

LAS LICENCIAS DE SOFTWARE DESDE LA PERSPECTIVA DEL USUARIO FINAL

Por Juan Carlos Brocca y René Casamiquela

CURZA - UNCo

CONSIDERACIONES PRELIMINARES

El usuario final

El presente artículo está dirigido al *usuario final* de computadoras. Se lo define como la persona -física o jurídica- que utiliza las herramientas informáticas de manera directa en el ámbito de su trabajo. El usuario final se encuentra directamente alcanzado por las condiciones establecidas en la autorización concedida por quienes desarrollan, comercializan o distribuyen programas de computadoras.

El software

Una computadora es un dispositivo con capacidad para procesar datos a partir de un conjunto organizado de instrucciones, que son interpretadas y ejecutadas de manera automática. Se denomina *Programa de Computación*, o simplemente *Programa*, a la secuencia coherente de instrucciones que permite a la computadora realizar una determinada tarea. A la acción de poner en funcionamiento un programa en una computadora se la denomina *ejecución o utilización* del programa.

Para operar adecuadamente, los programas requieren medios que permitan su funcionamiento, tales como las estructuras de datos - que les permiten manipularlos - y la documentación, que describe la operación de los mismos. Para referenciar a todo el conjunto se utiliza el término *Software*, al que puede definirse, de acuerdo con el *Institute of Electrical and Electronics Engineers* (IEEE), como la parte de un sistema de computación constituido por los programas, procedimientos, reglas informáticas, documentación y datos asociados, necesarios para ejecutar ciertas tareas. El vocablo *software* surge por oposición a la palabra *hardware* - empleada para denominar a los componentes físicos de la computadora - en un intento de señalar la intangibilidad del mismo.

Desde el punto de vista del usuario final, la utilización de computadoras modernas con fines generales permite realizar una clasificación del software sobre la base de sus funciones. Se establecen así dos categorías principales: el *software de base o de sistemas*, y el *software de aplicación*. Entre los primeros, se destaca el *sistema operativo*, cuya finalidad es la de controlar las actividades de la computadora, administrar los recursos y permitir la comunicación con el usuario. El software de aplicación está destinado a realizar las tareas específicas de los usuarios finales. A esta clase pertenecen los programas de uso común y frecuente, tales como, procesadores de texto, hojas de cálculo, administradores de bases de datos y programas de comunicaciones. Las aplicaciones se diseñan para operar sobre un determinado tipo de sistema operativo.

Procedimiento de elaboración de software

Para comprender el mecanismo de protección legal del software es conveniente tener presente el principal método de construcción de los programas para computadoras. En dicho proceso, los desarrolladores se valen de un medio idóneo para la comunicación entre el ser humano y la máquina, denominado *lenguaje de programación*; esto es un conjunto de reglas que permite expresar adecuadamente las instrucciones que debe ejecutar la computadora.

Generalmente, por razones prácticas y productivas, la selección del lenguaje se orienta a lograr un programa entendible y fácil de modificar por el ser humano; como así también, independiente del tipo de máquina sobre el que será ejecutado. Se lo conoce con el nombre de *programa fuente* de una aplicación. Luego, mediante un proceso de traducción adecuado, se lo convierte de tal manera que

pueda ser interpretado por la computadora y pierde la capacidad de ser directamente legible por el ser humano. Se obtiene así el denominado *programa objeto*, al que posteriormente se transformará en un *programa ejecutable*.

Protección legal del software

En los orígenes de la informática, el software era considerado un componente más de la máquina y los fabricantes de computadoras lo desarrollaban y distribuían, incluso poniendo a disposición de los usuarios el código fuente, casi sin restricciones en cuanto a su utilización o destino. Pero, para evitar inconvenientes por acciones antimonopólicas, a partir de 1970 la firma IBM independiza la comercialización del software de la venta de sus equipos y ejerce derechos legales sobre el mismo.

Se inicia así un cambio de modelo económico donde el software se desvincula del fabricante del equipo y se desarrollan aplicaciones que pueden correr sobre diversas plataformas. En este nuevo escenario, la mayoría de los diseñadores de software comienza a utilizar todos los medios técnicos y legales disponibles para proteger su desarrollo y mantener su posición en el mercado.

Sin embargo, las características inmateriales del software no han facilitado su encuadre en las categorías jurídicas existentes y, por lo tanto, ha resultado difícil de tutelar. Las opiniones al respecto no son unánimes: en algunos casos, se propone regularlo a través de marcas o patentes -en el marco del régimen jurídico de la propiedad industrial- o utilizar los mecanismos de protección de la propiedad intelectual, mientras que en otros, se pretende desarrollar un sistema especial.

Debe aclararse que la legislación Argentina diferencia los derechos de propiedad intelectual de los de propiedad industrial. Es en ese contexto que se los menciona en el presente artículo. No obstante, a nivel internacional, a la expresión '*propiedad intelectual*' actualmente se le otorga un sentido más amplio, incluyendo también las patentes y marcas registradas.

Las marcas registradas

La *marca* es un signo o un medio material de cualquier clase o forma que permite distinguir un producto o servicio de otros similares. Un complemento es el *nombre comercial*, un indicativo de quien comercializa un producto. Esta combinación resulta un medio de protección eficaz del software debido a la utilización de simbología y nombres específicos para cada producto. De esta manera, es posible distinguir un producto de otro, a tal extremo que los usuarios lo identifican por su marca, asociándolo con un determinado nivel de calidad.

Los países cuentan con un Registro de Marcas con alcance territorial, por lo que éstas tienen validez dentro del estado que otorga el registro. Cumplimentados los trámites pertinentes, se le garantiza, al tenedor del registro, la propiedad y uso exclusivo de la marca durante cierto período de tiempo, pudiendo ejercer defensa legal frente al uso no autorizado por parte de terceros. En la Argentina se aplica la Ley 22.362 - Ley de Marcas - y sus Decretos Reglamentarios 558/81 y 1141/03.

El régimen de patentes

Una *patente de invención* es un derecho que un estado otorga durante un determinado período de tiempo a un inventor, permitiéndole explotar de manera exclusiva su invención, impidiendo a otros la fabricación, venta o utilización del producto patentado, sin su consentimiento. Como contrapartida, el inventor brinda a la sociedad el producto de su investigación. Finalizado el plazo establecido, la patente pasa al dominio público, por lo que es posible hacer uso de la misma sin tener que abonar regalías. También este régimen de protección tiene alcance territorial, por lo que el solicitante debe presentar un requerimiento en cada país donde le interese proteger su invención. En ese caso, es de aplicación el *Convenio de París*, un tratado internacional de 1884, que establece que los países signatarios del mismo se constituyen en Unión y que "Los nacionales de cada uno de los países de la Unión gozarán en todos los demás países de la Unión, en lo que se refiere a la protección de la

propiedad industrial, de las ventajas que las leyes respectivas concedan actualmente o en el futuro a sus nacionales”. En la República Argentina la protección está establecida por la Ley 24.481 de Patentes de Invención y Modelos de Utilidad y sus modificatorias, Leyes N° 24.572 y 25.859. Asimismo, el Convenio de París ha sido ratificado por la Ley 17.011.

El régimen de patentes no fue visto en un principio como el método idóneo de protección legal del software. Tan es así, que varios países lo excluyen del significado de invención y, por lo tanto, de la protección que brinda el derecho de patentes. La Oficina de Patentes y Marcas de los Estados Unidos de Norteamérica, en un primer momento rechazaba esos pedidos de patentes, utilizando como fundamento la imposibilidad de otorgar propiedad sobre una concepción mental. Pero, desde 1987 admite patentar aplicaciones que incorporen innovación, bajo la figura de algún conjunto inventivo que incluya software. Por tal motivo, en la actualidad las grandes corporaciones norteamericanas dedicadas a la producción de software, enfrentan en su país demandas por infracción a la Ley de patentes, a la vez que protegen sus propios desarrollos con el mismo instrumento legal.

En Europa, la Convención Europea de Patentes celebrada en Munich en 1973 estableció expresamente que el software no es patentable. Pero, en el año 2002, fue presentada una propuesta para una directiva sobre “la patentabilidad de invenciones implementadas para computadoras” con la intención de legalizarlas en esa región. Luego de un complejo debate que no se considera totalmente cerrado, se aprobó el rechazo a las patentes de software.

La legislación Argentina, en el artículo 6 de la Ley 24.481 excluye taxativamente los programas de computación de la protección legal bajo este régimen. No obstante, algunos autores afirman que tal disposición se contradice con los Aspectos de los Derechos de Propiedad Intelectual relacionados con el Comercio (ADPIC) de la Organización Mundial del Comercio que, de acuerdo con la Constitución Nacional, es una norma superior. Por tal motivo, se esperan discusiones sobre el tema en un plazo no muy lejano.

El derecho de autor

El *derecho de autor* y el *copyright* constituyen dos concepciones de la propiedad intelectual no totalmente equivalentes porque parten de sistemas jurídicos diferentes. Desde 1886 cuando se firmó el *Convenio de Berna* para la protección de las obras literarias y artísticas - destinado a otorgar protección jurídica internacional al derecho de autor - se viene dando un proceso de armonización entre ambos conceptos, con cierta inclinación hacia la preeminencia del primero. Bajo la figura de *derecho de autor*, éste ejerce en forma exclusiva el derecho a utilizar o autorizar a terceros el uso de su obra bajo condiciones acordadas.

En Argentina, la ley 11.723 de Propiedad Intelectual establece que “El derecho de propiedad de una obra científica, literaria o artística, comprende para su autor la facultad de disponer de ella, de publicarla, de ejecutarla, de representarla, y exponerla en público, de enajenarla, de traducirla, de adaptarla o de autorizar su traducción y de reproducirla en cualquier forma.” Mediante la Ley 25.036 se introduce la siguiente modificación: “a los efectos de la presente ley, las obras científicas, literarias y artísticas comprenden los escritos de toda naturaleza y extensión, entre ellos los programas de computación fuente y objeto; las compilaciones de datos o de otros materiales”. Es decir, que legalmente está protegiendo al software.

Régimen legal especial

Con la finalidad de establecer un régimen jurídico *sui generis*, se han elaborado proyectos que toman los elementos más relevantes de la legislación vigente y contemplan las características especiales del software. Hasta el momento, no se considera viable una iniciativa de este tipo porque no se cuenta con ningún tratado internacional que la respalde y no podría articularse con alguno de los convenios existentes.

Licencias de software

En la gran mayoría de los casos, el software se entrega con una licencia de uso. Dicha licencia es un contrato entre el productor y el usuario que establece cuáles son los derechos y obligaciones de cada una de las partes.

Al instalar, utilizar o copiar un producto de software bajo licencia, el usuario está aceptando las condiciones estipuladas en la misma y queda obligado por los términos de dicho contrato. Es importante, entonces, leer detenidamente la licencia de un programa para conocer las condiciones para su utilización. Debe tenerse en cuenta que algunas licencias son tan restrictivas que incluso el producido con la herramienta en cuestión no es de libre disponibilidad, mientras que otras invitan a su modificación o duplicación.

Clasificación del software sobre la base de las Licencias

Las licencias presentan algunos aspectos que permiten realizar una clasificación del software en relación con éstas. Este agrupamiento no es único, dado que es frecuente encontrar la utilización tendenciosa de los distintos términos, como también la asignación de significados diferentes.

De acuerdo con la normativa vigente, la utilización del software puede ser *legal*, si cumple con los requerimientos establecidos para la obtención de la licencia y se emplea de acuerdo con lo establecido en ella; o *ilegal*, cuando es adquirido en desarreglo con las condiciones estipuladas o se utiliza para fines no contemplados en el contrato.

Con relación al costo que representa la obtención del producto, el software puede ser *gratuito*: no requiere erogación monetaria o a lo sumo una equivalente al valor del soporte que lo contiene; u *oneroso*: donde el usuario debe adquirirlo efectuando un pago al proveedor.

En cuanto a la comercialización, si bien no existe una única interpretación al respecto, las licencias pueden establecer que el software sea *comercial o no comercial*. En el primer caso, quienes lo producen lo hacen como parte de una actividad económica y por lo tanto obtienen un beneficio por el producto, su distribución o soporte. En el segundo, lo hacen sin fin de lucro. Corresponde aclarar que las empresas desarrolladoras de software más importantes a nivel mundial - que obtienen sus ingresos a partir de la venta de licencias de programas - establecen claramente que sus productos son comerciales e interpretan que todo software que pueda obtenerse de una manera diferente es no comercial.

Algunas licencias establecen condicionamientos al usuario en relación con la utilización que puede dar al producto. Por eso, es frecuente encontrar el término de software de *uso comercial*, que no establece limitaciones en cuanto a las actividades productivas realizadas con el programa. Los de *uso no comercial* son aquellos en los que la licencia exige una utilización predeterminada, por ejemplo, la versión educativa de un producto, impide el empleo del software en otra actividad.

Cuando las licencias permiten la distribución del código fuente de la aplicación, se clasifica al software en *abierto*. Es *cerrado*, si la licencia contempla solamente la entrega del código ejecutable.

Pero tal vez sea la característica más importante la que considera los derechos concedidos a los usuarios con relación al uso, modificación, copia y distribución. En este caso el software puede ser *privativo*, cuando las licencias plantean restricciones al respecto o *libre*, en aquellos casos en que otorgan plenas libertades a los usuarios. Nótese que para que un software pueda considerarse libre, necesariamente debe ser abierto, dado que para ejercer derechos de modificación es necesario contar con el código fuente del programa.

Modelos de desarrollo y distribución del software

Es posible establecer dos modelos de desarrollo y distribución del software. Ambos representan posiciones filosóficamente contrapuestas respecto de las licencias. Por un lado, el que responde al paradigma tradicional de comercialización a gran escala, privativo, generalmente cerrado y oneroso, denominado *software propietario*. Por otra parte el modelo libre - abierto, sin restricciones de

utilización al usuario ni de distribución, tales como el *software libre* y el *software de código fuente abierto*. Como puede apreciarse, la utilización de los términos *libre* y *abierto* presenta algún tipo de confusión que se expondrá más adelante.

Software propietario

El software propietario es creado por personas o empresas con finalidad comercial, generando ingresos mediante la venta masiva de la licencia. Bajo esta modalidad, se le entrega al adquirente una copia del programa ejecutable y un *contrato de licencia de usuario*. La posesión de este documento es el único elemento legal que autoriza la tenencia y uso del software.

Las licencias propietarias no responden a un modelo común pero, por lo general, establecen que se puede instalar, utilizar, tener acceso y ejecutar un número de copias del producto en igual cantidad de equipos como número de licencias adquiridas. Es decir que cada licencia individual puede instalarse en un solo equipo y si alguna de ellas debiera utilizarse en otra máquina, previamente habrá que retirar el software de la primera. Además, si el producto se instala en el servidor de una red de computadoras, se deberá adquirir una licencia individual por cada uno de los equipos donde se ejecute el software. El propietario deja perfectamente claro que al usuario no le está permitido compartir ni usar simultáneamente en diferentes equipos una misma licencia. Si el usuario vende el equipo donde está instalado el software, todos los derechos de la licencia pueden transferirse al nuevo usuario, con la sola condición de no conservar ninguna copia del producto.

Las licencias propietarias establecen que el usuario adquiere solamente la facultad de *utilizar* el programa, pero *no la propiedad* del mismo. Tampoco el usuario adquiere el derecho de modificar ni analizar el funcionamiento interno del software.

Es frecuente que las licencias indiquen que el desarrollador proporciona algún servicio de soporte técnico, que puede estar incluido o no en el costo de la licencia. En esos casos, éste se reserva el derecho de utilizar, incluso con fines comerciales, la información que recabe del usuario con motivo del servicio. En todas ellas, otorga una garantía limitada sobre el producto, haciéndose cargo de cualquier defecto que impida el normal funcionamiento, pero deslindando toda responsabilidad respecto de las pérdidas que pudiera ocasionar para el comprador el uso del programa, incluyendo los daños por fallas debidas a errores de diseño.

Siempre en el terreno de la generalización, las licencias propietarias establecen que la adquisición de una versión en particular de un software no autoriza al usuario para obtener versiones posteriores del mismo de forma gratuita. Claro que también es frecuente que el propietario libere planes de actualización de sus productos, que representen alguna ventaja económica para el adquirente que cumpla con los requisitos.

Hasta hace relativamente poco tiempo atrás, el software propietario solamente se comercializaba en cajas con indicaciones del producto conteniendo los manuales impresos, los medios magnéticos u ópticos de soporte del software y la licencia de uso. Internet y la posibilidad de utilizar el comercio electrónico, han modificado en muchos casos esta práctica, dado que es posible pagar y obtener el producto por la red. Pero, en todo caso, durante la transacción mediante esta modalidad, se entrega al usuario algún código o número de serie con valor legal que autorice el uso del programa.

Se puede percibir un desconocimiento generalizado entre los usuarios de computadoras acerca de que la utilización, copia o reproducción de productos de software no siempre es un acto lícito. Si estas acciones se llevan a cabo sin respetar los términos establecidos en la licencia - o bien sin contar con ella - se está cometiendo un delito englobado dentro de lo que se ha dado en denominar *piratería informática*.

Muchas veces los usuarios se ven involucrados en actos de piratería, sobre todo cuando se cae en supuestos de que la cotización de un equipo incluye legalmente el software y éste es preinstalado en la máquina adquirida, sin otorgar las correspondientes licencias o las mismas no se corresponden con el uso que tendrá el producto.

Otras veces se cometen estos actos por el hecho de considerar que al comprar una computadora, todo lo necesario para su utilización debería estar incluido y que el comprador tiene

derechos absolutos sobre el equipamiento, pudiendo instalar en él programas, contando o no con las licencias.

También es frecuente el uso compartido de una misma licencia en varios equipos infringiendo lo establecido en ella. A esta acción se la denomina *copia casual* o *softlifting*, y es la forma más común de piratería en ambientes corporativos.

Desde el punto de vista legal, se convierte en infractor tanto quien obtiene el software como quien lo distribuye sin la correspondiente licencia de utilización. La Ley 11.723 de la legislación Argentina establece en estos casos una pena de un mes a seis años de prisión.

Con la finalidad de hacer valer sus derechos, las principales corporaciones productoras de software han patrocinado en distintos países instituciones que accionan legalmente frente a la posesión de copias piratas de sus productos. Nació así la *Business Software Alliance* (BSA), una asociación creada para evitar la copia ilegal de los productos. En Argentina, opera en colaboración con una organización denominada *Software Legal Argentina* que tiene entre sus objetivos principales “los de promover el crecimiento continuo de la industria del software a través de programas de concientización, impulso de políticas públicas y privadas y esfuerzos para el correcto uso y aprovechamiento del software original”.

También al respecto, pero ya en el terreno técnico, se han desarrollado dispositivos físicos que se instalan en la computadora para permitir la ejecución de un software determinado. Si se obtiene la copia del software pero no el dispositivo, el programa no podrá funcionar. Además se están implementando gradualmente tecnologías controversiales - en especial en la industria del entretenimiento - denominadas *Gestión de Derechos Digitales* o DRM por sus iniciales en lengua inglesa. Se basan indistintamente en programas o dispositivos físicos que operan en un entorno informático, con la finalidad de controlar la difusión de contenidos de acuerdo con los permisos otorgados por el autor, como así también certificar la legalidad de la copia de la obra. Además, el *Trusted Computing Group* (TCG), una iniciativa liderada por las principales corporaciones de hardware y software, propone la utilización de una plataforma informática fiable (*trusted-computing*) donde las aplicaciones no puedan ser alteradas o modificadas, y que bloqueará el uso de software sin licencia autorizada.

Software libre (Free software)

Se entiende por *Software libre* al que se obtiene bajo una licencia que otorga a los usuarios la libertad de disponer de ellos de la manera que deseen, pudiendo incluso copiarlo, distribuirlo, estudiarlo o modificarlo. De modo riguroso, y de acuerdo con la definición de los autores del modelo, el software libre se refiere a cuatro libertades fundamentales de los usuarios. Otorga la libertad de: 0) usar el programa, con cualquier finalidad, 1) estudiar su funcionamiento y adaptarlo a necesidades específicas, 2) distribuir copias de la manera que se desee, y 3) mejorar el programa haciendo públicas dichas mejoras. Para que estas libertades puedan existir, es necesario disponer del código fuente del programa, como condición necesaria. En este tipo de licencia, debe entenderse *libre* en el sentido de *libertad*. Es frecuente que se lo confunda con software gratis. Esta situación es y debe ser permanentemente aclarada sobre todo si se utiliza el inglés, puesto que 'free' significa ambas cosas. Por lo tanto, puede existir algún costo asociado a su adquisición o con servicios de soporte con licencia bajo el paradigma de software libre. Esta modalidad no significa *no comercial*, por tal razón este software puede comercializarse. En realidad el software libre propone un modelo de economía solidaria, donde los ingresos son directamente proporcionales al trabajo efectivo y no un sistema de comercialización de licencias. No obstante, gran parte del software libre existente también es gratuito.

Esta filosofía tiene sus orígenes en la década de 1980 en los Estados Unidos de Norteamérica, por iniciativa de Richard Stallman y otros programadores, quienes dieron origen a la *Free Software Foundation* (FSF) y al Proyecto GNU. GNU es un acrónimo recursivo que significa “GNU is not Unix”, en alusión a ese sistema operativo con licencia propietaria. Dicho proyecto consistía en desarrollar un sistema operativo compatible con Unix, pero bajo la filosofía del software libre. En particular crearon la *licencia pública general*, GPL por sus siglas en inglés, cuyo propósito es declarar que los programas cubiertos por la misma cumplen con la definición de software libre.

Con la finalidad de evitar que se establezcan licencias de otra clase para cualquier modificación realizada sobre un producto de software libre, se desarrolló el concepto de *copyleft*, una regla que, además de garantizar las libertades fundamentales, establece que cuando se redistribuya el programa, éstas se trasladen a los receptores del mismo. Para que el copyleft sea efectivo, se debe hacer uso de las leyes que rigen la propiedad intelectual, renunciando expresamente a algunos de los derechos que de ellas se derivan.

No obstante, y fieles al principio de libertad enarbolado, existen licencias de software libre compatibles o no con la licencia GPL, que cuentan o no con protección del copyleft. Su tratamiento escapa al propósito de este artículo.

En 1992 se liberó la versión 2 de la Licencia Pública General, la cual se mantiene vigente a la fecha, y es empleada por una incontable cantidad de proyectos de software en todo el mundo. Actualmente se encuentra en proceso de discusión pública el primer borrador de la versión 3.

Software de código fuente abierto (Open source software)

Preocupados por la resistencia hacia el software libre por parte de la comunidad empresarial, causada principalmente por las implicancias de considerar posiciones en extremo libertarias y la ambigüedad que presenta para personas de habla inglesa el término *free*, algunos miembros de la comunidad del software libre - interesados en la producción de software con disponibilidad del código fuente pero con menor compromiso ideológico frente al propietario - acuñaron el término *Open Source Software*. En español significa *Software de Fuente Abierta*, pero atento a su significado, quizá resulte más adecuado traducirlo como *Software de Código Fuente Abierto*. Con la finalidad de promover este modelo, en 1998 se creó la *Open Source Initiative* (OSI) que cuenta entre sus referentes a Eric S. Raymond y Bruce Perens. Esta organización está dedicada a gerenciar y promover la *Open Source Definition* (OSD), una definición minuciosa y pragmática que determina las diez condiciones que debe cumplir una licencia de software para ser considerada de código abierto. La OSD original ha sufrido variaciones a lo largo del tiempo, tanto en sus criterios como en los comentarios adjuntos. En la actualidad se encuentra disponible la versión 1.9 que hace referencia a los siguientes ítems:

1. *Redistribución libre*: no pueden establecerse restricciones para que el software pueda ser vendido o regalado, aún cuando forme parte de una distribución que contenga programas de distintas fuentes. Tampoco debe requerir el pago de los derechos de autor u otra tasa por su venta.

2. *Código fuente*: el programa debe incluir el código fuente, y tiene que permitir la distribución tanto en código fuente como compilado.

3. *Trabajos derivados*: la licencia debe permitir modificaciones o trabajos derivados, y que éstos se distribuyan bajo las mismas condiciones de la licencia del software original.

4. *Integridad del código fuente del autor*: la licencia puede impedir que se distribuya el código fuente modificado de una aplicación, solamente si permite que se distribuyan *parches* junto al código original, con el propósito de poder modificar el programa al momento de la compilación.

5. *No a la discriminación de personas o grupos*: ninguna licencia de código fuente abierto puede impedir por sí misma a personas o grupos la participación en el proceso.

6. *No discriminar campos de aplicación*: la licencia no debe restringir la utilización del programa en un campo laboral específico.

7. *Distribución de la licencia*: deben aplicarse los mismos derechos a todo el que reciba el programa sin la necesidad de ejecutar una licencia adicional.

8. *La licencia no debe ser específica de un producto*: los derechos aplicados a un programa no deben depender de la distribución particular de software de la que forma parte. Si el programa se extrae de esa distribución y es usado o distribuido de acuerdo con los términos de la licencia del programa, quienes lo reciben deben tener los mismos derechos.

9. *La licencia no debe restringir a otro software*: ésta no debe obligar que todos los demás programas distribuidos en el mismo medio deban ser software de código fuente abierto.

10. *La licencia debe ser tecnológicamente neutral*: ninguna de sus estipulaciones debe estar basada en un tipo determinado de tecnología o estilo de interfaz.

La comunidad FLOSS

La comparación de las definiciones de software libre según la FSF y la de software de código abierto establecida por la OSD, permite determinar que el software libre cumple con la definición de software de código abierto, mientras que la situación inversa no está garantizada. El propio Stallman afirma que: “el software de código fuente abierto describe una categoría de licencias de software casi, *pero no completamente*, igual que software libre” fundamentalmente porque la definición de software de código abierto posibilita la inclusión de otros tipos de software, incluso programas con licencia propietaria. También la OSD permite la modificación de una aplicación de código abierto sin publicar el código fuente de las modificaciones, mientras que de acuerdo con la definición de software libre es obligatorio hacerlo.

No obstante, las coincidencias entre ambos modelos son más que las discrepancias - siendo estas últimas principalmente de carácter filosófico - a tal punto que es frecuente encontrar en la literatura términos acuñados expresamente para describir la unión de estos conceptos y referirse a la comunidad como un todo. Así, han aparecido expresiones tales como *free / open source software* (F/OSS), *open source software / free software* (OSS/FS) o bien *free - libre / open source software* (FLOSS). En lo que resta del artículo se utilizará la expresión *software libre - de código abierto* para hacer referencia ambos.

Características del Software propietario

El software propietario se encuentra altamente difundido y cubre todas las posibilidades técnicas existentes al momento, tan es así que en algunos casos muy específicos, solamente existen soluciones bajo este paradigma.

Muchas de las piezas de software propietarias constituyen un estándar de facto, a tal extremo que los usuarios finales se encuentran familiarizados con ellas y utilizan el nombre comercial de un producto para referenciar un sistema operativo o un determinado tipo de aplicación.

Como el software propietario responde fundamentalmente a los intereses comerciales de quien lo desarrolla, es necesario adquirir las correspondientes licencias para usarlo dentro de la legalidad. Además, y por el mismo motivo, periódicamente se liberan al mercado nuevas versiones de los productos, que corrigen los defectos encontrados e incorporan nuevas funcionalidades. Son justamente estas nuevas prestaciones las que, tarde o temprano, obligan al usuario a migrar a las versiones actualizadas, con el consiguiente costo asociado. También debe tenerse en cuenta que pasado cierto tiempo, la empresa comercializadora deja de dar soporte técnico al producto. Esta es también una frecuente causa de migración.

Debe tenerse en cuenta, en aquellos casos donde no se dispone del código fuente de estos programas, que existen situaciones de utilización potencialmente riesgosas en cuanto a errores en el diseño o vulnerabilidades.

Características del Software libre - de código abierto

El uso de software libre -de código abierto- ha sido en sus orígenes una actividad de expertos, a tal punto que existe una importantísima cantidad de servidores y equipos destinados a las comunicaciones e Internet que funcionan con productos de estas características. Pero últimamente se ha popularizado gracias a la avalancha de aplicaciones destinadas al usuario final no informático. Los más de veinte años de desarrollo del modelo y la producción cooperativa de programas lo han ido

acercando a éste, de tal manera que actualmente dispone de alternativas a los sistemas operativos y cuenta con un amplio conjunto de aplicaciones, tanto o más funcionales que sus equivalentes bajo licencia propietaria. Las versiones actuales de estos productos poseen un elevado grado de amigabilidad, permitiendo una instalación relativamente sencilla, fácil utilización y compatibilidad con otras herramientas.

Las ventajas más importantes que presentan son la adaptabilidad y seguridad, que se desprenden de la libre disponibilidad del código fuente y el importante ahorro en costos de licencias de utilización. Como desventajas se puede señalar que la curva de aprendizaje de alguno de estos productos generalmente se extiende algo más en el tiempo que su contraparte propietaria y que a la fecha algunos dispositivos físicos no funcionan adecuadamente bajo software de este modelo.

El software libre -de código abierto- produjo una verdadera revolución en el mundo de la informática. Las grandes empresas desarrolladoras de software propietario se están sumando a esa corriente: en algunos casos entregan de forma gratuita sus productos, en otros ponen a disposición de la comunidad el código fuente de sus aplicaciones, compatibilizando a veces sus licencias con las de la comunidad FLOSS.

CONCLUSIONES

Actualmente ambos modelos de licencias se encuentran consolidados y coexisten a tal punto que es posible encontrar aplicaciones libres -de código abierto que corren sobre sistemas operativos propietarios como también aplicaciones propietarias que funcionan sobre sistemas operativos libres- de código abierto. Esta situación redundará en beneficios para el usuario final, que dispone de una amplia oferta de productos y servicios relacionados para ambos modelos, pudiendo adoptar las soluciones técnicas que mejor se ajusten a sus necesidades o convicciones.

Recibido: 25/10/05

Evaluado: 15/02/06

BIBLIOGRAFÍA

Fernández Delpech, Horacio *Protección jurídica del software* Buenos Aires: Abeledo-Perrot, 2000.

Carranza Torres, Martín *Problemática jurídica del software libre* 1ª ed. Buenos Aires: Lexis Nexis, 2004.

Moisset de Espanés, Luis Hiruela de Fernández, María del Pilar *Protección jurídica del software* Academia Nacional de Derecho y Ciencias Sociales de Córdoba, 2000 [en línea] Requiere lector de archivos en formato pdf <<http://www.acader.unc.edu.ar/artsoftware.pdf>> [consulta: Diciembre 2005].

Free Software Foundation, Inc. *Philosophy of the GNU Project*, Enero 2006 [en línea]. <<http://www.gnu.org/philosophy/philosophy.html>> Traducción al español disponible [Consulta: enero 2006].

Open Source Initiative. *The Open Source Definition*, 2006 [en línea] <<http://www.opensource.org/docs/definition.php>> [Consulta: febrero 2006]

Microsoft Corporation. *Licencias de software y modelos de desarrollo*, 2004 [en línea] <<http://www.microsoft.com/spain/sharesource/Articles/default.msp>> [Consulta: Diciembre 2005]

Microsoft Corporation. *Contrato de Licencia para el Usuario Final de Software Microsoft*. Microsoft Office 2000 Professional. USA: Microsoft, 2000.

Microsoft Corporation. *Contrato de Licencia para el Usuario Final de Software Microsoft*. Microsoft Windows 98. España: Microsoft, 1998.

Instituto Nacional de la Propiedad Industrial República Argentina, 2005 [en línea] Buenos Aires <<http://www.inpi.gov.ar>> [Consulta: Diciembre 2005]

Fernández Delpech, Horacio *Propiedad Intelectual - Software. Legislación Argentina y Extranjera - Doctrina* 2005 [en línea] Buenos Aires <http://www.hfernandezdelpech.com.ar/Propiedad_intelec.htm> [Consulta: enero 2006]

Software Legal Argentina, [sin fecha] [en línea] <<http://www.softwarelegal.org.ar/HTML/index.asp>> [Consulta: Diciembre 2005]

Organización Mundial de la propiedad Intelectual *Tratados administrados por la OMPI* [marzo 2002] [en línea] <<http://www.wipo.int/treaties/es/>> [Consulta: enero 2006]