

# La GPLv3: Copyleft para el siglo XXI

Miquel Vidal

25 de enero de 2008

## Resumen

La Licencia Pública General (*General Public License*), conocida popularmente por sus siglas GPL, es la plasmación jurídica del concepto de *copyleft* en los programas de ordenador. Escrita originalmente por Richard Stallman en 1989 como parte del Proyecto GNU, es la licencia de software libre por antonomasia, la más antigua y la más usada. En el verano de 2007, después de más de 15 años desde su última revisión, fue aprobada la versión 3 de la GPL, tras un largo proceso de debate y enmiendas que duró año y medio y concitó la atención de gran parte de la comunidad del software libre.

## 1. Fundamentos de la GPL

El código informático está protegido en todos los países por los derechos de autor (llamado *copyright* en el ámbito anglosajón), como cualquier otra obra creativa del intelecto. Las llamadas *licencias libres* se basan por tanto, como cualquier otra licencia de software, en la legislación del derecho de autor y se comportan a todos los efectos como cualquier otra licencia de software salvo que otorga ciertas libertades a su receptor, en lugar de restringírselas. ¿Y cuáles son esas libertades que el software libre garantiza a todo usuario, sin necesidad de permiso previo?

- 0 Libertad de ejecutar el programa como se desee, con cualquier propósito (libertad de uso).
- 1 Libertad para estudiar el código fuente y modificarlo para que haga lo que se desea o se necesita que haga (libertad de modificación).
- 2 Libertad para hacer copias y distribuir las a otros (libertad de copia).

### 3 Libertad de publicar o distribuir versiones modificadas (libertad de redistribución).

Las licencias libres son el dispositivo jurídico para implementar legalmente las cuatro libertades citadas para cualquier propósito. En otras palabras, las licencias libres son el dispositivo legal que convierte un programa de ordenador en software libre. De ahí su importancia.

Para considerarse software libre, deben darse las cuatro libertades citadas de forma simultánea, no basta con que se concedan algunas de ellas (por ejemplo, una cláusula que impida su venta o restrinja su uso por razones de cualquier tipo, no es software libre). Hay muchas licencias de software libre, cualquiera puede escribir una, basta con que conceda las cuatro libertades mencionadas. Las licencias públicas libres tienen la ventaja de que ahorran este trabajo al programador, garantizan la compatibilidad entre el software bajo la misma licencia y ofrecen tranquilidad a los usuarios sobre su consistencia jurídica.

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

#### 1.1. Copyleft

La Free Software Foundation considera que no basta con conceder las cuatro libertades citadas, sino que debe asegurarse que todo usuario que obtenga una copia del programa reciba estas mismas libertades. Por esta razón, además de las cuatro libertades mencionadas arriba, común a todas las licencias libres, la GPL incorpora una cláusula conocida como *copyleft*.

El copyleft requiere que cualquier obra modificada conserve la licencia original. De ese modo, garantiza que toda obra derivada a partir de software protegido por la GPL conservará siempre las mismas libertades. Como otras cláusulas que veremos más adelante, el copyleft no persigue restringir derechos al usuario, sino garantizar que el software libre bajo GPL no deje de serlo. Por eso a esta cláusula se la conoce también como «licencia recíproca» o «compartir-igual» (*share-alike*). El copyleft es pues una forma de defender activamente las libertades de los usuarios.

## 2. Historia de la GPL

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

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

## 3. La GPLv3

Desde 1991, fecha en que se publicó la versión 2 de la GPL, el contexto tecnológico y social en relación a la informática ha cambiado enormemente. En esos quince años ha nacido Linux, los entornos gráficos, se ha globalizado Internet y extendido los ordenadores personales. De ser algo marginal y solo conocido en unas cuantas universidades estadounidenses, el software libre se ha convertido en una alternativa real a la industria del software tradicional. Además, el cambio de contexto tecnológico ha provocado cambios en las legislaciones y nuevos conflictos legales, por ejemplo en relación a las patentes de programación, a los sistemas de gestión de derechos digitales (DRM, *Digital Rights Management*, también conocido como *Digital Restrictions Management*) y, en general, a los derechos de los usuarios a usar libremente la tecnología que adquieren. En resumen, tras más de 15 años, hacía falta actualizar la GPL para que siguiese siendo eficaz ante los desafíos tecnológicos del siglo XXI.

El proceso de elaboración tuvo lugar entre el 16 de enero de 2006 en que se publicó el primer borrador y el 29 de junio de 2007 en que fue publicada la versión final. Durante ese año y medio se publicaron cuatro borradores y se celebraron cinco conferencias internacionales (Boston, Porto Alegre, Barcelona, Tokio y Bruselas). La forma de reelaboración de la licencia, discutida y consensuada por la comunidad, fue un cambio cualitativo respecto a versiones anteriores, cuya elaboración fue básicamente un trabajo individual.

### **3.1. Mismos objetivos**

La nueva versión de la GPL no anula la anterior ni obliga a relicenciar el software a la nueva versión. Y, aunque son incompatibles (la GPLv2 no permite añadir ningún tipo de requisito que no contenga ella misma), no impide en modo alguno que programas con distintas versiones de la GPL se distribuyan juntos e incluso que haya código fusionado con GPLv3 con sus propios requisitos. Además, hay mucho software libre bajo licencia GPLv2 que indica que es aplicable también a «cualquier versión posterior» de la licencia, lo que da la opción de seguir también los términos de la GPLv3 si se desea.

La GPLv3, en cualquier caso, mantiene intactos los mismos objetivos que las versiones anteriores: garantizar de la forma más eficaz posible las cuatro libertades del software libre. Simplemente, añade algunos ajustes que permiten a desarrolladores, usuarios y distribuidores defenderse de amenazas aparecidas en los últimos años, sobre todo en relación a dos cuestiones, que son abordadas de forma explícita desde el mismo preámbulo de la nueva licencia: la gestión de restricciones digitales y las patentes de programación. Es en estas dos cuestiones en donde se han concentrado las discusiones y donde las novedades son más sustanciales.

### **3.2. La gestión de restricciones digitales**

El modo de afrontar el desafío de los DRM fue quizá el tema que más polémica despertó y el que absorbió más energías para lograr una solución equilibrada que garantice tanto la libertad de los usuarios como la de los fabricantes de dispositivos empotrados. En particular, la GPLv3 busca evitar que el hardware que implemente software libre empotrado no imponga restricciones al usuario vía DRM, impidiéndole actualizar o modificar dicho software. Los sistemas TiVO (los sistemas de grabación de TV en disco duro) utilizan software empotrado e incorporan tecnología para limitar las cosas que puede hacer el usuario con dicho software, entre ellas impedirle ejecutar

versiones modificadas del software (además de otras cuestiones más escabrosas, como límites a la grabación e incluso monitorización de las acciones del usuario). Incluso existen leyes (como el caso de la DMCA estadounidense y la EUCD europea) que penalizan la modificación de dicho software (por ejemplo, para retirar limitaciones impuestas por el fabricante o para añadirle funcionalidades no previstas). De ese modo, los sistemas TiVO convierten en papel mojado, dejando sin efecto, varias de las libertades del software libre. Este tipo de sistemas se están aplicando de forma masiva en numerosos dispositivos con software libre incorporado (como grabadoras de TV, teléfonos móviles, reproductores multimedia, etc.), convirtiendo en papel mojado la GPL.

Los promotores de la licencia GPL asumen, por tanto, que los sistemas de DRM son incompatibles con el ejercicio de las libertades de los usuarios. Por ello, la GPLv3 se concentra en evitar que el fabricante impida al usuario instalar o ejecutar versiones modificadas del programa GPL que dichos dispositivos usan internamente. Si existen tales limitaciones (por ejemplo binarios firmados digitalmente), el fabricante deberá facilitar los mecanismos para que las versiones modificadas puedan firmarse y funcionen igual que si fuese la original. Es importante puntualizar que la GPLv3 no prohíbe a los fabricantes incorporar dispositivos DRM con software libre, pero garantiza jurídicamente que el usuario está autorizado a modificar o retirar dichas limitaciones. Como siempre, la GPL no restringe lo que personas y empresas pueden hacer con el software: solo se les impide que traten de restringir la libertad de otros.

### **3.3. Las patentes de programación**

Como ya se ha expuesto, el software está protegido por las leyes de derechos de autor (o copyright). Sin embargo, en algunos países (particularmente EE.UU y Japón) se permite también patentar métodos (algoritmos) de software. Pese a que ha habido presiones para legalizarlas, en la Unión Europea las patentes de programación no son válidas, ya que a los algoritmos se les considera ideas (del mismo tipo que los métodos matemáticos), y las ideas no pueden patentarse. Las patentes de software son desde hace años objeto de gran controversia. Desde el ámbito del software libre, se objeta que, a diferencia del derecho de autor que se ha demostrado útil para proteger tanto el software libre como el software privativo, las patentes de software son dañinas tanto para la industria informática como para los desarrolladores y los usuarios, ya que favorecen los monopolios, no aportan ninguna protección que no exista ya y ponen trabas a la innovación y a la libertad de mercado (se usan a menudo como forma de intimidación de la competencia). En par-

ticular, las patentes pueden convertir un programa libre en privativo en la práctica.

Una licencia no puede evitar el problema de las patentes de programación allí donde son operativas, pero sí puede minimizar sus daños, y eso es lo que ha procurado realizar la GPLv3, asegurándose de que las patentes no puedan usarse para convertir un programa en no-libre.

También se discutió mucho acerca de incluir en la GPLv3 cláusulas de represalias por patentes (*Patent retaliation clauses*). Este tipo de cláusulas pretenden proteger al autor y repeler pleitos de patentes por parte del licenciatario (el beneficiario), aunque finalmente se excluyeron por su excesiva complejidad a la hora de establecer requisitos para que no se usasen de forma agresiva.

Además de los dos asuntos tratados, también fueron objeto de atención algunos temas de menor calado, en concreto respecto a la internacionalización de la licencia (evitando términos que diesen lugar a interpretaciones diferentes), respecto a clarificaciones semánticas para facilitar su uso y comprensión y respecto a la compatibilidad con otras licencias de software libre.

### 3.4. Compatibilidad

Por su propia naturaleza, las licencias copyleft estrictas son poco compatibles con otras licencias libres, pues generalmente obligan a que cualquier trabajo fusionado con licencia copyleft mantenga la licencia copyleft e iguales condiciones, sin añadir otras nuevas (incluso aunque sean cláusulas inocuas). La GPLv3 ha tenido esto en cuenta, tratando de evitar incompatibilidades innecesarias e incrementando las posibilidades de compatibilidad con otras licencias libres. Esto lo logra con una concepción más modular que permite algunos requerimientos adicionales, concretamente admite cláusulas de garantía y de responsabilidad y también permite imponer limitaciones al uso de nombres o marcas publicitarias, ya que realmente es algo que concierne a la legislación de marcas, no a la licencia en sí.

## 4. Conclusión

La GPLv3 introduce muchos cambios, pero es más una evolución que una revolución y mantiene intactos los mismos objetivos que las versiones anteriores: garantizar de la forma más eficaz posible las cuatro libertades del software libre a desarrolladores y usuarios. Para ello, se ha actualizado con el fin de afrontar diferentes retos tecnológicos y jurídicos que han aparecido

en los últimos años, muy especialmente las patentes de programación y los sistemas de restricción de derechos digitales (DRM).

La GPLv3 añade una mayor protección mediante ingeniería jurídica ante la amenaza de las patentes: si cualquier usuario pretende ejercer un derecho de patentes contra otro mediante la GPL, su licencia de uso se termina. La GPLv3 también neutraliza la *tivoización* (que dejaba en papel mojado las licencias libres), ya que obliga al fabricante a proveer una forma en que el usuario pueda ejercer su derecho a la modificación del software GPL integrado.

Aunque el asunto de las patentes y del DRM han acaparado la mayoría de esfuerzos por ser amenazas directas a la misma existencia del software libre, probablemente el cambio mayor haya sido la forma de reelaboración de la licencia: mientras las versiones 1 y 2 fueron obra prácticamente individual de Richard Stallman, la versión 3 ha sido objeto de discusión pública, tanto en la Red como en encuentros presenciales, a lo largo de año y medio, con participantes a nivel global de toda la comunidad de software libre, no solo norteamericana, también europea, asiática e iberoamericana, lo que, aparte de dotarle de gran consistencia jurídica, ha demostrado un grado de madurez global sin precedentes por parte de la comunidad del software libre en un asunto tan delicado y complejo como este.

## 5. Bibliografía

- Versión final de la licencia: <http://www.gnu.org/licenses/gpl-3.0.html>
- Traducción no oficial al español: <http://www.viti.es/gnu/licenses/gpl.html>
- GPLv3: Creando la 3ª versión de la General Public License de GNU: <http://www.fsfeurope.org/projects/gplv3/gplv3.es.html>
- A Quick Guide to GPLv3: <http://www.gnu.org/licenses/quick-guide-gplv3.html>

Copyright © 2008 Miquel Vidal

Este artículo se publica bajo licencia libre Creative Commons Reconocimiento 2.5