

Discusiones sobre Aspectos Legales en torno a Software Libre y Formatos Digitales

Revisión: 1.10

Jens Hardings, Alejandro Fuentes
(jens@hardings.cl, alejandro.fuentes@csol.cl)

8 de julio de 2003

Resumen

El centro de software libre se dedica a estudiar y fomentar el fenómeno de Software Libre (conocido también como “Open Source Software” y/o “Free Software”). Este fenómeno abarca múltiples disciplinas, generando múltiples discusiones e incógnitas en áreas tan variadas como la sociología, ingeniería de software, política y por supuesto los aspectos legales en los cuales nos concentramos en este documento. Las disposiciones legales pueden tener mucha influencia en el desarrollo del software libre, y el sistema legal puede verse afectado por nuevos desafíos creados por cambios en la tecnología y sobre todo en los usos de esas tecnologías. Esta es una invitación a participar de la discusión de estos temas y así crear conciencia de la importancia que tienen y tendrán en nuestra sociedad. La invitación está abierta a todos los interesados, independiente de su posición frente a las ideas planteadas en este documento.

1. Introducción

El fenómeno de software libre¹ abre una serie de inquietudes en varias áreas distintas, en particular en el ámbito legal lo relativo propiedad intelectual. De hecho, Moglen describe la posibilidad de que el software libre pueda finalmente

¹Para un pequeño resumen sobre lo que es el software libre, ver (Hardings and Fuentes, 2003).

marcar el derrumbe del sistema de propiedad intelectual tal como se conoce hoy en día (Moglen, 1999). En esta sección enunciaremos algunos de los temas interesantes que surgen directa o indirectamente de nuestra preocupación por el software libre y esperamos que puedan asimismo despertar el interés de expertos en áreas legales como propiedad intelectual para trabajar en conjunto.

Los temas a tratar incluyen derecho de autor, tanto por el lado de licencias de software como abusos por parte de quienes gozan de la protección y patentes en áreas como software, donde más que incentivar el desarrollo y la innovación son un freno para la industria.

A pesar de las constantes críticas al esquema seguido por EEUU, se sigue imponiendo este esquema a países en todo el mundo a cambio de beneficios económicos. Esto ya fue observado en 1994 (Barlow, 1994) y sigue ocurriendo hasta el día de hoy, ya sea a través de tratados de libre comercio², el Acuerdo sobre los Aspectos de los Derechos de Propiedad Intelectual relacionados con el Comercio / Trade Related Intellectual Property (ADPIC / TRIPS)³, así como otros acuerdos internacionales que contienen elementos que intentan forzar esta política respecto de propiedad intelectual en el resto del mundo. Por lo general, estos acuerdos se realizan sin el conocimiento y menos la participación de los principales afectados. Sin embargo, ya comienzan a escucharse voces que apuntan en la dirección contraria, argumentando que no es necesario contar con una protección a la propiedad intelectual para alcanzar avances en varios ámbitos. Un ejemplo de esto es una carta⁴ dirigida a la WIPO (World Intellectual Property Organization), en la cual se hace notar la importancia de tomar en cuenta las experiencias exitosas de desarrollos de bienes públicos en forma colaborativa. Se enumeran estos ejemplos (sin traducir):

1. The IETF and Open Network Protocols.
2. Development of Free and Open Software.
3. The World Wide Web.
4. The Human Genome Project (HGP).
5. The SNP Consortium.
6. Open Academic and Scientific Journals.

² http://www.direcon.cl/html/acuerdos_internacionales/estadosunidos_12.php

³ http://www.wto.org/english/tratop_e/trips_e/trips_e.htm

⁴ <http://www.cptech.org/ip/wipo/kamil-idris-7july2003.pdf>

7. The Global Positioning System.

2. Licencias de software

El tema de las licencias de software ha sido ampliamente ignorado por parte de las personas que las aceptan. Sin embargo, estas licencias pueden ser consideradas contratos según las circunstancias, y en la mayoría de los casos los usuarios ignoran sus contenidos, los cuales no aceptarían si conocieran las potenciales consecuencias.

Destacamos los siguientes aspectos respecto de licencias:

- Nula responsabilidad del desarrollador. Independiente de que se trate de software propietario o libre, no existe ningún programa usado masivamente en el cual el desarrollador no haya excluido cualquier responsabilidad en caso de falla o hasta en casos de clara negligencia. Si bien exigiríamos responsabilidades de los desarrolladores de un objeto como un automóvil si presenta fallas atribuibles a errores en el desarrollo, a nadie parece importarle que los programas que se usan a diario tengan errores por los cuales no se puede responsabilizar a nadie, y que manejan aspectos críticos de nuestra vida. ¿Cuál es (o debiera ser) el límite legal hasta el cual los desarrolladores de software, ya sea software propietario o libre, pueden desligarse de su responsabilidad?
- Una incógnita aún no resuelta es si la GPL (GNU General Public License, ver Hardings and Fuentes (2003)) tendrá el efecto deseado por sus creadores en caso de que sea puesta a prueba en una corte. Está la pregunta de los límites que se pueden exigir en una licencia, los cuales ciertamente son bastante más reducidos que las exigencias que se pueden hacer cuando existe un contrato de por medio (Malcolm, 2003).
- Se pueden considerar contrato los “End User License Agreements”? ¿Cuáles son los límites de los acuerdos en tal caso y qué consecuencias legales tienen esos acuerdos con otras obligaciones de los usuarios? Por ejemplo, es posible usar un software que le da el derecho a la empresa desarrolladora de ingresar al sistema y usar datos existentes allí? ¿Qué pasa si por ejemplo un médico acepta ese tipo de acuerdos y deja datos de pacientes en ese sistema, los cuales quedarán a disposición del desarrollador? Este mismo

caso se puede extrapolar a cualquier profesión donde exista una obligación de secreto profesional, por ejemplo en el caso de abogados.

Este tipo de licencias son las que aceptan usuarios de sistemas tan comunes como *Microsoft Windows XP*, sin percatarse de las desastrosas consecuencias que puede tener el presionar un botón si resultara que estas licencias son imponibles. Ese mismo sistema se conecta sin intervención del usuario y por lo general sin su conocimiento, en 16 circunstancias distintas a servidores de Microsoft, entregando información a dicha empresa⁵. Los problemas de privacidad que surgen con este tipo de prácticas son evidentes, sin embargo no se discuten con la seriedad necesaria.

3. Derechos de Autor

Los tratados internacionales de la OMPI (Organización Mundial de la Propiedad Intelectual) sobre Interpretación o Ejecución y Fonogramas (WPPT) y sobre Derecho de Autor (WCT), que en EEUU se han implementado bajo una ley conocida como “Digital Millenium Copyright Act” (DMCA) y en Europa como “European Copyright Directive” (EUCD), dan lugar a abusos por parte de beneficiarios del derecho de autor. Desde la entrada en vigencia, estas leyes han sido usadas exclusivamente para evitar competencia y mantener situaciones de beneficio que son ajenas al espíritu de las leyes de derecho de autor. Es importante discutir cómo se pueden implementar en Chile estas directivas sin cometer los mismos errores, o bien si eso no es posible, generar la crítica necesaria para modificar los tratados (incluido también el tratado de libre comercio entre Chile y EEUU que trata de llevar a Chile hacia el mismo estado que EEUU en cuanto a sobreprotección de ciertos derechos).

Ejemplos concretos que afectan el desarrollo de software en general y el de software libre en particular:

- La controvertida “Church of Scientology” usa la DMCA para evitar la publicación versiones electrónicas de documentos en sitios web que los usan para criticar dicha institución.
- La DVD Copy Control Association usa la DMCA para evitar el desarrollo de un software que permita ver en el sistema operativo Linux DVDs ad-

⁵ <http://www.hevanet.com/peace/microsoft.htm>

quiridos legítimamente, evitando así que los compradores hagan uso de su legítimo derecho a ver la película.

- Adobe usa la DMCA para encarcelar a Dmitri Sklyarov, un joven ruso que mostraba las falencias de seguridad de su software de manejo de eBooks. La empresa en la cual trabajaba Sklyarov se dedicaba a crear software que permite a personas discapacitadas visualmente escuchar el contenido de un eBook, para lo cual es necesario acceder al contenido encriptado del mismo.
- Lexmark usa la DMCA para argumentar que todos los fabricantes de cartuchos de tinta de repuesto de tinta para sus impresoras producirían artefactos ilegales. El razonamiento está en que el driver (programa que interactúa con la impresora) que está instalado en el computador interactúa con la impresora de forma secreta, y la única forma de saber cómo funciona esta comunicación es copiando ilegalmente los métodos protegidos tecnológicamente dentro de los cartuchos de Lexmark. De esta forma, un cliente de Lexmark no puede legalmente comprar productos que esta empresa no desee, y deberá aceptar sus precios.

Es preocupante que se ponga en las manos del autor, creador o desarrollador exclusivamente los derechos que un usuario tiene o no sobre una obra, sin garantizar que las excepciones al derecho de autor se sigan cumpliendo. Con las tecnologías de Digital Restrictions Management (también llamadas Digital Rights Management o simplemente DRM), es el creador el que selecciona qué derechos entregarle a determinada persona. Esto destruye completamente las limitaciones que tradicionalmente han tenido los derechos de autor, así como por ejemplo el simple hecho de realizar una copia de respaldo, trasladar la obra a otro medio para su uso o simplemente usar la obra de una forma distinta a la predefinida por el autor⁶. Si bien en teoría estos derechos siguen existiendo, en la práctica no se podrá hacer uso de ellos sin violar la ley.

⁶Es perfectamente posible definir por ejemplo, que el usuario no pueda leer un libro de atrás para adelante o que no pueda ver el final de la novela si no ha pasado por todas las páginas anteriores al menos un determinado período de tiempo. Suena ridículo, pero por ejemplo los DVDs tienen un sistema que evita al usuario final adelantar la película y existen DVDs (si bien es por error) que tienen ese sistema activado durante todo el contenido de la película.

4. DRM

El tema de las restricciones manejadas digitalmente abre muchas interrogantes, por ejemplo si el usuario que compra un computador debiera tener control sobre éste, o si el control lo ejercerá la empresa que desarrolla el computador y/o el software para éste. Si bien lo racional es que el usuario debiera poder elegir tener el control absoluto sobre el sistema, muchos indicios apuntan a que no será así en la práctica. El argumento de quienes favorecen la opción de que el usuario no tenga completo control es que esto aumentaría la seguridad, pero en realidad simplemente se provee una ventaja comercial para quienes controlan los computadores (Anderson, 2002).

Como ya se menciona en Hardings and Fuentes (2003), es importante notar que, con la introducción de medios digitales, existen varios cambios en cuanto al potencial uso y nuevas oportunidades para obras intelectuales y artísticas. Por un lado, es posible reproducir las obras a un costo muy cercano a cero. Pero por otro, se abre la posibilidad de controlar, por medio de medidas tecnológicas, qué uso se le da a las obras. Es extraño por lo tanto que las iniciativas legales hayan ignorado estos últimos cambios y se hayan concentrado únicamente en los primeros. Tampoco se garantiza que el material protegido vaya finalmente a estar disponible en el dominio público sin restricciones al cumplirse el plazo establecido por ley para la protección.

5. Patentes de Software

Si bien es cierto que las patentes en general tienden a incentivar el desarrollo tecnológico por la vía de permitir al que invierte gozar de beneficios monopólicos por un tiempo determinado, existe común acuerdo de que en el caso de software las patentes tienden a desincentivar el desarrollo y estancan la industria. Tanto así, que incluso los más acérrimos defensores de hasta los más extremos mecanismos para proteger la propiedad intelectual, la Business Software Alliance (BSA) ha expresado que está contra el principio seguido en EEUU⁷, donde se permiten pa-

⁷Extracto de un artículo de la revista InfoWorld: [http://www.infoworld.com/article/03/06/18/EU-headed-for-limited-software-patent-law-\(IDGNS\)_1.html](http://www.infoworld.com/article/03/06/18/EU-headed-for-limited-software-patent-law-(IDGNS)_1.html) : “We support the move towards a harmonized European approach to software patents,” said Francisco Mingorance, director of public policy, Europe for the BSA, adding that the BSA doesn’t want a U.S.-style system. “We agree that technical effects should be a condition of patentability,” he said. Por otro lado, hay quienes aseguran que la condición de *technical effects* no impide patentes de software, solamente obliga a

tentes no sólo de software, sino que incluso de ideas y procesos de negocios. Más información se puede obtener por ejemplo en <http://lpf.ai.mit.edu/Patents/>.

La razón de estas posiciones contrarias a permitir patentes de software se debe a que existe evidencia de que, contrariamente a lo argumentado por defensores de la liberalización de patentes de software, la industria no se beneficia, como se pudo comprobar en EEUU en la década de los '80 (Bessen and Maskin, 2000), ver también la argumentación económica de Boldrin and Levine (2001). En general, no solamente en la industria del software, existe la percepción de que la mayor utilidad de patentes es para bloquear competencia y para defenderse frente a demandas de terceros por patentes (Hall, 2003).

Entre los argumentos en contra de las patentes de software se cuentan:

- La cantidad de patentes necesarias para producir un solo producto pueden ser del orden de miles. En cambio, en otras áreas de desarrollo tecnológico, donde se justifica la práctica de patentabilidad, suele ser bastante limitada la cantidad de patentes por producto comercializable. Es por ejemplo el caso de la industria farmacéutica, donde para una droga o proceso generalmente es aplicable una sola patente.
- La inversión requerida para crear un software es mínima comparada con la investigación en el campo de la medicina, mecánica y otras áreas. Para tomar nuevamente el ejemplo de la industria farmacéutica, desarrollar un fármaco generalmente toma años de trabajo, requiere del uso de equipamiento costoso y muchos estudios igualmente costosos. Por otro lado, la imitación en campos como la industria farmacéutica es trivial comparado con la industria del software, en el cual la imitación implica el desarrollo completo del sistema (por estar protegido por derechos de autor y por la no disponibilidad del código fuente en general) (Hall, 2003).

En el caso de las patentes de software, detectar si un producto contiene código que está sujeto a patentes por lo general tiene un costo muchas veces superior a la creación del código afectado. Esto se debe por un lado a la gran cantidad de patentes que existen en los países donde son aceptables las patentes de software, además del lenguaje en el que están descritos los procesos protegidos, los cuales se alejan mucho del lenguaje tradicionalmente usado por los programadores y requiere largas interacciones entre abogados y programadores para que ambas partes comprendan qué es realmente lo

usar más creatividad en describirlos.

que está protegido por una patente dada. Por lo tanto, la práctica de usar una base de datos de patentes como medio de información dentro del proceso de desarrollo de un producto relativo a la informática no sólo no existe, sino que no tiene sentido de ser instaurada por la ineficiencia inherente en el sistema.

- La publicación de la patente debe incluir una descripción que permita a alguien entendido en el área respectiva reproducir el proceso o invención. Sin embargo, al no incluirse el código fuente (ver Hardings and Fuentes (2003)), para reproducir un procedimiento es necesario volver a desarrollar el código fuente, con lo cual el aporte de una patente es en el mejor caso dudoso. Por ejemplo, para el caso de invenciones es necesario proveer planos y diagramas que explican claramente cómo funciona un elemento dentro del sistema que se pretende patentar. En el caso de software, solamente se agregan descripciones vagas de qué es lo que se pretende patentar, y es necesario desarrollar el producto para tener un programa concreto y usable.

También es importante notar que no es necesario que la innovación que se pretende patentar realmente exista, sino que basta que se haya descrito. En el caso de patentes industriales que se refieren a procesos físicos esto no es un problema, puesto que la descripción es lo suficientemente detallada para implementar el invento, de lo contrario la patente no tiene valor porque el proceso no es aplicable industrialmente. Pero en la industria del software, donde para una patente tampoco se requiere una implementación funcional, el texto de la patente puede ser lo suficientemente vago como para ser aplicado a otras ideas o implementaciones que las propuestas por los inventores. En tal caso, el legítimo inventor va a quedar excluido de utilizar y recibir los beneficios económicos que le corresponderían por su invento, ya que la patente ha sido entregada a quien no ha implementado la idea.

- Para PYMES es muy difícil crear una cartera de patentes con los cuales competir en una industria dominada por unos pocos protagonistas que poseen derechos sobre miles de patentes cada uno. Contrariamente a la intención de un sistema de patentes, que pretende proteger al pequeño inventor o innovador, se logra aumentar la barrera de entrada a la industria del software de forma significativa y artificial.

Incluso el supuestamente simple hecho de litigar, para evitar el pago por una demanda injustificada de licencias por patentes, puede hacer peligrar la

existencia a empresas que no tienen los recursos necesarios para esto, por el alto costo que conlleva.

- Donald Knuth preguntó una vez: “¿Qué pasaría si abogados individuales pudiesen patentar sus métodos de defensa, o si cortes supremas patentaran sus precedentes?”⁸. Con patentes de software pasa algo análogo, ya que los algoritmos son elementos fundamentales para programadores y patentes sobre ellos simplemente dificulta o incluso impide su trabajo. El desarrollo de la industria del software es secuencial (Bessen and Maskin, 2000).
- El tiempo entre la solicitud de una patente y la aprobación en la práctica es de entre 5 a 10 años. En una industria como la del software, en donde cada 2 a 3 años surgen nuevas generaciones de productos, claramente el sistema de patentes no calza. Un producto puede haber estado 7 años en el mercado cuando de un momento a otro se convierte en un artefacto ilegal mientras no se negocie una licencia con algún ente que obtuvo la patente respectiva. Más aún, al momento de ser concedida la patente, el invento muy probablemente ya ha pasado a la categoría de trivialidad y es usado en una multitud de programas y productos. Esto hace que las patentes no sean una herramienta efectiva para el que la solicita, tampoco para la industria y menos para la sociedad, que primero puede acceder a tecnología masivamente y después tiene que financiar el desarrollo.
- Finalmente, los ejemplos de patentes de software concedidas⁹ hablan por sí solos. Queda en evidencia que, a pesar de tomar un tiempo demasiado extenso en revisar los requisitos, la revisión por parte de las oficinas de patentes no es lo prolija que debiera, o bien los revisores no cuentan con la experiencia ni conocimientos necesarios para discernir sobre la trivialidad de una invención.

6. Conclusiones

Se han presentado algunos de los aspectos legales que surgen por la existencia y masificación de medios digitales, así como por el fenómeno del software libre.

⁸ <http://lpf.ai.mit.edu/Patents/knuth-to-pto.txt>

⁹ Ver por ejemplo <http://swpat.ffii.org/patents/index.en.html>
<http://www.base.com/software-patents/examples.html>
<http://www.freepatents.org/examples/>.

Si bien los puntos son muy interesantes de discutir y analizar, existen pocas iniciativas, sobre todo en países en vías de desarrollo donde pueden tener una mayor importancia que en países desarrollados.

Es claro que no todos los lectores de este documento van a estar de acuerdo con las ideas expresadas, la invitación a participar de la discusión se extiende sobre todo a ellos, para así generar discusión real y no una declaración de acuerdos. Los temas son muy complejos y por lo mismo es tiempo de tratarlos con la seriedad que merecen, más que dejar el tema en manos de gente especializada en el tema, que por lo mismo tiene una parcialidad bastante marcada y no es capaz de ver el tema desde otras perspectivas tanto o más importantes.

Referencias

- Anderson, R. (2002). Security in Open versus Closed Systems - The Dance of Boltzmann, Coase and Moore. In *Papers of Open Source Software: Economics, Law and Policy*, Toulouse, France. Institut d'Economie Industrielle, Université des Sciences Sociales.
- Barlow, J. P. (1994). The Economy of Ideas. http://www.wired.com/wired/archive/2.03/economy.ideas_pr.html.
- Bessen, J. and Maskin, E. (2000). Sequential Innovation, Patents, and Imitation. <http://www.researchoninnovation.org/patent.pdf>.
- Boldrin, M. and Levine, D. K. (2001). Perfectly Competitive Innovation. Available online at <http://levine.sscnet.ucla.edu/papers/pci23.pdf>.
- Hall, B. H. (2003). Business Method Patents, Innovation and Policy. Competition Policy Center. Paper CPC03-039.
- Hardings, J. and Fuentes, A. (2003). Introducción al Software Libre. <http://www.hardings.cl/publications/hardings2003intro.pdf>.
- Malcolm, J. (2003). Problems in Open Source Licensing. In *Australian Linux conference*.
- Moglen, E. (1999). Anarchism Triumphant: Free Software and the Death of Copyright. *First Monday*, 4(8).