



Fundación
Vía Libre



El dilema del Copyright en el campo del Software

Por Beatriz Busaniche



Este trabajo se distribuye bajo una
licencia Creative Commons, Atribución,
Compartir Obras Derivadas Igual de Argentina.

EL DILEMA DEL COPYRIGHT EN EL CAMPO DEL SOFTWARE

POR BEATRIZ BUSANICHE¹

Trabajo final para el Seminario de Derecho de Autor. Maestría en Propiedad Intelectual. Flacso Argentina.

The very object of publishing a book on science or the useful arts is to communicate to the world the useful knowledge which it contains².

Los expertos de la Organización Mundial de la Propiedad Intelectual entienden que un programa de ordenador o “software” es *“un conjunto de instrucciones expresadas mediante palabras, códigos, planes o en cualquier otra forma que, al ser incorporadas en un dispositivo de lectura automatizada, es capaz de hacer que un “ordenador” -un aparato electrónico o similar capaz de elaborar informaciones- ejecute determinada tarea u obtenga determinado resultado”³.*

La cuestión sobre si los programas de ordenador se pueden ubicar entre los bienes intelectuales amparables bajo el derecho de autor despertó no pocas controversias y debates. Desde la década del 60 y con mayor intensidad en los 70s, la cuestión del marco jurídico para contener los programas de ordenador generó intensas discusiones de las que trataremos de dar cuenta en las próximas páginas.

Este trabajo pretende rastrear las principales controversias alrededor de la cobertura legal de Copyright de los programas de ordenador, dar cuenta de las decisiones tomadas y sus implementaciones legales, así como las consecuencias de estas decisiones y algunas estrategias jurídicas desarrolladas a partir de la inclusión de los programas de computadora como obras amparadas por el sistema de Copyright. En este marco, abordaremos en particular la construcción jurídica del Software Libre y cómo su estrategia legal intenta solucionar los problemas surgidos con la inclusión del Software bajo la normativa del Copyright.

¹Beatriz Busaniche es Licenciada en Comunicación Social. Este trabajo se distribuye bajo una licencia Creative Commons, Atribución, Compartir Obras Derivadas Igual de Argentina. Para más información visite http://creativecommons.org/licenses/by-sa/3.0/deed.es_AR

²Baker v. Selden, 101 U.S. 99 Volume 101 (1879) U.S. Supreme Court <http://supreme.justia.com/us/101/99/case.html>

³Definición incluida en el artículo I, del Proyecto de Disposiciones Tipo para Leyes en Materia de Derecho de autor. Documento OMPI CE/MPC/I/2-II del 11 de Agosto de 1989. Publicado en el Sitio de la OMPI en http://www.wipo.int/mdocsarchives/CE_MPC_I_1989/CE_MPC_I_2_II_S.pdf (visitado el 8 de agosto de 2010).

La definición elaborada por los expertos de la OMPI en relación a los programas de ordenador no acierta en la especificidad real de lo que un programa de computación es, y por lo tanto, complica la definición de la obra como tal. Un uso preciso del término debería contemplar la definición de “programa” como código o texto fuente y código de máquina (o código objeto o ejecutable).

Fuente y ejecutable son similares en tanto ambos son un set de instrucciones detalladas para darle órdenes a una computadora, para que pueda realizar sus funciones o una tarea particular. Sin embargo, el código fuente es el texto escrito en un lenguaje de programación que es comprensible para una persona mientras que el código de máquina se limita al código que da las instrucciones a la computadora. El código de máquina no es legible para las personas⁴.

Los programas en código fuente, escritos por programadores y comprensibles para otras personas que dominan la técnica, contienen una intención comunicativa subyacente y pueden existir, incluso, independientemente de la existencia de una máquina⁵. El texto fuente de los programas tiene una capacidad expresiva, es el vehículo idóneo para comunicar algoritmos, soluciones a problemas, de la misma manera que un músico comunica a sus pares la composición mediante la escritura de partituras. En el siguiente fragmento escrito en lenguaje C se puede apreciar la intención comunicativa de quien lo redactó⁶:

```
/* Esta función imprime la raíz cuadrada de su argumento */
static void printsqrt(float x) {
if (x < 0) /* la raíz de un numero negativo es imaginaria */
printf("El numero es menor que cero!\n");
else /* el numero es positivo, todo bien */
printf("%f\n", sqrt(x));
}
```

La solución de un problema puede tener múltiples expresiones algorítmicas, así como cada algoritmo puede tener múltiples expresiones en código que en ocasiones dependen incluso de decisiones estéticas. Los programas en *texto fuente* incluyen, además de las instrucciones, una serie de *comentarios* que permiten a otros programadores entender el por qué de determinadas decisiones, continuar la tarea, modificar los programas y aprender de lo hecho por el programador original. El siguiente paso del proceso consiste en compilar, mediante un procedimiento automatizado, el texto fuente, para obtener un código objeto que constituye en sí mismo un estricto imperativo,

⁴Samuelson, Pamela “CONTU Revisited: The Case against Copyright Protection for Computer Programs in Machine-Readable Form”. Duke Law Journal, Vol. 1984, No. 4 (Sep., 1984), pp. 663-769 Duke University School of Law

⁵El matemático Donald Knuth diseñó el lenguaje MIX para expresar algoritmos en su serie “The Art of Computer Programming”. No existe ninguna máquina que ejecute ese lenguaje. Véase el sitio personal de Donald Knuth en <http://www-cs-faculty.stanford.edu/~knuth/index.html>

⁶Heinz, Federico, (2010) “Software vs. Copyright” disponible en línea en <http://www.vialibre.org.ar/2010/07/23/software-vs-copyright/>

las instrucciones para la máquina por las cuales ésta ejecuta las tareas previstas en el programa. Ese código objeto, como ya dijimos, no es comprensible por el ser humano.

LA REGLA DE LA DUDA⁷

La separación entre el negocio de la venta de equipos informáticos y los programas de cómputo se produjo a lo largo de la década del 60. A medida que las computadoras se hacían más avanzadas y con modelos más accesibles, los negocios vinculados específicamente al software generaron y ampliaron sus mercados. Los cambios en la arquitectura de las computadoras fomentaron esta transformación, que arrojó como primer resultado la expansión del mercado del software como tal. La separación de las aplicaciones de software de las máquinas de IBM en 1970, como consecuencia de presiones *antitrust*, acentuó esta tendencia⁸.

Cuando en la década del 60, los programadores comenzaron a acercarse a la Oficina de Copyright de los Estados Unidos, se plantearon las primeras controversias sobre la posibilidad o no de amparar los programas bajo esa forma legal. La respuesta que encontró la US Copyright Office fue lo que denominó “*la regla de la duda*”. Los funcionarios de la oficina tomaron debida nota de la diferencia entre el texto fuente de los programas y los ejecutables, en particular de su carácter utilitario y la imposibilidad de ser leídos por las personas. La Oficina aceptó los registros y comenzó a emitir certificados en los que expresamente indicaba que “la oficina de copyright tiene *dudas* sobre la posibilidad de otorgar copyright a las versiones de máquina de los programas pero decide dejar la resolución legal a los tribunales en caso de litigio”⁹. En paralelo a la regla de la duda, la Oficina exigía a los registrantes el depósito de una copia del código fuente de los programas, que también podía ser inspeccionada públicamente. Entre los años 1966 y 1978, la Oficina de Copyright expidió aproximadamente 1200 certificados de registro de software bajo la regla de la duda y con la obligación de depositar el texto fuente.

Las opiniones en relación al marco jurídico del software estuvieron divididas durante todo ese tiempo. Los autoristas temían que la inclusión del software, que, según consideraban, no tenía carácter literario o artístico, debilitara la construcción jurídica del derecho de autor. Los estudios publicados en los primeros años de la década del 70, en particular, las conclusiones del estudio del alemán Eugen Ulmer encargados por la UNESCO y la OMPI, concluyeron que la protección del software por derecho de autor era posible. El trabajo de Ulmer reconoció que el programa de ordenador constituye una obra del intelecto resultante de un proceso creativo similar al que demanda una obra literaria o un libro cinematográfico, que, cuando es original en su composición y en su expresión, está protegido por el derecho de autor¹⁰. Siguiendo la argumentación de Ulmer se puede decir que el programa de ordenador es una obra, que las facultades en tanto derechos patrimoniales que otorga el derecho de autor son aplicables al campo de los programas de ordenador, y que el destino y la forma de expresión de la obra son criterios ajenos al reconocimiento de la protección de derecho de autor (Lipszyc, 2006:109).

⁷En inglés “rule of doubt”

⁸Lemley, Menell, Merges, Samuelson. “Software and Internet Law” (2006) Third Edition. Aspen Publishers. Pag. 34.

⁹Ibidem.

¹⁰Lipszyc, Delia “Derecho de autor y derechos conexos”. Ediciones UNESCO, CERLALC ZAVALIA. Reedi- ción 2006. pag. 107

Debido a una serie de controversias legales en cuanto a nuevas tecnologías, el Congreso de los EEUU decidió, en 1974, el establecimiento de la Commission on New Technological Uses of Copyrighted Works (de ahora en más CONTU) para realizar investigaciones y proponer recomendaciones al respecto.

CONTU

La CONTU fue establecida por el Congreso de los EEUU y funcionó entre los años 1975 y 1978 para tratar de determinar la forma en la cual la ley de Copyright debería abordar los aspectos relacionados a las computadoras y las máquinas copiatoras. El congreso trabajaba en esos años en una reforma integral de la ley de Copyright para afrontar algunos desafíos vinculados a las nuevas tecnologías, en particular, si los derechos exclusivos de los titulares debían extenderse a las copias electrónicas de las obras o si estos derechos se limitaban a las copias en soportes tangibles como libros o discos, y a resolver el problema planteado por la existencia de fotocopiadoras. El tema del copyright sobre los programas de ordenador, que aparecía en el mandato de la CONTU como un tema menor fue luego el que cobró mayor trascendencia por la magnitud del mismo y de las controversias vinculadas al problema (Samuelson, 1984:695).

La Comisión emitió su Reporte Final el 31 de Julio de 1978¹¹. Paradójicamente, los temas centrales de CONTU en relación a las fotocopias y las copias en soporte electrónico no generaron ninguna controversia al interior de la Comisión. Sin embargo, los temas vinculados al copyright para el software fueron los que cobraron más complejidad a lo largo del trabajo. Por unanimidad se estableció que los programas de ordenador en versión objeto merecían algún tipo de protección, pero varios miembros de la comisión dejaron asentadas sus dudas sobre si el copyright sería el medio idóneo para tal fin¹².

Vale mencionar que la composición de la comisión estaba relacionada principalmente con el tópico primario de la misma: analizar los temas vinculados a la fotocopia y la copia electrónica. Por esta razón, los miembros no tenían experiencia especial en el tema vinculado al copyright y los programas de ordenador. No había científicos de computación, ni representantes de la industria del software y el hardware, ni usuarios ni académicos vinculados a sistemas de software complejos en la comisión. La comisión designó a tres de sus miembros para trabajar específicamente en este tema¹³.

¹¹CONTU Final Report. <http://digital-law-online.info/CONTU/PDF/index.html> (visitado el 8 de agosto de 2010).

¹²"The Commission is unanimous in its belief that computer programs are entitled to legal protection. But the unanimity has not extended to the precise form that protection should take" Informe Final CONTU disponible en <http://digital-law-online.info/CONTU/contu5.html> Los miembros que expresaron dudas fueron Nimmer, Hersey, y Karpatkin. Véase también <http://digital-law-online.info/CONTU/contu14.html#sec2.1.9>. Dissent of Commissioner Hersey "In the early stages of its development, the basic ideas and methods to be contained in a computer program are set down in written forms, and these will presumably be copyrightable with no change in the 1976 Act. But the program itself, in its mature and usable form, is a machine-control element, a mechanical device, which on constitutional grounds and for reasons of social policy ought not be copyrighted."

¹³El subcomité de protección de software estuvo formado por Stanley Fuld, el Profesor Arthur Miller y el ejecutivo de Time, Inc. Gabriel Perle. La CONTU tenía otros tres subcomités: uno sobre fotocopias, otro sobre bases de datos y otro sobre obras creadas por computadora.

CONTU recomendó tres cambios estatutarios en la ley de Copyright de 1976:

- retirar la sección 117
- incluir la definición de programa de computadora en la sección 101
- y establecer una nueva sección 117 que permitiera a los usuarios adaptar los programas a sus necesidades y guardar copias de respaldo¹⁴.

CONTU sostenía la opinión de que la definición estatutaria de “copia” eliminaba toda duda sobre el hecho de que la versión ejecutable (de máquina) de los programas se encontraba cubierta bajo la ley de Copyright. Bajo la ley de 1976, las copias se definían como “objetos materiales, en los que una obra es fijada, por cualquier método conocido hoy o desarrollado posteriormente y por el cual la obra se pueda percibir, reproducir o comunicar, sea directamente o con la ayuda de una máquina o dispositivo”¹⁵. Bajo la ley de 1909, el término copia requería necesariamente la posibilidad de ser leído por un humano (en inglés “*human readability*”).

Ninguna parte del informe CONTU refiere a discusión alguna sobre algún método *sui generis* para amparar las versiones de máquina de los programas. La tendencia dominante en el informe es la de considerar el Copyright como la forma inevitable de cobertura legal, aunque dejaba en claro que el copyright sólo cubriría la expresión de las ideas volcadas en el programa y no las ideas y procesos en sí. La idea general de todo el informe es que todas las representaciones del programa eran lo mismo y ninguna presentaba problema alguno en relación a su cobertura bajo Copyright (Samuelson, 1984:703). De conformidad con las recomendaciones de la Comisión, en 1980, el Congreso de los EEUU sancionó una ley que establecía la protección de los programas de ordenador por medio del copyright (Lipszyc, 2006:107).

LOS PROBLEMAS DEL INFORME CONTU

El informe CONTU tiene una serie de problemas que se trasladaron directamente a las leyes en la materia. En primer lugar, interpreta la jurisprudencia en materia de *programas objeto* como si pacíficamente se hubiera aceptado la posibilidad de ampararlos bajo copyright cuando existieron numerosas controversias al respecto. La segunda crítica al informe CONTU es su negativa a analizar si la cobertura del Copyright para los programas objeto es consistente con la meta constitucional de promover el arte y las ciencias, que en el caso de las patentes y el copyright implica necesariamente la publicación del trabajo protegido. En consonancia con esto, el informe CONTU no es claro a la hora de hablar de *texto fuente* y *programa ejecutable*, mezcla los conceptos bajo la idea de programa – en el mismo sentido que la definición mencionada de la OMPI- sin hacer la salvedad de que muchas de las afirmaciones que hace son ciertas para una pero no necesariamente para otra versión del mismo. En varias ocasiones habla sobre la “legibilidad” de los programas, cuando debería quedar claro que una vez convertido a objeto, el programa deja de ser legible para una persona. Falla también en el análisis de la cuestión utilitaria. Existe una larga tradición

¹⁴<http://digital-law-online.info/CONTU/contu6.html>

¹⁵“material objects, other than phonorecords, in which a work is fixed by any method now known or later developed, and from which the work can be perceived, reproduced, or otherwise communicated, either directly or with the aid of a machine or device” en Samuelson, Pamela “CONTU Revisited: The Case against Copyright Protection for Computer Programs in Machine-Readable Form”. *Duke Law Journal*, Vol. 1984, No. 4 (Sep., 1984), pp. 663-769 Duke University School of Law pag. 701

jurídica de denegar protección de copyright a obras estrictamente utilitarias. El informe describe todos los programas como si funcionaran de manera semejante, cuando es sabido que muchos programas cumplen tareas que nada tienen que ver con comunicar o desplegar información para las personas. Tal es el caso de los sistemas de software de base, cuya función es estrictamente ejecutar tareas entre el equipo, los dispositivos y los programas o entre los programas mismos, sin comunicación alguna con las personas que utilizan las computadoras. Los sistemas operativos entran dentro de esta caracterización (Samuelson, 1984:704).

De estos puntos críticos, el más importante es justamente el que tiene que ver con la publicación de las expresiones amparadas bajo Copyright. Para maximizar sus ganancias, las empresas generalmente distribuyen sólo los programas en forma ejecutable, y mantienen en secreto los textos fuentes de los mismos. Permitir el amparo del Copyright a los programas objeto subvierte uno de los fundamentos de la ley de copyright: promover la publicación de trabajos que nutran el dominio público ha sido la base principal del intercambio en el cual se acepta otorgar a autores e inventores un monopolio limitado en el tiempo (Samuelson, 1984:707). En el actual estado del arte en materia de software, la sociedad sigue ofreciendo a los autores un monopolio exclusivo, pero a cambio no recibe las expresiones de ideas y las innovaciones que puedan ser aprendidas por el público, alimentar el avance de la disciplina y nutrir el dominio público. Cuando un autor publica un libro, un dibujo o una partitura musical, la expresión de las ideas contenidas allí queda a inmediata disposición del público que puede leer, estudiar y aprender, citar o incluso discutir la obra publicada. Nada de esto ocurre cuando un programa de computadora se distribuye exclusivamente en su versión objeto. Ni siquiera es posible ejercer el derecho de los autores a auditar eventuales casos de plagio en ese campo, ya que no es posible disponer del texto fuente de los programas.

La comparación que realiza Lipszyc al decir “que el programa de ordenador no sea directamente legible por el ser humano no obsta para que se lo considere como obra; lo mismo ocurre con otras obras, como las audiovisuales y las musicales fijadas en cintas electromagnéticas, que sólo son perceptibles por medio de aparatos electrónicos”(Lipszyc, 2006:109); es de difícil aplicación en este campo. Un músico versado en la materia puede comprender, al escuchar un registro fonográfico en qué tempo, con qué armonía, qué compases y qué acordes utiliza el músico que interpreta la obra. Incluso, es un ejercicio habitual de los estudiantes de música, escuchar registros de diverso tipo y volcarlos a partituras. La obra se aprecia en toda su expresión cuando se escucha un registro fonográfico, por más que para acceder a ella haga falta un aparato electrónico que la emita. En software, la sola utilización del ejecutable del programa nada dice de cómo ha sido escrito, ni de cómo se han resuelto los diferentes problemas y expresado los algoritmos. Existen algunos procedimientos de ingeniería inversa, pero estos no permiten la reconstrucción de la obra, por más versado que sea el programador en la materia. El texto permanece oculto por más que el programa se pueda utilizar. No es posible aprender cómo está hecho sin el código fuente. Los aprendices de música, y en particular los que tienen el oído entrenado, pueden aprender a partir de escuchar un registro y ejecutar incluso las canciones después de escucharlas. Los aprendices de programación no pueden aprender a programar o entender cómo está hecho un programa sólo usando el mismo, si no tienen la posibilidad de leer la obra original (en texto fuente).

Por otro lado, la asimilación de los programas de ordenador a las obras literarias y en consecuencia a su mismo marco regulatorio, trae otro problema sustancial para el campo: la excesiva duración del monopolio antes de que una pieza de software ingrese al dominio público. La dinámica propia de la informática indica que la vida útil de un programa es de muy pocos años, en la mayoría de los casos, no más de cinco años, por lo que la duración actual del monopolio de Copyright es definitivamente excesiva.

La adopción del régimen de Copyright para cubrir las obras de Software, sea en texto original o en versión ejecutable fue incluida en la modificación de la ley norteamericana de 1980, a partir de las recomendaciones de la Comisión CONTU. Esta normativa se trasladó a los convenios internacionales en la materia. El acuerdo sobre Aspectos de Propiedad Intelectual relacionados al Comercio (ADPIC) incluyó en su artículo 10, párrafo 1, que “los programas de ordenador, sean programas fuente o programas objeto, serán protegidos como obras literarias en virtud del Convenio de Berna (1971)”¹⁶. Los programas de ordenador también están incluidos en el tratado de Derechos de Autor de la OMPI (1996) en el cual indica que “los programas de ordenador están protegidos como obras literarias en el marco de lo dispuesto en el Artículo 2 del Convenio de Berna. Dicha protección se aplica a los programas de ordenador, cualquiera que sea su modo o forma de expresión”¹⁷.

Antes de que el tema sea incluido en la legislación argentina, la jurisprudencia ya se había pronunciado en el sentido de asimilar las obras de software a las obras amparadas en el artículo 1 de la Ley 11.723. Así lo entendió la justicia para declarar que los programas de computación, en cuanto expresiones de ideas vertidas en lenguaje escrito, están incluidos en el concepto de obra de acuerdo a la definición del art. 1° de la ley N° 11.723. La primera resolución en este sentido se produjo en 1988 y correspondió a la Justicia Nacional en lo Civil, en el caso Bunge y Born S.A. c/ M.E.C. Métodos Económicos Cuantitativos S.R.L.¹⁸

La inclusión de la cobertura del Software bajo el régimen de derechos autorales en el acuerdo ADPIC tuvo una inmediata repercusión en la normativa argentina, en particular, bajo la consideración del artículo 75, inciso 22 de la Constitución Nacional, que otorga rango constitucional a los tratados internacionales¹⁹. El decreto 165/94, emitido el 3 de febrero de 1994 por el Poder Ejecutivo Nacional incluyó las obras de software en la enumeración de las obras amparadas en la Ley 11.723 de Propiedad Intelectual (art. 1ro.) estableciendo también el procedimiento de registro de estas obras. Sin embargo, este decreto sólo tuvo injerencia en el ámbito civil, porque el Poder Ejecutivo carece de facultades para, por vía reglamentaria, crear nuevos tipos penales²⁰.

En octubre de 1998 fue sancionada la Ley 25.036 que incorpora de forma expresa los programas de cómputo, en fuente y objeto en el artículo 1 de la Ley de Propiedad Intelectual y por lo tanto les ofrece cobertura jurídica, tanto en el ámbito civil como penal, dando conformidad a las obligaciones adquiridas a partir de la firma de ADPIC. La modificación legislativa incluye la cobertura del programa fuente y objeto, la asignación de titularidad de los derechos de propiedad intelectual a la firma contratante en caso de la realización de software por encargo o por parte de empleados (art.

¹⁶ADPIC Art. 10 . Disponible en http://www.wto.org/spanish/docs_s/legal_s/27-trips_04_s.htm (Visitado el 8 de agosto de 2010)

¹⁷Tratado de la OMPI sobre Derecho de Autor (WCT) adoptado en Ginebra, el 20 de diciembre de 1996. Disponible en http://www.wipo.int/treaties/es/ip/wct/trtdocs_wo033.html#TopOfPage

¹⁸Millé, Antonio “Los programas de computación en el régimen argentino de Derecho de Autor ”. Disponible en línea en <http://www.mille.com.ar/mille-imagenes/programas-de-computacion.pdf>

¹⁹Emery, Miguel Angel “Propiedad Intelectual” Editorial Astrea. 4ta. Reimpresión.(2009) Pag. 55.

²⁰Fernández Delpech, Horacio. “Protección Jurídica del Software” Editorial Abeledo Perrot. (2000)

4 inciso d), la posibilidad de realizar una copia de salvaguarda del programa para quien haya recibido una licencia de uso de un programa (art. 9) y la inclusión de los contratos de licencia como mecanismo para la explotación de la propiedad intelectual (art. 55 bis)²¹.

LICENCIAMIENTO DE SOFTWARE

Es común, y está prácticamente generalizada la idea de que los proveedores de software no venden sus productos sino que los licencian. El licenciamiento ha sido el paradigma fundamental de la distribución de programas, en principio como medida para saltar incluso las normativas de la propia ley de copyright. De hecho, históricamente, el licenciamiento se utilizaba para preservar secretos comerciales cuando la cobertura de copyright no estaba asegurada en los programas de computadora. Sin embargo, una vez establecida la cobertura tanto para fuentes como para objetos, la práctica del licenciamiento persistió, por diversas razones; la principal es que el sistema de licencias permite a los proveedores de software evadir la doctrina de la primera venta²². La doctrina de la primera venta indica que el comprador puede hacer con esa particular copia lo que desee, incluso revenderla, en tanto no infrinja algún otro de los derechos exclusivos del titular del copyright. Otra razón para conservar el sistema de licenciamiento es que permite mantener el control de la obra más allá de la distribución a los consumidores.

Por otra parte, las licencias también permiten atar la provisión de diferentes paquetes de productos (*bundling*), aún cuando estas estrategias puedan eventualmente colisionar con las leyes de defensa de la competencia²³. El licenciamiento y posterior control de reventa permite además la discriminación de precios en función del cliente²⁴.

“Una licencia es un instrumento legal que expresa derechos de autor y que indica qué se puede y qué no se puede hacer con una obra intelectual”²⁵. Sobre esta lógica, la Free Software Foundation y el movimiento de Software Libre utilizó el sistema de licencias para solucionar los problemas derivados de la inclusión del Software en el marco del Copyright y la creciente distribución de programas de cómputo en formato binario sin disponibilidad del código fuente.

²¹Ley de Propiedad Intelectual 11.723. Disponible en http://www.mincyt.gov.ar/11723_act.htm

²²Véase el caso *Microsoft Corp. V. Harmony Computers* en Lemley, Menell, Merges, Samuelson. “Software and Internet Law” (2006) Third Edition. Aspen Publishers. Pag. 301.

²³Sirven como ejemplo los casos de abuso de posición dominante iniciados contra Microsoft por distribuir sistemas operativos con programas de aplicación, en particular reproductores multimedia. Véase por ejemplo http://www.elpais.com/articulo/internet/Tribunal/UE/confirma/multa/4972/millones/Microsoft/abuso/posicion/dominante/elpepuntec/20070917elpepuntec_1/Tes

²⁴Lemley, Menell, Merges, Samuelson. “Software and Internet Law” (2006) Third Edition. Aspen Publishers. Pag. 319.

²⁵Vercelli, Ariel. (2009) “Guía de Licencias Creative Commons 2.0” disponible en <http://www.arielvercelli.org/2009/09/07/guia-de-licencias-creative-commons-20/>

"Programs must be written for people to read, and only incidentally for machines to execute." - Abelson & Sussman, SICP, preface to the first edition²⁶

Existen varios problemas en la inclusión del software en el marco del copyright que los mismos programadores de software, y en particular los partidarios del Software Libre, han intentado solucionar mediante la construcción de un sistema jurídico de licencias de derechos de autor: a los problemas relacionados a la distribución de binarios sin fuentes, se suma la prohibición de hacer obras derivadas y adaptaciones en un área tan dinámica del conocimiento humano, la prohibición de copiar los programas, la no disponibilidad de los textos fuentes para estudio y conocimiento público, la imposibilidad de estudiar cómo funcionan los programas sea con fines prácticos como académicos, así como la duración excesiva de los plazos de protección.

Formado en el laboratorio de Inteligencia Artificial del Instituto Tecnológico de Massachussets, el físico y programador de sistemas operativos Richard Stallman planteó los fundamentos de lo que luego se conocería como el *Software Libre*. A mediados de la década del 80, cuando ya se había consolidado la idea de regular la distribución de programas de computadora bajo la normativa del Copyright, Stallman ideó un sistema legal que permitiera sortear los principales problemas presentes en esa regulación. En 1984 renunció a su puesto en el laboratorio de inteligencia artificial del MIT²⁷ y se dedicó a escribir un sistema operativo completamente desde cero, similar a UNIX, pero de libre distribución al que denominó Sistema GNU (acrónimo recursivo que significa GNU is not UNIX)²⁸.

Herederero de una tradición académica en la cual todo programa se distribuía con su respectivo código fuente, Stallman rechazó de plano la conformación de un modelo de distribución en la cual el código fuente permaneciera oculto o fuera delito la redistribución, fundó la Free Software Foundation y acuñó el concepto de Software Libre.

Un programa es libre cuando los términos de su licencia permiten a usuarios y desarrolladores cuatro libertades específicas: a) usar los programas con cualquier propósito, b) estudiar su código fuente y realizar adaptaciones según las propias necesidades, c) hacer y distribuir copias idénticas (aún cobrando por ellas), d) realizar mejoras y cambios al programa y redistribuir las versiones mejoradas.

El aspecto central de la estrategia de Stallman no fue técnico sino jurídico: construir un sistema de licencias para promover estas libertades. De este modo, el Software Libre aparece como solución

²⁶<http://mitpress.mit.edu/sicp/full-text/sicp/book/node3.html>

²⁷*En enero de 1984 abandoné mi empleo en el MIT y comencé a escribir software GNU. Abandonar el MIT era imprescindible si quería que nadie interfiriera en la distribución de GNU como software libre. De haberme quedado, el MIT podría haberse apropiado de mi trabajo e impuesto sus propios términos de distribución, o incluso convertir el trabajo en un paquete de software propietario. No tenía ninguna intención de hacer una gran cantidad de trabajo para ver como se convertía en algo inútil en relación a su propósito inicial: crear una nueva comunidad dedicada a compartir software.* Narra el propio Stallman en su libro "Software libre para una sociedad libre", para evitar que el MIT reclamara derechos sobre su obra. Véase <http://biblioweb.sindominio.net/pensamiento/softlibre/softlibre005.html#toc1> Publicado originalmente en "Open Sources: Voice from the Open Source Revolution", O'Reilly, (1999)

²⁸Véase <http://www.gnu.org>

a los problemas identificados por la propia comunidad de programadores en materia de Copyright y Software: impulsa la publicación y distribución del código fuente junto con el binario con lo cual torna legible y apropiable socialmente el conocimiento incluido en los programas, y termina con el monopolio de copia, por lo que no sólo elimina el problema del copyright sobre obras netamente utilitarias, sino que limita el problema de la duración excesiva del monopolio de copia sobre los mismos.

Stallman comprendió que dedicar un programa libre al *dominio público* no necesariamente aseguraría la libertad de usuarios y desarrolladores, porque cualquier persona podría tomar la versión del dominio público y redistribuir obras derivadas como software propietario, sólo en binario y sin código fuente²⁹. Para esto, la estrategia de Stallman y la Free Software Foundation fue reclamar el copyright sobre las obras realizadas y distribuirlas bajo licencias que garantizaran y preservaran la libertad³⁰. “Copyleft utiliza la ley de *copyright*, pero dándole la vuelta para servir a un propósito opuesto al habitual: en lugar de privatizar el software, ayuda a preservarlo como software libre.”³¹

Así fue como surgió la Licencia Pública General del Proyecto GNU³², más conocida como la Licencia GNU GPL, una licencia de software que reivindica el copyright del programa y fija los términos de distribución conocidos como Copyleft³³, la única cláusula restrictiva de este modelo de licenciamiento, que fija como condición de la redistribución, sea de copias literales como modificadas del programa, que quienes reciban la obra reciban con ella las mismas libertades aseguradas previamente. Es lo que en el campo de las Licencias Creative Commons se conoce como “Share Alike”³⁴ o distribuir obras derivadas de la misma forma que la obra original, es decir, bajo los mismos términos de licencia.

Poco más de un cuarto de siglo después de aquella primera iniciativa de Richard Stallman y la Free Software Foundation, el movimiento de Software Libre se ha probado exitoso tanto en materia de innovación y negocios como en solidez jurídica.

Como ejemplo, vale citar que en el año 2000, el Presidente del Comité Asesor en Tecnologías de Información recomendó al gobierno federal de los EEUU que sostenga el desarrollo de Software Libre como una estrategia para sostener el liderazgo de los EEUU en el desarrollo de software crítico³⁵.

Yochai Benkler distingue el modo de producción de bienes comunes entre pares del modo de propiedad, basado en modelos sostenidos por contratos, empresas y mercados. La característica de

²⁹Lemley, Menell, Merges, Samuelson. “Software and Internet Law” (2006) Third Edition. Aspen Publishers. Pag. 371.

³⁰Las licencias reconocidas por la Free Software Foundation se encuentran en <http://www.gnu.org/licenses/licenses.html>

³¹<http://biblioweb.sindominio.net/pensamiento/softlibre/softlibre005.html#toc10>

³²Una copia de la Versión 3 de la Licencia GNU GPL está disponible en <http://www.gnu.org/copyleft/gpl.html>

³³“El copyleft es un método general para hacer un programa (u otro tipo de trabajo) libre, exigiendo que todas las versiones modificadas y extendidas del mismo sean también libres”. <http://www.gnu.org/copyleft/copyleft.es.html>

³⁴Un ejemplo de este tipo de licencias está disponible en <http://creativecommons.org/licenses/by-sa/2.5/>

³⁵Citado en Benkler, Yochai “Coase’s Penguin, or, Linux and The Nature of the Firm” The Yale Law Journal (2002) <http://www.yalelawjournal.org/the-yale-law-journal/content-pages/coase%27s-penguin,-or,-linux-and-the-nature-of-the-firm/> (visitado el 8 de agosto de 2010).

este modo de producción de colaboración a gran escala es que sigue diversas tendencias motivacionales más allá de las ganancias, el mercado y la cadena de mando de una empresa. Existen numerosos trabajos, en particular de economistas y sociólogos, que tratan de explicar la motivación y las razones del éxito masivo de la producción de Software Libre en un mercado que está en la cresta de la innovación a nivel mundial. La existencia de proyectos exitosos como el núcleo Linux, las diversas distribuciones de GNU/Linux, el navegador Mozilla Firefox, el servidor web Apache, los aplicativos de oficina de OpenOffice y la existencia de departamentos de desarrollo de Software Libre en grandes empresas como IBM, Sun Microsystems, Google, o de empresas específicamente dedicadas al Software Libre como Canonical (promotora de la distribución Ubuntu) o Red Hat indican que el fenómeno económico no es menor. El fenómeno político tampoco lo es, si se considera la incorporación del Software Libre como política de Estado en numerosos municipios y países (Munich en Alemania, París y Región en Francia, Extremadura en España, Rosario en Argentina, Massachusetts y California en los EEUU, Brasil, Ecuador y Venezuela entre muchos otros). El software libre no es un fenómeno marginal. Es un fenómeno de innovación en un área de fuerte dinamismo como la industria del Software, que ha probado que puede avanzar el conocimiento bajo la explícita renuncia al monopolio de copia y la publicación abierta de los códigos fuentes de todos sus programas³⁶.

Incluso desde los aspectos legales, no existen dudas fundadas sobre la validez de las licencias Copyleft. Si bien en su libro *"Problemática Jurídica del Software Libre"*³⁷ el abogado argentino Martín Carranza Torres argumenta que la cláusula Copyleft debe ser tomada como inválida, sus argumentos se limitan a una serie de tesis de dudosa rigurosidad. La primera, es que las licencias de Software Libre igualan los derechos de los autores y de los usuarios, cosa que de por sí es errónea, ya que los autores de Software Libre no renuncian a la titularidad de sus derechos como autores sino que ejercen su derecho al permitir explícitamente mediante una licencia de uso la copia, estudio y modificación de sus programas. La reserva del copyright garantiza que los programadores puedan eventualmente litigar en caso de violaciones de la licencia, realizar doble licenciamiento y recibir los beneficios de la atribución de los derechos morales a la paternidad de la obra, hecho que en el campo del Software Libre es de capital importancia³⁸. En el sistema social del Software Libre, construido en buena medida a partir de principios de meritocracia, el reconocimiento de la autoría de software es una motivación importante para los contribuyentes³⁹.

Otra de las tesis de Carranza Torres indica que "el copyleft y la licencia GPL no son estrategias jurídicamente aceptables, puesto que a través de ellas el autor de la versión original de un Software Libre obliga a sus potenciales usuarios a renunciar por anticipado a la exclusividad que, en justicia se les reconoce y garantiza. Esto fomenta un modelo en el que quien no ha hecho suficiente mérito (el autor de la versión original) termina siendo el que hace realidad el principio de Autor = Usuario respecto de las modificaciones introducidas a su software. La GPL es válida, pero

³⁶Un creciente uso de Software Libre por parte de empresas queda de manifiesto en numerosos reportes de consultoras especializadas, entre ellas Gartner, que anticipó que para el año 2012, el 80% de los programas comerciales incluirán elementos de tecnologías abiertas (Gartner, "The State of Open Source 2008" Abril 2008). http://news.cnet.com/8301-13505_3-9863802-16.html

³⁷Carranza Torres, Martín "Problemática Jurídica del Software Libre" Lexis Nexis. (2007) Cap. V Disponible en línea en <http://www.carranzatorres.com.ar/images/pdf/problematica-juridica-del-software-libre-capV.pdf>

³⁸Derechos morales al reconocimiento de la paternidad de la obra en los términos del Art. 6bis del Convenio de Berna. Disponible en http://www.wipo.int/treaties/es/ip/berne/trtdocs_wo001.html#P132_23079

³⁹Motives for Writing Free Software. <http://www.gnu.org/philosophy/fs-motives.html>

las cláusulas que obligan a los licenciarios a renunciar a la exclusividad sobre eventuales obras derivadas son nulas y se deben tener por no escritas⁴⁰. Si esto fuera cierto, ningún autor de obras protegidas podría fijar las condiciones o tener injerencia sobre las traducciones o adaptaciones de sus obras, cosa que sabemos, no ocurre en ningún marco legal vigente. La cláusula copyleft no es otra cosa que el ejercicio de los derechos exclusivos de disponer, traducir, adaptar o autorizar la traducción y adaptación de la obra de la que goza el autor en Argentina en función del artículo 2 de la Ley 11.723⁴¹.

Existe muy poca jurisprudencia en relación a la cláusula Copyleft de la licencia GPL, ninguna que sostenga la tesis propuesta por el Dr. Carranza Torres, y hay unanimidad en los tribunales en relación a la validez jurídica de la licencia de Software Libre diseñada por la Free Software Foundation. La razón por la cual existe poca jurisprudencia es la solidez legal de la licencia, que hace que la mayoría de los casos reportados de violación se resuelvan en instancias preliminares, siempre a favor del cumplimiento de la misma.

La licencia GPL es tan sencilla que no requiere aceptación por parte de los usuarios, ya que explícitamente les otorga la libertad de usar el software sin limitación alguna, estudiarlo, modificarlo y copiarlo según sus propias necesidades. La única restricción que fija la licencia es que toda aquella obra derivada de un código liberado bajo GPL mantenga esta cláusula y otorgue a los próximos usuarios las mismas libertades. Sólo en el caso de redistribuir programas se debe aceptar y mantener la licencia⁴². De hecho, las escasas ocasiones en que los casos no se resolvieron con una simple carta de notificación, y han llegado a tribunales, la justicia ha fallado en favor de la validez de la cláusula copyleft. El fallo de la Corte del Distrito de Frankfurt en Alemania contra D-Link Germany GmbH por violación de los términos de licencia del núcleo Linux, disponible públicamente bajo una licencia GPLv2, es uno de los ejemplos más claros de cumplimiento legal (*enforcement*) de la GPL⁴³ resuelto en instancia judicial⁴⁴. Lo más interesante del caso alemán fue que la demanda incluyó la cobertura de los costos de la misma, ya que la firma D-Link había corregido su conducta infractora tras la carta de notificación enviada por los letrados que promovieron la acción civil. La corte obligó a la firma alemana a cubrir los costos del litigio tras encontrarla culpable de violar la licencia GPL. Al caso alemán de 2006 se suma un reciente fallo, de agosto de 2010⁴⁵, en la Corte del Distrito Sur de Nueva York⁴⁶, en los EEUU, donde el caso contra Westinghouse Digital Electronics por violación de la licencia del programa BusyBox terminó con una condena para la

⁴⁰Carranza Torres, Martín "Problemática Jurídica del Software Libre" Lexis Nexis. (2007)

⁴¹Emery, Miguel Angel (2009) Pag. 62-73-74.

⁴²Moglen, Eben "Free Software Matters: Enforcing the GPL, I" (2001) <http://moglen.law.columbia.edu/publications/lu-12.html> Eben Moglen es Profesor de Leyes e Historia Legal en la Universidad de Columbia y ejerce como asesor legal de la Fundación del Software Libre. Es además el Fundador del Software Freedom Law Center <http://www.softwarefreedom.org/>

⁴³En un comunicado de prensa sobre el caso, el Hacker de Linux Harald Welte expresó: "The Free Software community is very happy to see more and more vendors to use Linux and other Free Software in their products. However, Free Software is copyrighted material, much like any other software. Redistribution may only take place in accordance with its license." http://gpl-violations.org/news/20060922-dlink-judgement_frankfurt.html

⁴⁴GPL passes acid test in German Court <http://www.linux.com/archive/feed/57353>

⁴⁵GPL scores historic court compliance victory (04/08/2010) http://www.theregister.co.uk/2010/08/04/gpl_violation_westinghouse/

⁴⁶http://sfconservancy.org/docs/2010-07-27_dj-opinion.pdf

empresa que distribuía versiones modificadas del programa en sus sets de HDTV sin respetar los términos de la licencia GNU GPL.

“Nuestra forma de hacer las normas ha sido tan inusual como nuestra forma de hacer software, pero ese es justamente nuestro punto. El Software Libre importa porque manifiesta que esta forma diferente de hacer las cosas es, después de todo, la correcta”(Moglen, 2001).⁴⁷

BIBLIOGRAFÍA

- Emery, Miguel Angel “Propiedad Intelectual” Editorial Astrea. 4ta. Reimpresión.(2009)
- Fernández Delpech, Horacio. “Protección Jurídica del Software” Editorial Abeledo Perrot. (2000)
- Lemley, Menell, Merges, Samuelson. “Software and Internet Law” Third Edition. Aspen Publishers (2006) .
- Lipszyc, Delia “Derecho de autor y derechos conexos”. Ediciones UNESCO, CERLALC, ZAVALLIA. Reedición (2006).

PAPERS Y DOCUMENTOS CONSULTADOS

- ADPIC Art. 10 . Disponible en http://www.wto.org/spanish/docs_s/legal_s/27-trips_04_s.htm
- Baker v. Selden, 101 U.S. 99 Volume 101 (1879) U.S. Supreme Court <http://supreme.justia.com/us/101/99/case.html>
- Benkler, Yochai “Coase’s Penguin, or, Linux and The Nature of the Firm” The Yale Law Journal (2002) <http://www.yalelawjournal.org/the-yale-law-journal/content-pages/coase%27s-penguin,-or,-linux-and-the-nature-of-the-firm/>
- Carranza Torres, Martín “Problemática Jurídica del Software Libre” Lexis Nexis. (2007) Cap. V Disponible en línea en <http://www.carranzatorres.com.ar/images/pdf/problematica-juridica-del-software-libre-capV.pdf>
- CONTU Final Report. <http://digital-law-online.info/CONTU/PDF/index.html>
- Convención de Berna http://www.wipo.int/treaties/es/ip/berne/trtdocs_wo001.html#P132_23079
- Disposiciones Tipo para Leyes en Materia de Derecho de autor. Documento OMPI CE/MPC/I/2-II del 11 de Agosto de 1989. en http://www.wipo.int/mdocsarchivos/CE_MPC_I_1989/CE_MPC_I_2_II_S.pdf
- Free Software Foundation <http://www.fsf.org>
- GPL Violations <http://gpl-violations.org/index.html>
- Heinz, Federico, (2010) “Software vs. Copyright” disponible en línea en <http://www.vialibre.org.ar/2010/07/23/software-vs-copyright/>
- Ley de Propiedad Intelectual 11.723. Disponible en http://www.mincyt.gov.ar/11723_act.htm

⁴⁷Moglen, Eben, “Free Software Matters: Enforcing the GPL, II” (2001) <http://moglen.law.columbia.edu/publications/lu-13.html>

- Millé, Antonio “Los programas de computación en el régimen argentino de Derecho de Autor”. Disponible en línea en <http://www.mille.com.ar/mille-imagenes/programas-de-computacion.pdf>
- Moglen, Eben “Free Software Matters: Enforcing the GPL, I” (2001) <http://moglen.law.columbia.edu/publications/lu-12.html>
- Moglen, Eben, “Free Software Matters: Enforcing the GPL, II” (2001) <http://moglen.law.columbia.edu/publications/lu-13.html>
- Proyecto Creative Commons <http://www.creativecommons.org>
- Proyecto GNU <http://www.gnu.org>
- Samuelson, Pamela “CONTU Revisited: The Case against Copyright Protection for Computer Programs in Machine-Readable Form”. *Duke Law Journal*, Vol. 1984, No. 4 (Sep., 1984), pp. 663-769 Duke University School of Law
- Stallman, Richard “Software libre para una sociedad libre”, traducción al español por Traficantes de Sueños. Véase <http://biblioweb.sindominio.net/pensamiento/softlibre/softlibre005.html#toc1> Publicado originalmente en “*Open Sources: Voice from the Open Source Revolution*”, O'Reilly, (1999)
- Tratado de la OMPI sobre Derecho de Autor (WCT) adoptado en Ginebra, el 20 de diciembre de 1996. Disponible en http://www.wipo.int/treaties/es/ip/wct/trtdocs_wo033.html#TopOfPage
- Vercelli, Ariel. (2009) “Guía de Licencias Creative Commons 2.0” disponible en <http://www.arielvercelli.org/2009/09/07/guia-de-licencias-creative-commons-20/>