genera una nueva licencia GPL en favor del receptor de éste para copiar, distribuir o

---

43. Una ‘obra mayor’ es «una obra que combina código cubierto [por la MPL] o porciones de éste con código no gobernado por los términos de esta licencia» (Mozilla, «Mozilla Public License», cláusula 1.7).

44. Al respecto, Bain, Gallego, Martínez Ribas y Rius (2004) comentan que «es muy fácil crear un archivo adicional propietario que llame al código original bajo MPL y distribuirlo [como obra mayor] todo bajo una licencia propietaria. [...] Sin embargo, en todos los casos, el código fuente de la parte libre original debe distribuirse u ofrecerse al destinatario» (17).

45. Mozilla, «MPL 1.1 FAQ», disponible en <<http://www.mozilla.org/mpl/1.1/faq.html>>.

46. Mozilla, «MPL 1.1 FAQ».

modificar el programa. Sin embargo, la MPL permite que el código cubierto (en formato ejecutable o dentro de una obra mayor) sea distribuido bajo cualquier licencia, las cuales pueden no permitir a los receptores copiar, distribuir o modificar el código recibido, lo que finalmente genera incompatibilidad entre los términos de ambas licencias.

Además, cuando la MPL concede la posibilidad de distribuir el código cubierto en formato ejecutable —bajo cualquier licencia— sólo requiere que se otorgue el código fuente del código cubierto (mas no del ejecutable en su totalidad), lo que se contrapone con la siguiente exigencia de la GPLv2: «para un trabajo ejecutable, el código fuente completo significa todo el código fuente de todos los módulos que contiene». Observamos que las exigencias respecto a la distribución en formato ejecutable de ambas licencias son, nuevamente, incompatibles: por una parte la MPL sólo exige la entrega del código fuente del código cubierto y, por otra, la GPL exige la entrega de todo el código que comprende el ejecutable (sea el código cubierto y/o cualquier código agregado).

En tercer lugar advertimos que la MPL considera que «si se agrega un nuevo archivo que no contiene el código original, no es una modificación», es decir, si se distribuye una obra que contiene el código cubierto por la MPL más una adición en código no-MPL, el conjunto no es considerado (por la MPL) como obra derivada y, por ende, podrá distribuirse bajo cualquier licencia. Esto produce una evidente contradicción con la exigencia presentada por la GPL, la cual requiere que cualquier obra derivada sea licenciada bajo sus propios términos, por lo que existe una incompatibilidad entre ambas licencias.

Finalmente consideramos que existen otros elementos menores<sup>47</sup> que implican, igualmente, una infracción a la sección sexta de la GPLv2, ya que se trata de restricciones adicionales impuestas a los receptores no contempladas por la antedicha licencia.

No obstante lo anteriormente señalado, la FSF sí reconoce que esta licencia «tiene una salvedad (sección 13) que permite que un programa (o partes de él) ofrezca otra licencia como alternativa. Si una parte de un programa permite a la GNU GPL como una licencia alternativa, o

---

47. En específico, estamos hablando de las restricciones en el uso de ciertos nombres en las versiones modificadas de la licencia contenidas en la cláusula 6.3 y la cláusula 11 sobre elección de jurisdicción.

cualquier otra licencia compatible con ella como alternativa, esa parte del programa es compatible con la GPL». <sup>48</sup>

Esta cláusula especial de la MPL, titulada ‘Código multilicenciado’, dispone lo siguiente:

El desarrollador inicial puede designar porciones del código cubierto como ‘multilicenciado’. ‘Multilicenciado’ significa que dicho desarrollador inicial le permite utilizar porciones del código cubierto bajo su elección entre la NPL o la licencia alternativa especificada por el desarrollador inicial en el archivo descrito en el Anexo A.

En definitiva, esta sección de la MPL permite al desarrollador inicial establecer que cierto código podrá utilizarse bajo una licencia alternativa (distinta a la MPL); si dicha licencia alternativa es o la GNU GPL u otra licencia compatible con ésta, ese código será compatible con la GPL, manteniéndose el resto del código licenciado como incompatible con la antedicha licencia (ya que se encuentra gobernado por la MPL, licencia incompatible).

Actualmente, Mozilla ha redactado la versión 2.0 de la presente licencia, versión que es compatible con la GNU GPL.

## CONCLUSIÓN

Luego de revisar las razones de incompatibilidad entre licencias esgrimidas por la FSF, observamos que la principal razón de incompatibilidad se encuentra en la sección sexta de la GPL, es decir, la incompatibilidad surge porque la licencia impone mayores restricciones a los usuarios que al recibir un *software* GPL. Siendo éste el problema, la solución más simple sería que la FSF (en una próxima versión de la GNU GPL) otorgara un concepto acabado de lo que se entiende por ‘mayores restricciones’. <sup>49</sup> Sin perjuicio del esfuerzo señalado, hemos observado que dicha institución cataloga a las licencias como incompatibles sin una metodología concreta de fondo, lo que decanta en razones de incompatibilidad imprecisas y

48. FSF, «Various Licenses and Comments about Them».

49. La GPLv3 ya ha realizado un paso en ese sentido al permitir que, al combinar código GPL con código licenciado bajo otra licencia, se permite la imposición de ciertos pactos adicionales (véase FSF, «GNU General Public License, Version 3», cláusula 7.º, disponible en <<http://www.gnu.org/licenses/gpl-3.0-standalone.html>>).

confusas. Así, por ejemplo, la FSF estima como una razón de incompatibilidad que una licencia utilice el término ‘propiedad intelectual’ por ser «desafortunado»;<sup>50</sup> sin embargo, existen varias otras licencias que utilizan dicho término y no es considerado expresamente<sup>51</sup> por la FSF como un elemento de incompatibilidad.

La carencia de un protocolo preestablecido —con pautas objetivas— para considerar una licencia como compatible o incompatible, junto con omitir la entrega de una justificación acuciosa (en la mayoría de las licencias) respecto de la incompatibilidad, nos permiten concluir una tendencia de la FSF a evitar la utilización por parte de los desarrolladores de *software* gobernado por licencias que dicha institución considera (de forma muy subjetiva) inadecuada. Y esta conclusión tiene cierto grado de veracidad toda vez que la FSF tiene por objetivo desalentar el uso de licencias no *copyleft* y siendo la mayoría de las licencias *copyleft* de su autoría, se podría inferir que la FSF intenta desacreditar las licencias que no han sido de su autoría. Lo anterior es de mayor gravedad si consideramos que la guía presentada por la FSF es una orientación de importancia para cientos de desarrolladores de *software*, a nivel mundial, a la hora de escoger una licencia para gobernar su creación. En dicho sentido es importante recalcar que la lista de licencias compatibles e incompatibles de la FSF es sólo una pauta realizada desde el punto de vista de la FSF, cuyo valor aún no ha sido debatido ante un tribunal.

Consideramos relevante que la FSF instaure un procedimiento objetivo mediante el cual establezca, con justificaciones de peso, por qué se considera que cierta licencia es incompatible. En ese sentido la Open Source Initiative ha llevado la delantera, ya que esta organización posee un comité encargado de revisar las licencias que solicitan su aprobación como licencia de código abierto, bajo un procedimiento conocido.<sup>52</sup>

---

50. En específico al analizar la Common Development and Distribution License (CDDL) (en FSF, «Various Licenses and Comments about Them»).

51. Así sucede, por ejemplo, al presentar la licencia libre compatible CeCILL versión 2. En dicha situación la propia FSF se arroga la facultad de decidir que la utilización del término ‘propiedad intelectual’ fue «una decisión inapropiada [...] [que] no causa ningún problema en particular» en relación con dicha licencia (FSF, «Various Licenses and Comments about Them»).

52. En relación con este tema, véase Open Source Initiative, «The License Review Process», disponible en <<http://www.opensource.org/approval>>.

## REFERENCIAS

- APARICIO, Juan Pablo (2004). *Licencias de uso no personalizadas de programas de ordenador*. Granada: Comares.
- BAIN, Malcolm, Manuel GALLEGO, Manuel MARTÍNEZ RIBAS y Judit RIUS (2004). *Aspectos legales y de explotación del software libre*. Barcelona: UOC. Disponible en <[http://materials.cv.uoc.edu/contin-guts/XW08\\_M2114\\_00341/index.html](http://materials.cv.uoc.edu/contin-guts/XW08_M2114_00341/index.html)>.
- BARO, José Luis (2007). *Ingeniería inversa de software y derecho de autor*. Tesis (Licenciatura en Ciencias Jurídicas y Sociales). Universidad de Chile, Facultad de Derecho. Santiago: Inédita.
- CARRANZA, Martín (2004). *Problemática jurídica del software libre*. Buenos Aires: Lexis Nexis.
- CARRASCOSA, Valentín, María Asunción POZO y E. P. RODRÍGUEZ DE CASTRO (2000). *La contratación informática: el nuevo horizonte contractual*. Granada: Comares.
- DAVARA, Miguel Ángel (2005). *Manual de Derecho Informático*. Navarra: Aranzadi.
- FERNÁNDEZ, Enrique (1996). *La protección internacional de los programas de ordenador*. Granada: Comares.
- GOMEZ PADILLA, Oscar (2011). *La validez jurídica de la licencia GPL versión 3 en el marco normativo de los derechos de autor en Colombia*. Colombia: Autoedición. Disponible en <<http://www.etnassoft.com/biblioteca/validez-juridica-de-la-licencia-gplv3/>>.
- GONZÁLEZ, Jesús, Joaquín SEOANE y Gregorio ROBLES (2004). *Introducción al software libre*. Barcelona: UOC. Disponible en <[http://cv.uoc.edu/~fcaulas/20041/90.783/portada\\_Into.pdf](http://cv.uoc.edu/~fcaulas/20041/90.783/portada_Into.pdf)>.
- HARDINGS, Jens (2003). *Efectos de las patentes de software*. Disponible en <<http://www.hardings.cl/publications/hardings2003patentes.pdf>>.
- HARDINGS, Jens y Alejandro FUENTES (2003). *Software libre: introducción histórica*. Disponible en <<http://www.hardings.cl/publications/hardings2003intro.pdf>>.
- HERNANDO, Isabel (1995). *Contratos informáticos: derecho informático, legislación y práctica*. San Sebastián: Carmelo.
- HERRERA, Dina (1999). *Propiedad intelectual, derechos de autor: Ley no. 17.3336 y sus modificaciones*. Santiago: Jurídica de Chile.

- LIPSZYC, Delia (1993). *Derecho de autor y derechos conexos*. Buenos Aires: Unesco-Cerlalc.
- NONIUS, Jorge (2002). *Introducción a las licencias de software libre*. Disponible en <<http://quark.fe.up.pt/laespiral/articulos/licencias/licencias.html>>.
- ORGANIZACIÓN MUNDIAL DE LA PROPIEDAD INTELECTUAL, OMPI (1996) «*Implicaciones del Acuerdo sobre los ADPIC en los tratados administrados por la OMPI*». Ginebra: OMPI.
- PERENS, Bruce (1999). «The Open Source Definition». Disponible en <<http://oreilly.com/catalog/opensources/book/perens.html>>.
- RAMÍREZ, Ramón (2005). *Gestión del desarrollo de sistemas de telecomunicación e informáticos*. Madrid: Paraninfo.
- SATANOWSKY, Isidro (1954). *Derecho Intelectual*. Buenos Aires: Tipográfica.
- SOLER, Pere (2004). *El contrato para la elaboración de programas de ordenador*. Cizur Menor: Aranzadi.
- STALLMAN, Richard (2004). *Software libre para una sociedad libre*. Madrid: Traficantes de Sueños.
- VIDAL, Miquel (2008). *Informe sobre licencias libres*. Disponible en <<http://gsyc.es/~mvidal/docs/ikusnet.pdf>>.

#### **SOBRE LA AUTORA**

NAYARETH GUERRA GACITÚA es egresada de la Facultad de Derecho de la Universidad de Chile. El presente artículo es una versión resumida de su memoria para optar al grado de Licenciada en Ciencias Jurídicas y Sociales de esa casa de estudios, que fue evaluada con distinción máxima. Su correo electrónico es <[nayarethguerra@gmail.com](mailto:nayarethguerra@gmail.com)>.

Este trabajo fue recibido el 28 de agosto de 2012 y aprobado el 19 de mayo de 2013.

