El negocio del software: licencias, derechos y alternativas

Vicente Matellán Olivera, Jesús González Barahona,
Pedro de las Heras Quirós, José Centeno González, Francisco Ballesteros Cámara
gsyc-profes@gsyc.inf.uc3m.es
Departamento de Informática, Universidad Carlos III de Madrid

9 de febrero de 2005

Resumen

El mercado actual del software está construido sobre las licencias "propietarias". Estas licencias limitan férreamente (usando la legislación sobre *copyright*) los derechos de quien recibe un programa, prohibiendo por ejemplo su copia, su redistribución, e incluso su uso en ciertas circunstancias. Frente a este modelo, el llamado "software libre" propone una nueva forma de desarrollar y distribuir software, basada en permitir explícitamente la copia, la redistribución, y el acceso al código fuente. En esta ponencia se exponen algunas de las implicaciones de este nuevo modelo sobre el negocio del software, y se analizan las licencias más comunes utilizadas en él.

1. Introducción: el derecho a copiar software

Desde hace tiempo la industria del software está realizando muchos esfuerzos por informar a la sociedad de los perjuicios que produce la copia ilegal de software. Sin embargo, existen modelos de desarrollo de software basados precisamente en el derecho de los usuarios a copiar y redistribuir programas, con muy pocas restricciones.

De hecho, si lo analizamos sin prejuicios, podría decirse que la copia de software no perjudica directamente a ninguno de los que intervienen en ella. El software puede "duplicarse" (copiarse) sin que se degrade en ninguna medida la información original, y teniendo el duplicado (la copia) exactamente la misma calidad. Ninguna de las partes que intervienen en la copia (el individuo que tiene el original y el que recibe la copia) están motivados para no hacerla, dado que ninguno de ellos pierde nada, sino más bien al contrario (especialmente quien la recibe). Por supuesto, están también los derechos del autor de ese software, que sí podría resultar perjudicado. Pero al no intervenir directamente en la copia (salvo, naturalmente, en la "primera"), no puede evitarla de forma directa. Es importante reconocer que los problemas que se plantean en este

caso son radicalmente diferentes de los que plantea la propiedad privada, dado que aún no se han descubierto formas de duplicar bienes materiales, a coste prácticamente cero.

Hay una analogía muy interesante, que a menudo repite Richard Stallman[1]. Es la de un nuevo invento que permite copiar rebanadas de pan, con coste prácticamente cero. Las nuevas rebanadas, resultado del proceso, son exactamente iguales que las "originales". Para tener una nueva rebanada, todo lo que hace falta es que alguien te preste momentáneamente la suya. Es sencillo entender que con este invento cualquiera estaría dispuesto a prestar su rebanada para obtener nuevas rebanadas... El proceso de copia de información es muy parecido. Sin embargo, tendemos a pensar en él de forma diferente a lo que posiblemente pensaríamos de este "copiador de pan".

Antes de discutir los motivos que pueden aconsejar la restricción de los derechos de copia, vamos a presentar brevemente algunos ejemplos que refuerzan la idea de que la copia de información es algo habitual en nuestra experiencia diaria:

- Las recetas de cocina, que se transmiten de boca en boca.
- Las teorías científicas (y su soporte matemático), cuya "copia" permite el aprendizaje y el progreso científico.
- El cotilleo, donde la información se "copia" con la seguridad de que será a su vez "copiada".
- Los exámenes no competitivos (donde no hay un número predeterminando de aprobados, y todos los examinandos pueden tender a "dejarse copiar").
- En general, la difusión de la cultura.

Pero si la copia de información es tan sencilla y aparentemente beneficiosa para todos, ¿por qué la sociedad permite prohibir la copia?. Pues bien, esa prohibición es algo relativamente moderno, que se puede trazar a las primeras legislaciones sobre *copyright* (derechos de copia). En el resto de esta ponencia vamos a exponer los orígenes de esta legislación, cómo afecta a los métodos de desarrollo software más habituales y las alternativas que proponen las licencias de software libre.

2. Los derechos de copia (*copyright*)

En general, se tiende a justificar su existencia como un balance entre el derecho a copiar información (natural como se ha justificado en el apartado anterior) y la necesidad de asegurar que los autores son recompensados adecuadamente, para aumentar su motivación, y se supone que por tanto su productividad.

[1] Richard Stallman es el fundador de la Free Software Foundation y del proyecto GNU, y uno de los primeros promotores del modelo de software libre.

La necesidad de esta legislación no surgió hasta la aparición de la imprenta. Hasta entonces el único método de reproducción de la información era la copia manual, casi tan costosa como la producción de una obra nueva. De hecho no hubo legislación hasta dos siglos después de la aparición de la imprenta (en Inglaterra fue la primera y el resto se han inspirado en ella).

Hasta entonces el férreo control de la iglesia y el gobierno vigilaban la impresión de libros, prohibiendo la publicación de aquellos "no adecuados". El mecanismo era sencillo, solo estaban autorizados para imprimir libros ciertos impresores que se comprometían a no imprimir esas obras.

Eso generó rápidamente un monopolio, formado en Inglaterra por la *Stationers' Company* compuesta por aquellos impresores con licencia. Dicho monopolio fijó un registro central en el que los impresores inscribían el libro que tenían intención de publicar. Una vez inscrito un libro por un impresor ningún otro podía imprimirlo. Esto generaba problemas muy importantes:

- El autor no tenía ningún derecho sobre su obra, de hecho si el editor no llevaba a cabo su intención, el autor no tenía forma de imprimirlo en ningún otro sitio.
- El precio de los libros era muy elevado, pues no existía competencia entre editores. Lo que dificultaba la extensión de la cultura.

En 1710 el parlamento inglés promulgo la primera legislación sobre copyright, conocida como: "Statute of Anne". Su sentido era fomentar el desarrollo de la cultura, muy diferente al propósito de las normas de *Stationers* que simplemente protegían a los editores de la competencia. Así, en el título de la nueva ley se especificaba que la legislación se establecía para "*for the encouragement of Learning*".

Para conseguir el abaratamiento de los libros, se traspasaron los derechos de las obras de los editores a los autores, favoreciendo la competencia entre imprentas. Con el mismo objetivo de fomentar la publicación de nuevos libros, se especificó que los derechos del autor solo eran válidos sobre obras originales. Por último, para garantizar de nuevo la mayor extensión de la cultura se fijó una duración máxima de vigencia para esos derechos, 14 años únicamente en aquel tiempo (ahora llega hasta los 50 o más años después de la muerte del autor).

Hay que hacer notar que en aquel tiempo los autores no tenían los conocimientos, ni el equipo necesarios para editar sus propios libros, ni tampoco para venderlos. La industria editorial surgió de hecho para proporcionar esas funciones y la sociedad reguló los derechos de copia para favorecer la extensión de la cultura.

Desde otro punto de vista, podría entenderse la legislación de *copyright* como un "impuesto" al uso de la información. Ese impuesto es la cantidad que hay que pagar a quien detenta los derechos de copia (copyright), cada vez que esta copia se realiza (por ejemplo, cada vez que se compra una película de vídeo, un disco de música o incluso un libro). Es un impuesto desde el momento que el hecho en sí (sacar una copia de la película, del disco o del libro) sería muy barato, o virtualmente gratis con las tecnologías actuales, y sólo porque la legislación lo

impone hay que pagar un cierto sobrecoste. Curiosamente, este impuesto no lo recibe el estado, sino directa o indirectamente el dueño de los derechos de copia. La suposición implícita es que este dinero extra que se paga permite que se mantenga o incremente (en términos cualitativos o cuantitativos) la producción del tipo de información en cuestión (películas, música o libros, por ejemplo). Otra cosa es que esto sea realmente así, o al menos que sea así en todos los casos y para todos los tipos de información donde se aplica este modelo.

Por lo tanto, la sociedad permite la limitación *legal* del derecho *natural* de copia como una forma de fomentar una mayor producción (cuantitativa y cualitativamente) de información. En cuanto el lector se percata de este hecho, surge inmediatamente la pregunta: "En el caso del software, ¿se beneficia la sociedad de la limitación de los derechos de copia?".

3. La producción del software

La respuesta a la pregunta anterior genera largas discusiones. A nuestro entender, la respuesta es, al menos en muchos casos, "NO". Es más, es muy posible que la aplicación de una legislación diseñada para garantizar los derechos de los intermediarios, generados en una sociedad de hace dos siglos y pensados para objetos materiales (libros) esté en la raíz de muchos de los problemas que se han presentado en la evolución del software. En pocas palabras, podría decirse que la legislación sobre copyright, tal y como se está aplicando al software, está limitando severamente la cantidad y calidad de los programas existentes en la actualidad así como la velocidad de adaptación a las nuevas posibilidades del hardware y de las comunicaciones.

Para sostener esta afirmación, analicemos algunos de los mecanismos habituales de evolución y mejora del software. Desde luego, la lista que ofrecemos a continuación no es exhaustiva, y cada desarrollador de software añadirá (o quitará) algún punto de ella. El enfoque de este análisis se centra en el impacto que produce en cada mecanismo el acceso libre al código fuente, que a nuestro modo de ver es el aspecto esencial (desde el punto de vista técnico).

- **Mejora por imitación**. Es muy común hacer un programa similar a otro que ya existe, mejorando alguno de sus aspectos, pero manteniendo muchos otros. Esto puede hacerse reescribiendo completamente el programa, y de hecho es el caso más común, debido a que normalmente el código fuente del programa imitado no está disponible.

  Si el código fuente estuviese accesible, los "imitadores" podrían concentrarse en las mejoras que quieren realizar, y reutilizar completamente el resto del programa. Al poder concentrar sus esfuerzos en la innovación, esta se realizaría con mucho menos esfuerzo, y a igualdad de recursos, el resultado sería en general mejor. Incluso si se decidiese empezar el nuevo producto de cero, el poder leer y estudiar el código del producto imitado allanaría mucho el camino a sus desarrolladores.

- **Mejora incremental**. Es la forma habitual de mejorar un producto software dado, mediante una sucesión de versiones. La empresa que lo desarrolla parte del código fuente de una versión, y corrigiendo erratas y añadiendo nueva funcionalidad, produce una nueva versión del producto. Naturalmente, si sólo la empresa en cuestión puede usar ese código fuente, sólo ellos pueden realizar mejora incremental sobre el producto.

  Sin embargo, si el código fuente estuviera disponible para otros desarrolladores, muchos más podrían realizar esta mejora incremental. Esto permitiría, por ejemplo, que el producto se desarrollase en direcciones no previstas por sus creadores, y que hubiese competencia en el desarrollo de nuevas versiones, lo que en general no puede sino beneficiar al usuario.

- **Mejora por integración**. En el mundo del software propietario es común que cuando una empresa necesita más funcionalidad en su producto, decida llegar a un acuerdo con otra empresa que ya tenga esa funcionalidad desarrollada, para integrarla directamente (quizás tras una adaptación). De hecho, en muchos casos el procedimiento utilizado es comprar a la empresa productora de esa funcionalidad (lo que tiene la ventaja para el comprador de eliminar, de paso, a un posible competidor). Pero sólo en los casos en que se llegue a este acuerdo, o en los que se usen productos de la propia empresa, se puede en general mejorar software por integración. Y en muchos casos las empresas dedicadas al software propietario están motivadas a no llegar a esos acuerdos, pues pueden ver en el "integrador" a un competidor.

Si este análisis fuese correcto, los modelos de desarrollo software basados en la disponibilidad del código fuente y en el permiso explícito de copia y redistribución permitirían desarrollar software de mejor calidad, y capaz de evolucionar más rápidamente. Ese software existe. Suelen usarse para denominarlo los términos de "software libre" (*"free software"*) o "fuente abierto" (*"open source"*). Pero el mundo del software libre no es uniforme, sino que lo hay de diversos tipos, según la licencia con la que se distribuya.

4. Las licencias de software libre

Con el marco legal actual, la licencia bajo la que se distribuye un programa delimita exactamente los derechos que tienen sobre él sus usuarios. Por ejemplo, en la mayoría de los programas propietarios, la licencia priva al usuario de los derechos de copia, modificación, préstamo, alquiler, uso en varias máquinas, etc. De hecho, las licencias suelen especificar que la propietaria del programa es la empresa editora del mismo, que simplemente vende derechos restringidos de uso del mismo.

Cuando se habla en inglés de "*free software*", hay una peligrosa ambigüedad, debido que "*free*" significa tanto "libre" como "gratis". Afortunadamente, en

castellano no tenemos esta ambigüedad[2], y el significado de "software libre" está mucho más claro. De todas formas, es bueno dejar claro que el software libre no tiene porqué ser gratis. Es más, no suele serlo, o al menos no completamente. Su característica fundamental es la libertad que tiene cualquier usuario para:

- Disponer del software para adaptarlo a sus necesidades. Naturalmente, esto incluye hacerle mejoras, corregir erratas, aumentar su funcionalidad o estudiar su funcionamiento.
- Redistribuir el software a otros usuarios, que a su vez podrán también disponer de él según sus necesidades[3].

Para poder cumplir las dos condiciones anteriores[4], hay una tercera que es básica, y se deriva necesariamente de ellas:

- Los usuarios del software en cuestión deben tener acceso a su código fuente.

Todas las licencias de software libre respetan estas condiciones. Normalmente, los detalles de estas licencias son el resultado de un compromiso entre varios objetivos hasta cierto punto contrapuestos, entre los que se pueden citar los siguientes:

- Garantizar estas condiciones básicas (de redistribución, de modificación, de uso) a los usuarios.
- Asegurar condiciones impuestas por los autores (cita del autor en trabajos derivados, por ejemplo).
- Procurar que los trabajos derivados sean también software libre.

Según el grado con que se quiera cumplir cada uno de estos objetivos, y los detalles que se quieran asegurar, los autores pueden elegir proteger su software con distintas licencias. De hecho, el autor puede (si así lo desea) distribuir su software con licencias diferentes por canales (y con precios) diferentes[5]. Por lo tanto, el autor de un programa suele elegir con mucho cuidado la licencia bajo la que lo distribuye. Y los usuarios, y especialmente quien redistribuya o modifique el software, debe estudiar con cuidado su licencia.

Afortunadamente, aunque cada autor podría utilizar una licencia diferente para sus programas, en realidad casi todo el software libre se define bajo una

[2] De hecho, en la comunidad software de habla inglesa se está extendiendo el término "libre software", usando el vocablo castellano, para evitar la ambigüedad de "*oree*'.

[3] Esta redistribución puede hacerse gratuitamente (incluso cuando el que hace la redistribución ha pagado por el software) o cobrando un precio, que no está fijado de antemano.

[4] Obviamente, el acceso al código fuente es necesario para poder modificar el software, pero también lo es para que los que reciben redistribuciones puedan, a su vez, hacerlo.

[5] Por ejemplo, hay autores que deciden distribuir su software con una licencia que garantice que los trabajos derivados sean también software libre. Sin embargo, en algunos casos, venden ese mismo software a empresas interesadas en mantener propietarios sus trabajos derivados, bajo una licencia diferente que lo permita.

de las licencias más habituales (GPL, LGPL, Artistic, "estilo" BSD, "estilo" Netscape, etc.). Para simplificar aún más las cosas, en los últimos tiempos están apareciendo organizaciones que promueven "marcas" que garantizan que todo el software que cubren está distribuido bajo licencias que aseguran ciertas condiciones "razonables" y simples de entender. Un ejemplo notable de esto es la marca *Open Source*[6].

A continuación se exponen con cierto detalle algunas de las licencias bajo las que distribuye habitualmente el software libre.

4.1. BSD (Berkeley Software Distribution)

Cuando las primeras versiones de Unix fueron distribuidas por ATT a algunas Universidades, el lugar donde más desarrollos se hicieron sobre ellas fue la Universidad de California en Berkeley. Estos desarrollos, compuestos íntegramente por código desarrollado por los investigadores de la Universidad o por sus colaboradores se distribuyeron como "Berkeley Software Distributions" (BSD). Con el tiempo, BSD se convirtió en una de las dos ramas más extendidas de la familia Unix, incluyendo muchas versiones propietarias (como SunOS, Ultrix, etc.).

Las características de la licencia bajo la que se distribuyó este software presenta las siguientes características:

- Se permite la redistribución, uso y modificación del software.
- Las distribuciones deben incluir copias literales de la licencia, anuncio de copyright y una "negación de responsabilidad" (*disclaimer*).
- Debe incluirse reconocimiento del origen del software (la Universidad de California) en cualquier anuncio.

En otras palabras, los redistribuidores pueden hacer casi cualquier cosa con el software, incluyendo usarlo para productos propietarios. Los autores sólo quieren que su trabajo sea reconocido. En cierto sentido, esta restricción asegura un cierto grado de posibilidad de comercialización. Es importante darse cuenta que este tipo de licencia no incluye ninguna restricción orientada a garantizar que los trabajos derivados sigan siendo libres. De hecho, ya se ha mencionado anteriormente cómo muchos sistemas operativos derivados de versiones de BSD han sido distribuidos como software no libre. Puede argumentarse que esta licencia asegura "software completamente libre", en el sentido que el usuario tiene libertad ilimitada con respecto al software, y puede decidir incluso redistribuirlo como no libre. Otras opiniones están más orientadas a destacar que este tipo de licencia no contribuye al desarrollo de más software libre, incluso cuando es básicamente la versión libre original redistribuida como un producto propietario.

[6] `http://www.opensource.org`

4.2. GPL (GNU Public License)

El proyecto GNU fue anunciado por Richard Stallman en 1983, y es uno de los orígenes principales del movimiento de software libre, tal y como lo conocemos hoy día. Con él se pretendía realizar un sistema operativo completo, con todas las aplicaciones habituales en la época, donde todos sus componentes fuesen software libre. La mayor parte del software de GNU (y muchos otros programas, como por ejemplo el kernel Linux) están protegidos por la "GNU Public License" (GPL). Las principales características de esta licencia son las siguientes:

- Permite la redistribución binaria.
- Permite la redistribución fuente (obliga a ella en caso de redistribución binaria).
- Permite las modificaciones sin restricciones.
- Integración completa sólo con software cubierto por la GPL.

La idea básica de esta licencia es utilizar la legislación actual de *copyright* para obligar a los que reciben un programa a cumplir ciertas condiciones. Esas condiciones están cuidadosamente diseñadas para conseguir que ese programa, o cualquier trabajo derivado de él, sigan siendo software libre. Desde este punto de vista, la GPL no sólo permite distribuir software libre, sino que se esfuerza para asegurarse de que a partir de él nunca se derivará software propietario.

4.3. LGPL (Library GPL)

La GPL resultó ser demasiado estricta para algunos tipos de software. Por ejemplo, si una biblioteca está protegida por la GPL, no puede utilizarse para enlazarla con código propietario. Esto impide, por ejemplo, el uso de un compilador con bibliotecas protegidas por GPL para crear software propietario. En su momento, Richard Stallman decidió que para asegurarse el éxito de *GCC* (el compilador de C de GNU) necesitaba cubrir sus bibliotecas con otra licencia. Así nació la licencia GNU para bibliotecas (LGPL). Sus características principales son las siguientes:

- Está pensada para permitir el uso de bibliotecas libres con software propietario (por ejemplo, en el caso de un compilador).
- Se comporta como la GPL cuando se redistribuye la biblioteca como tal.
- Permite la integración con cualquier otro software. En este caso, no hay prácticamente limitaciones.

Por lo tanto, la biblioteca queda protegida prácticamente como quedaría si estuviera protegida por la GPL, pero permitiendo su uso para el "mezclado" (enlazado o *linking*) con software propietario.

4.4. Otras licencias de software libre

Aunque la mayor parte del software libre disponible hoy día utiliza alguna de las licencias anteriores, o ligeras modificaciones de las mismas, hay muchas más. De hecho, potencialmente, cada autor puede escribir su propia licencia para especificar las condiciones exactas que le interesen. Por ejemplo, los siguientes son algunos otros ejemplos:

- **Artistic** es una de las licencias bajo las que se distribuye Perl, y es hasta cierto punto similar a la de BSD.
- **Aladdin** es la licencia que protege a Aladdin Ghostscript. En caso de redistribución con ánimo de lucro, las condiciones son similares a las del software propietario, pero en caso contrario, es muy similar a la GPL.
- **MPL** (Mozilla Public License) es la licencia bajo la que se distribuye el nuevo navegador libre de Netscape. Incluye ciertos privilegios para el "primer autor", y es por lo demás similar a la GPL.

Alguna de estas licencias, como la MPL o la de Aladdin están diseñadas intentando maximizar los derechos del autor del software, sin violar las condiciones que hacen de ellas "licencias de software libre". De hecho, no son más que un exponente de la etapa de exploración de nuevos modelos de negocio que se está viviendo desde hace unos años en el mundo del software libre.

Por último, es importante hacer notar que alguna de las licencias analizadas (y en particular la GPL) serían probablemente innecesarias si no hubiese legislación que permitiese la restricción de los derechos de uso, copia y distribución del software.

5. Conclusiones

En esta ponencia hemos tratado de exponer un nuevo modelo de distribución de software que se está abriendo paso en el mercado informático: el llamado "software libre".

Este modelo pone en tela de juicio que la aplicación de las leyes del copyright (pensadas fundamentalmente para proteger los derechos de los autores y los editores de obras literarias) a la distribución de software, sirva para obtener las ventajas sociales que llevaron a su promulgación: garantizar la producción de obras de calidad y en cantidad suficiente para hacer progresar a la sociedad.

Afirmamos que dichas ventajas no se obtienen en el caso de la producción de software y aventuramos, que de la misma forma que en el siglo pasado la sociedad detectó la necesidad de legislar para garantizar a la disponibilidad de libros de calidad, de nuevo y con los mismos objetivos (garantizar la calidad y disponibilidad de software), la sociedad deberá generar leyes diferentes. Sin duda, esa nueva legislación reflejará la experiencias obtenidas con los distintos modelos actuales de distribución de software libre.

Como parte fundamental de dichos modelos, se han expuesto también las características principales de alguna de las licencias de software libre más habituales (entre ellas, la GPL, la LGPL y la BSD). Cada una de ellas genera distintos modelos de negocio, y restringe distintos aspectos de lo que puede hacer o no el usuario con el programa que recibe. Pero en ningún caso prohiben las libertades básicas de uso, copia y redistribución (incluido el código fuente).

Referencias en Internet

GNU Project: `http://www.gnu.org`. Incluye versiones literales de las licencias GPL y LGPL, y comentarios sobre las mismas. Incluye también varios artículos interesantes.

Open Source: `http://www.opensource.org`. Incluye la definición exacta del término "open source".

Mozilla Project: `http://www.mozilla.org`. Incluye una versión literal de la licencia Mozilla, y comentarios sobre la misma.

Grupo SoBre: `http://www.gsyc.inf.uc3m.es/sobre/`. Información general sobre software libre, y lista de correo electrónico sobre temas relacionados con él.

Referencias

Copyright and Digital Libraries, Pamela Samuelson, Communications of the ACM, Vol 38, num 4, April 1995, pp. 15-21 y 110. Orígenes históricos de la legislación sobre copyright.