

Propiedad intelectual del Software - Licencias y Patentes

Castaño Ezequiel Leonardo^{a,1}, Mulassi Matías^{a,1}

^aEstudiante de la cátedra Legislación, Comisión 403 de Ingeniería en Sistemas de Información de la Universidad Tecnológica Nacional Facultad Regional Rosario

Enunciado

En este documento se detallan las distintas licencias de software (privativas, comerciales, libres y de dominio público), las diferencias entre las más usadas, las leyes del contexto local con respecto a la propiedad intelectual del software y lo que ocurre cuando un software o un fragmento de código no tiene una licencia explícita. Es de suma importancia que el desarrollador de software esté al tanto de esta información ya que a nivel legal puede ser pasible de una medida si no tuvo en cuenta las limitaciones que éstas imponen.

1. Introducción

En la mayoría de los países, el software no es considerado una invención y por lo tanto no puede patentarse, por lo tanto cae en el área de la Propiedad Intelectual (PI). La propiedad intelectual es tratada en múltiples tratados internacionales, Argentina actualmente es firmante de 9 de esos tratados y de ellos 8 están en vigor (Ver Apéndice 2 para más información).

1.1. Términos específicos

En este documento se utiliza vocabulario técnico del área informática por lo tanto, en esta sección se detallan las definiciones pertinentes.

- **Freeware:** Software que puede usarse gratuitamente.
- **Shareware:** Software que puede usarse de manera limitada, ya sea por tiempo limitado y/o con una cantidad de funcionalidades reducidas.
- **Código Objeto:** código ejecutable, es el que utiliza el usuario final
- **Código Fuente:** código escrito por el programador, con éste se genera el código objeto
- **Software de Código Abierto (Open Source):** Software en el cual se tiene acceso al código fuente, aunque este acceso puede no ser gratuito.
- **Software de Dominio Público:** Software sin derechos de atribución y que puede usarse libremente por cualquier individuo, todo software cuyos derechos de autor hayan caducado entra en esta categoría (la duración de los derechos de autor depende de cada país).
- **Software Comercial:** es todo software que se compra/vende, no implica que sea propietario.

- **Software propietario o privativo:** es todo software donde no se le da acceso al usuario final al código fuente, no implica que sea comercial

Si bien en la Ingeniería del Software se define como tal al conjunto de los datos, los programas informáticos y las documentaciones, en el contexto de este trabajo, se utilizará el término software para referirse únicamente al programa informático ya sea código fuente u objeto

1.2. Libertades del software libre

A continuación se definen las libertades que un software debe tener para ser considerado Software Libre, ser considerado software libre no es una licencia en sí sino una serie de libertades que una licencia debe permitir.

1. **Libertad de Uso (Libertad 0):** El software debe poder utilizarse para lo que el usuario final desee sin restricciones, esto incluye la libertad del usuario final de utilizar el software con fines maliciosos. Esta libertad es conocida en otras áreas como No-Discriminación.
2. **Libertad de Análisis (Libertad 1):** El software puede ser estudiado y modificado por el usuario final de la forma que desee.
3. **Libertad de Distribución (Libertad 2):** El usuario final puede distribuir el software de la manera que desee y con quien desee, incluso con fines lucrativos.
4. **Libertad de Mejora (Libertad 3):** El usuario final puede modificar el software y distribuirlo con las mejoras que haya hecho de la manera que desee y con quien desee, incluso con fines lucrativos.

De estas libertades se deducen ciertas consideraciones especiales que deben ser tenidas en cuenta a la hora de desarrollar un software libre y que pueden no ser obvias en primera instancia.

- **Distribución Libre:** cualquiera debe poder distribuir el software si lo desea, esto es importante especialmente cuando se desarrolla software interno, si se utiliza una licencia de software libre, el uso exclusivamente interno viola la libertad 2 y 3 porque restringe qué usuarios tienen acceso al software.

Correos Electrónicos: castanoezequielleonardo@gmail.com
(Castaño Ezequiel Leonardo), mulassimatias@gmail.com (Mulassi Matías)

¹Este trabajo fue realizado en Septiembre de 2018 y se encuentra bajo la licencia de de Creative Commons CC BY-NC-SA 4.0, esta licencia permite a otros entremezclar, ajustar y construir a partir de su obra con fines no comerciales, siempre y cuando le reconozcan la autoría y sus nuevas creaciones estén bajo una licencia con los mismos términos.

2. PATENTES

- **Inclusión del código fuente:** el usuario final debe tener acceso al código fuente, de manera legible, entendible y no ofuscado, como se utilizó originalmente (no necesariamente gratis). Debe tenerse cuidado al desarrollar software libre y utilizarlo como servicio, al hacer esto no se le está dando la posibilidad al usuario de acceder al código fuente y por lo tanto es una violación de la libertad 1.
- **No interferencia:** El desarrollador no puede imponer sobre el usuario final una arquitectura, plataforma, etc. ya que esto viola la Libertad 2 y 3

Un ejemplo de software libre y comercial es la distribución de Linux: Red Hat Enterprise

1.3. Contexto Local

El proceso de registro de un software como PI varía según el país, en el caso de EEUU (uno de los mayores productores de software) todo software es considerado protegido por copyright en el momento de su creación, sin embargo existen beneficios especiales cuando éste es registrado. En Argentina sucede algo similar, el software está comprendido desde 1988 dentro de la Ley de Propiedad Intelectual (LPI) [1], por lo tanto, desde su creación, su autor posee los derechos. Adicionalmente en 1994 por medio de la ley 24.425 se incluye en la legislación argentina el Acuerdo sobre los Aspectos de los Derechos de Propiedad Intelectual relacionados con el Comercio (Acuerdo sobre los ADPIC [2], en inglés, TRIPS, que en su artículo 10 menciona explícitamente al software, tanto al código fuente como al código objeto. Según el art. 63 de la LPI, se suspenden los derechos de autor hasta que se haya efectuado el registro en la Dirección Nacional de Derechos de Autor (a pesar de esto, la inscripción no es constitutiva, el derecho nace junto con la creación de la obra) y supone una desventaja con respecto al software extranjero que no está supeditado a este tipo de condicionamientos. Por otro lado, este artículo entra en conflicto con el Art. 10 inciso 1 del acuerdo sobre los ADPIC ya que este inciso incorpora el mismo tratamiento para el software que establece el Convenio de Berna art 5.2 [3], el cual establece “El goce y el ejercicio de estos derechos no estarán subordinados a ninguna formalidad y ambos son independientes de la existencia de protección en el país de origen de la obra.”.

Hoy en día, el registro de software como PI en la Dirección Nacional de Derechos de Autor se realiza por medio de la Cámara de Empresas de Software y Servicios Informáticos (CESSI), que detalla 3 maneras de registrar un software como PI:

1. **Obras Inéditas:** son aquellas obras que los autores o titulares solamente utilizan en forma personal o dentro de una empresa.
2. **Obras Publicadas:** son aquellas obras que se venden, regalan, donan, distribuyen gratuitamente, etc. Este trámite contempla la inscripción de obras de software puestas en conocimiento del público.
3. **Contratos de Software** (licencias de uso, cesión de derechos y otros).

Cada una de las formas tiene su procedimiento y documentación específica que puede verse en la página de la CESSI [4]. El registro de Obra Inédita se renueva cada 3 años mientras que el de Obra Publicada se renueva cada 70 años. Los beneficios del registro según la CESSI son:

- **Seguridad:** lo que ingresa en el Registro de Derecho de Autor adquiere, mediante el acto administrativo que significa su admisión, luego de un examen, certeza de su existencia en determinada fecha, en lo que hace a su autor y contenido. Si se trata de un contrato, certeza de la fecha, contenido y partes contratantes.
- **Prueba de Auditoría:** es una presunción de autoría que otorga el Estado, con una fecha cierta de inscripción.
- **Elemento de comparación:** el registro en la Dirección Nacional de Derechos de Autor sirve de elemento de comparación en supuestos de plagio y piratería. En esos supuestos, la obra es remitida al Poder Judicial para su valoración.
- **Protección del Usuario de buena fe:** se presume autor de la Obra el que figura como tal en el certificado de registro, salvo prueba en contrario. El autor que publicara la obra conforme a las constancias que obran en la Dirección Nacional de Derechos de Autor, quedaría eximido de responsabilidad penal, en el supuesto de que se presente el verdadero autor reclamando sus derechos.
- **Publicidad de las obras y contratos registrados:** función primordial de un registro es dar a conocer su contenido. La información beneficia a todo aquel que tiene interés en oponer su derecho frente a terceros y a quienes buscan cerciorarse sobre la viabilidad y legitimidad en una contratación.

2. Patentes

Las patentes son un medio por el cual los inventores pueden proteger sus obras para restringir su uso y evitar su replicación no autorizada, sin embargo en la mayoría de los países, el software no cumple con los requisitos para ser patentado, esto ocurre también en Argentina (Ley 24.481 Art. 6.c, Ley de patentes de invención y modelos de utilidad [5]). Como consecuencia, la protección del software contra el plagio, el uso y la reproducción no autorizada recae en el área de los derechos de autor. Estos derechos, a diferencia de las patentes, duran toda la vida del autor más 70 años luego de la muerte (Ley 11.723 Art. 5 [6]), esta duración es válida únicamente para Argentina y puede variar en otros países. En cambio las patentes duran 20 años desde la fecha de presentación de la solicitud (Ley 24.481 Art. 35). En el caso de los derechos de autor, nacen cuando se crea la obra aunque quedan suspendidos hasta su inscripción como se detalló en las secciones anteriores.

Otro aspecto a tener en cuenta son las limitaciones de las patentes ya que presentar una invención para patentar suele ser costoso cuando se desea exportar masivamente la invención (como sucede habitualmente con el software), debido a que debe patentarse en cada país donde se busque la protección del uso

3. LICENCIAS

de la patente. En el caso de software, donde prácticamente no existen gastos de envío y distribución, resultaría sumamente inconveniente, no obstante, los autores de software buscan patentar sus productos debido a que los derechos de autor no protegen las ideas sino la “expresión literal” (LPI Art. 1) y muchas veces el valor comercial yace en la idea.

Aún cuando el patentamiento de software no es posible, otras medidas pueden tomarse, especialmente cuando el sistema resultante es complejo y se debe a que muchas veces los sistemas que involucran software hacen uso de algoritmos, métodos, subsistemas, dispositivos y hardware especialmente diseñado que pueden cumplir los requerimientos de patentamiento, no protegiendo al software en sí mismo, sino a la infraestructura necesaria para su uso.

La Oficina Europea de Patentes ha publicado un artículo [7] con respecto a este tema y del debate que existe actualmente. Argentina firmó dos tratados internacionales en relación al uso de las patentes:

- Acuerdo sobre los Aspectos de los Derechos de Propiedad Intelectual relacionados con el Comercio (Acuerdo sobre los ADPIC o, en inglés, TRIPS): firmado en su inicio el 1 de enero de 1995.
- Tratado de Cooperación en materia de patentes [8], conocido generalmente como PCT (Patent Cooperation Treaty): firmado el 21 de diciembre de 1970 pero aún no está en vigor.

3. Licencias

Debido a la imposibilidad de patentar el software, surgió una alternativa a fines de los años 80, las licencias de software (LS). Las LS son un instrumento legal que detalla las libertades de uso, modificación y distribución que tiene el licenciatario (generalmente el usuario final) sobre el software. Un software (ya sea código objeto o fuente) que no tenga una licencia explícita se asume como con licencia de copyright y con todos sus derechos reservados, esto incluye a todo código publicado en internet salvo que se hayan cedido los derechos (como puede ocurrir al registrarse en una página por ejemplo).

Existe un caso especial que son las licencias de Creative Commons, organización que no recomienda el uso de sus licencias para software pero que puede contribuir en la licencia de productos auxiliares o de soporte.

3.1. Creative Commons

Creative Commons recomienda no usar ninguna de sus licencias para software, y se debe a que algunos aspectos como la distribución no están contemplados y, que sus licencias son incompatibles con muchas licencias libres. El único caso en el que se recomienda su uso es para la CC0 Public Domain Dedication que es una declaración de dominio público. Adicionalmente la versión 4.0 de la CC BY-SA es compatible con la GPL V3.

A pesar de que su uso no está recomendado para software, puede usarse para productos auxiliares como documentación, música y arte que use el software.

3.2. Empresas y Software privativo

Las empresas que venden software generalmente no ceden ni parcial ni totalmente sus derechos y utilizan otro mecanismo en lugar de las licencias, llamado Acuerdo de Licencia del Usuario Final (conocido por su sigla en inglés EULA, también conocido como Términos y Condiciones de Uso). Tiene forma de contrato donde el autor del software da al usuario final permisos de uso bajo condiciones estipuladas y éste acepta las limitaciones y obligaciones impuestas por el autor. Generalmente el EULA debe ser consentido explícitamente por el usuario antes de instalar el software, y se hace habitualmente mediante un texto donde está el EULA detallado y un botón de “Acepto los términos y condiciones de uso”. En contraste, las licencias no necesitan la aceptación del usuario ya que consisten en ceder los derechos de autor sin que el usuario contraiga ninguna obligación.

Ya que las empresas utilizan mayoritariamente EULAs para limitar el uso de su software y se reservan todos los derechos de autor, el principal énfasis de las LS es dar un marco legal apropiado a aquellos autores que deciden ceder total o parcialmente sus derechos de autor.

3.3. Licencias Libres

Para que una licencia sea considerada una licencia libre, debe pasar por la revisión de la Open Source Initiative [9], organización sin fines de lucro que se dedica a revisar constantemente las licencias y abordar dudas sobre el software libre y open source. Todas las licencias libres dan al usuario las 4 libertades que caracteriza al software libre, sin embargo pueden categorizarse en función de las restricciones que imponen sobre los productos derivados.

- Totalmente Permisiva (Dominio Público): en esta categoría está la licencia de dominio público, con la cual, el usuario tiene la libertad de hacer lo que desee sin ningún tipo de restricción, lo que incluye cambiar la licencia, utilizar copyright e incluso lucrar con el producto derivado.
- Permisiva: El usuario final debe hacer atribución al autor original, sin embargo no está obligado a mantener la misma licencia, pudiendo cambiarla, utilizar copyright y lucrar con el producto derivado. Algunas de estas licencias incluyen a las patentes derivadas del software, quedando reservado el derecho de patentamiento para el autor original.
- Levemente Restrictiva: El usuario final debe hacer atribución al autor original, sin embargo no está obligado a mantener la misma licencia mientras que se cumplan ciertas condiciones, las cuales tienen relación con el nivel de integración del componente, es decir, si el producto es utilizado de manera externa o anexa al sistema, actúa como una licencia permisiva y si el sistema incorpora directamente el código del producto, debe mantener la licencia original como en un escenario de Copyleft
- Totalmente Restrictiva (Copyleft): El usuario final debe mantener la licencia del autor original y si el componente utilizado forma parte de un sistema, todo el sistema debe ser licenciado con la misma licencia del autor original

3. LICENCIAS

3.4. *Licencias Libres más comunes*

A continuación se detallan las 4 licencias libres más comunes, estas abarcan más del 80 % libre que uno puede llegar a encontrar. Una lista completa de las licencias de software libre puede encontrarse en el Apéndice 1.

En primer lugar, se tienen dos licencias que fueron originadas en el contexto académico, ambas son muy simples y fáciles de entender:

- MIT: esta licencia es la más permisiva de todas (con la excepción de dominio público), permite al usuario usar el software de la manera que desee y consta únicamente de dos cláusulas, la primera es la de atribución, en la que se exige que el usuario final haga mención del autor original en todos los productos derivados y la segunda es la cláusula de no garantía, en la que el autor no se hace responsable de los fallos o problemas que pueda generar su software.
- BSD: esta licencia consta de 3 variantes, la primera contiene 2 cláusulas y es idéntica a la MIT, la segunda incorpora una tercer cláusula de No Endoso, donde se detalla que el producto no puede utilizarse para promocionar un producto derivado ni tampoco de una forma en la que de a entender que el autor apoya o está relacionado de alguna manera con el producto derivado. Por último existe una versión que incorpora una cuarta cláusula que establece que, en caso de hacerse publicidad de un producto derivado, deben mencionarse todos los autores y colaboradores que intervinieron. Esta última versión no se usa con frecuencia ya que el número de contribuidores en un software libre suele crecer con facilidad.

Por otro lado, fuera de la academia se han desarrollado licencias que también son ampliamente usadas.

- Apache 2.0: Sucesora de la ya en desuso Apache 1.1, esta licencia a diferencia de las anteriores es mucho más extensa, teniendo 9 cláusulas e incluyendo no sólo los derechos de copyright sino los de patentamiento de los posibles productos derivados así como una cláusula de terminación de los derechos sobre esas patentes. Sigue siendo una licencia permisiva y la que da más seguridad al autor original.
- Licencia Pública General de GNU [10] (GPL): Publicada originalmente por Richard Stallman mediante la Free Software Foundation [11], constituye un conjunto de 4 licencias, éstas son de las más restrictivas para el usuario final y son todas consideradas de Copyleft. Todo producto derivado de uno licenciado con GPL debe publicarse con la misma licencia (con la excepción de LGPL, ver abajo):
 - GPL V2: el usuario final es libre de hacer las modificaciones que desee, sin embargo no puede cambiar la licencia, tampoco son contemplados los casos de patentes asociadas como en la Apache 2. Adicionalmente esta licencia originaba conflictos al querer usarse con software con otras licencias.

- GPL V3: incorpora la parte de patentes a la GPL V2 y mejora la compatibilidad con otras licencias, haciendo posible combinar productos con licencias como Apache 2 y GPL V3. Sin embargo, el producto derivado debe ser publicado en su totalidad por GPL V3.
- LGPL: Similar a la GPL V3 pero con la consideración de que en ciertos contextos no se impone el uso de la misma licencia en los productos derivados. Esto ocurre cuando el software licenciado con LGPL no forma parte íntegra del producto derivado sino que es utilizado como caja negra.
- AGPL: es la licencia con más restricciones, todo uso directo o indirecto de un software con esta licencia obliga a que el producto derivado sea publicado con la licencia AGPL.

3.5. *Compatibilidad*

Como puede inferirse de las libertades y restricciones de las licencias, existen escenarios de contradicción y conflicto, por lo que se debe evaluar la compatibilidad de las licencias a la hora de integrar diversos componentes de softwares con múltiples licencias. Este fue el principal motivo de la versión 3 de GPL. Se debe recordar que a pesar de que exista compatibilidad, las licencias GPL son Copyleft, por lo tanto si se usan varias licencias compatibles el producto final tiene que ser GPL independientemente de las licencias originales. Debido a los grandes problemas de compatibilidad, surgieron empresas que se dedican a auditar sistemas para detectar incompatibilidades en las licencias de los componentes. Algunos ejemplos de compatibilidades entre licencias son:

- Apache 2 no es compatible con GPL V2 pero si con la V3
- La BSD-4 no es compatible con ninguna GPL

3.6. *Contribuciones*

En un contexto empresarial, donde se desarrolla software, la empresa tiene los derechos de autor sobre el producto que vende y no los usuarios individuales que lo crearon, lo que puede lograrse de varias maneras, las más sencillas a considerar son cláusulas especiales en los contratos de empleo o locaciones de servicio o el LPI Art. 4.d - Incorporado por la Ley 25.036 [12].

Ahora bien, cuando se habla de software libre (donde múltiples usuarios pueden contribuir en el código), haya o no una organización de por medio, las contribuciones de código no se realizan bajo un contrato de empleo o de locación de servicio sino que son hechas por individuos particulares y esas contribuciones individuales están sujetas a sus respectivos derechos de autor, entonces la persona u organización que desarrolle un software libre debe considerar como serán manejadas las contribuciones y el impacto de los derechos de autor.

Por este motivo han surgido dos tipos de acuerdos que pueden existir entre el contribuidor y el desarrollador principal del software.

4. APLICACIONES

- Acuerdo de licencia de contribución (CLA por su sigla en inglés): con un CLA el contribuidor retiene formalmente el copyright de su contribución, pero da al proyecto un no-exclusivo, perpetuo, irrevocable, gratuito, libre de regalías, reproducible, preparado para hacer obras derivadas, derecho a usar, sublicenciar y distribuir la contribución y sus productos derivados. Es decir, se retiene el derecho de copyright pero se promete no ejercer la mayoría de los poderes que el copyright normalmente implicaría, facilitando al proyecto aceptar la contribución. Adicionalmente no se pueden tomar acciones legales contra el contribuidor si su contribución causara inconvenientes o problemas.
- Acuerdo de Transferecia de contribución (CAA/CTA por sus siglas en inglés): Es un acuerdo donde el contribuidor cede todos los derechos de autor sujetos a su contribución, en este caso el receptor puede tomar acciones legales contra el contribuidor si su contribución produjo perjuicio o problemas. Esta es la única manera en la que un individuo u organización puede tener todos los derechos de autor sobre un producto cuando no existe un contrato laboral o locación de servicio y cuando no aplica el Art. 4.d de la LPI

4. Aplicaciones

4.1. *Uso de software libre en un desarrollo*

En el caso de un desarrollo de software, el trabajo en equipo resulta fundamental y muchas veces varias de las funcionalidades del sistema pueden implementarse mediante la inclusión de componentes de software libre en el proyecto, sin embargo, es necesario ser sumamente cuidadoso con los componentes que se añaden al sistema y de las licencias de éstos, ya que puede traer problemas legales en el futuro. El método más sencillo para evitar estos problemas, es que cada proyecto debe tener una lista de licencias y el líder del proyecto debe especificar si se puede o no usar cada una de éstas. Esta lista debe hacerse en función de las ya conocidas compatibilidades entre licencias o con la ayuda de un asesor legal especializado en propiedad intelectual del software.

4.2. *Lucrar con software libre*

Asimismo, es importante destacar que las licencias de software libre no son exclusivas de organizaciones sin fines de lucro y que de hecho, cada vez más empresas utilizan y desarrollan software libre. A continuación se detallan brevemente algunos de los métodos lucrativos comúnmente utilizados en el software libre.

- Brindar el software como servicio (SaaS): La empresa desarrolla un software libre que cualquier usuario puede utilizar pero la empresa también provee el mismo software como servicio, evitando que el usuario tenga que implementarlo. Un ejemplo de esto es MongoDB Atlas, WordPress y Drupal
- Sponsor y Publicidad: Una empresa apoya financieramente un proyecto. Ejemplo: NumFocus

- Crowdsourcing: Se realiza una campaña de recolecta de fondos para financiar el desarrollo de un software.
- Añadir mejoras (Freemium): Probablemente la modalidad más popular, en ella, la empresa desarrolla un software con ciertas características con una licencia libre y añade una serie de módulos o mejoras bajo una licencia privativa de forma tal de que los usuarios pueden usar libremente la versión básica y aquellos que lo desean pueden comprar las mejoras. En este caso no se habla de un Freeware o un Shareware sino que la versión básica es software libre.
- Vender Garantías: Como se vio en las secciones anteriores, ninguna de las licencias da garantías con respecto a la calidad del software y la ausencia de errores o defectos, por lo que esa garantía puede venderse.
- Soporte y mantenimiento: La empresa desarrolla el software libre y aquellos que necesiten asistencia deberán pagar así como los que pidan características personalizadas. Un ejemplo de esto es Red Hat Linux
- Documentación: La empresa desarrolla el software pero no provee la documentación para usar el mismo de manera gratuita. En este contexto la documentación no necesariamente hace referencia a la propia del software sino a libros, artículos y cualquier material escrito que pudiera proveerse.
- Licencia Dual: La empresa desarrolla un software que sale al mercado con dos licencias posibles que se asignan en función del tipo de consumidor, lo que permite, por ejemplo, entregar el software con licencia GPL para aquellos usuarios que lo usan sin fines de lucro pero utilizar una licencia privativa si el software es empleado comercialmente. Un ejemplo de esto es Moodle, MySQL y Qt.
- Implementación: La empresa desarrolla el software libre pero cobra por su adaptación, instalación y despliegue en el cliente o usuario final.
- Entrenamiento: La empresa desarrolla el software libre pero cobra por dar entrenamiento para utilizarlo, esto es muy común en el sistema operativo Linux.
- Certificaciones: La empresa cobra exámenes de certificación a las personas que estén interesadas, la misma consiste en que la empresa garantiza que la persona tiene los conocimientos que necesita. Esto es común en el sistema operativo Linux.
- Donaciones: La empresa desarrolla el software libre y recibe donaciones de los individuos interesados.

Estas metodologías no son excluyentes y pueden combinarse para ajustarse al modelo de negocio que la empresa necesite, excede el alcance de este trabajo evaluar detalladamente cada una de estas alternativas. Dependiendo del país, algunas pueden ser utilizadas por organizaciones sin fines de lucro y otras no, y así mismo, algunas pueden estar exentas de impuestos y otras no, debe evaluarse cada posibilidad en el contexto específico.

5. RECURSOS ADICIONALES

4.3. Ahorrar dinero y problemas legales

La mayoría de los problemas legales relacionados con el software es debido a la piratería (una manera ilícita de evitar comprar las licencias de software comercial) y de la violación de las licencias y términos y condiciones de uso (EULA). Para solucionar este problema, se puede utilizar software libre en lugar del software comercial/privado que actualmente se usa. Esto implica un proceso de adaptación y transición pero trae consigo una reducción de costos en las licencias (el software libre es gratuito) y evitar problemas legales con las licencias (el software libre no tiene EULA y las restricciones de las licencias sólo aplican para productos derivados y no para el uso.). La Open Source Initiative ha compilado una serie de alternativas [13] para los softwares comerciales/privados más comunes entre los que se pueden mencionar Microsoft Word/Excel, Microsoft Project, Trello, Slack, Adobe Photoshop entre muchos otros.

5. Recursos Adicionales

Adicionalmente a la información presentada en este trabajo, se proporciona una lista de recursos de interés como material complementario para profundizar en los temas tratados.

- Preguntas frecuentes con respecto al software libre [14].
- Principales Organizaciones que apoyan el movimiento del software libre [15].
- Lista de softwares libres bajo la modalidad “self-hosted” [16].
- Empresas que ofrecen su software de manera gratuita para proyecto de software libre [17].
- Lista de herramientas que implementan la filosofía del software libre en distintas áreas de la sociedad como ética, privacidad, salud, entre otros [18].
- Compilado de material basado en la filosofía del software libre para aprender de manera autodidacta todo el currículum de carreras de Ciencias de la Computación [19].
- Videoconferencia: The Impact of Leveraging Open Source by Jeff Strauss[20].

Referencias

- [1] Ministerio de justicia y derechos humanos. Presidencia de la Nación (1933). “Ley de Propiedad Intelectual” (LPI - Ley 11.723: Art. 1). Argentina, pp. Disponible en: <http://servicios.infoleg.gob.ar/infolegInternet/anexos/40000-44999/42755/textact.htm>.
- [2] Organización Mundial del Comercio (OMC) (1994). Organización Mundial del Comercio (OMC) - “Acuerdo sobre los Aspectos de los Derechos de Propiedad Intelectual relacionados con el Comercio” (Acuerdo sobre los ADPIC) (1994). Marrakech, pp. Disponible en: http://www.wipo.int/wipolex/es/other_treaties/details.jsp?group_id=22&treaty_id=231.
- [3] Tratado Internacional, “Convenio de Berna para la Protección de las Obras Literarias y Artísticas”, Suiza, 1977. Disponible en: <http://www.wipo.int/treaties/es/ip/berne/> [Accedido el 25 de septiembre de 2018].
- [4] CESSI. “Registro de software”. [online] Disponible en: <https://www.cessi.org.ar/mostrar-registro-de-software-88> [Accedido 25 Sep. 2018].
- [5] Ministerio de justicia y derechos humanos. Presidencia de la Nación (1995). “Ley de patentes de invención y modelos de utilidad” (Ley 24.481) Art. 6.c. Argentina, pp. Disponible en: <http://servicios.infoleg.gob.ar/infolegInternet/anexos/25000-29999/27289/norma.htm>.
- [6] Ministerio de justicia y derechos humanos. Presidencia de la Nación (1933). “Regimen legal de la propiedad intelectual” - Ley 11.723 -. Argentina, pp. Disponible en: <http://servicios.infoleg.gob.ar/infolegInternet/anexos/40000-44999/42755/textact.htm>.
- [7] EPO. “Patents for Software?”. [online] Disponible en: [urlhttps://www.epo.org/news-issues/issues/software.html](https://www.epo.org/news-issues/issues/software.html) [Accedido el 25 de Septiembre de 2018].
- [8] Organización mundial de la propiedad intelectual (OMPI) (n.d.). “Tratado de Cooperación en materia de Patentes” (PCT). Washington , EE.UU, p. Disponible en: <http://www.wipo.int/treaties/es/registration/pct/>.
- [9] Open Source Initiative. (1998). “Revisión para licencias libres”. [online] Disponible en: <https://opensource.org/> [Accedido el 25 de septiembre de 2018].
- [10] R. Stallman, “Licencia Publica general de GNU”, 2018 Disponible en [urlhttps://www.fsf.org/licensing/](https://www.fsf.org/licensing/) [Accedido el 25 de septiembre de 2018].
- [11] R. Stallman, “Free software foundation”. San Petersburgo, Florida, Estados Unidos, 2003.
- [12] Ministerio de justicia y derechos humanos. Presidencia de la Nación, “Ley de la Propiedad Intelectual”- Ley 25.036, Argentina, 1998. Disponible en: <http://servicios.infoleg.gob.ar/infolegInternet/anexos/50000-54999/54178/norma.htm> [Accedido el 25 de septiembre de 2018].
- [13] “Open Source alternatives”, OpenSource, 2018. [Online]. Disponible en: [urlhttps://opensource.com/alternatives](https://opensource.com/alternatives). [Accedido el 25 de septiembre de 2018].
- [14] “Frequently Answered Questions — Open Source Initiative”, Opensource.org. [Online]. Disponible en: <https://opensource.org/faq>. [Accedido el 25 de septiembre de 2018].
- [15] “Open source organizations”, Opensource.com, 2018. [Online]. Disponible en: <https://opensource.com/resources/organizations>. [Accedido el 25 de septiembre de 2018].
- [16] “Lista de software libres bajo la modalidad “self-hosted””, GitHub, 2018. [Online]. Disponible en: <https://github.com/Kickball/awesome-selfhosted>. [Accedido el 25 de septiembre de 2018].
- [17] “Companies that offer their software for free software project”, GitHub, 2018. [Online]. Disponible en: <https://github.com/zachflower/awesome-open-source-supporters#readme>. [Accedido el 25 de septiembre de 2018].
- [18] “List of tools that implement the philosophy of free software in different areas of society”, GitHub, 2018. [Online]. Disponible en: <https://github.com/engagingspaces/awesome-humane-tech#readme>. [Accedido el 25 de septiembre de 2018].
- [19] “Philosophy of Free Software”, GitHub, 2018. [Online]. Disponible en: <https://github.com/ossu/computer-science#readme>. [Accedido el 25 de septiembre de 2018].
- [20] Strauss, J. (2017). “The Impact of Leveraging Open Source”. [video] Disponible en: <https://www.youtube.com/watch?v=jwrSMj-Prfw>. [Accedido el 25 de septiembre de 2018].

Apendice 1: Listado de Licencias

A continuación se detalla una lista completa de todas las licencias de software libre aprobadas por el Open Source Initiative.

Fuente: <https://opensource.org/licenses/category>

Licencias que son populares y que son ampliamente usadas o con una comunidad presente

- Apache License 2.0 (Apache-2.0)
- BSD

REFERENCIAS

- 3-clause BSD license (BSD-3-Clause)
- 2-clause BSD license (BSD-2-Clause)
- GNU
 - GNU General Public License (GPL)
 - GNU Lesser General Public License (LGPL)
- MIT license (MIT)
- Mozilla Public License 2.0 (MPL-2.0)
- Common Development and Distribution License version 1.0 (CDDL-1.0)
- Eclipse Public License version 2.0

Licencias internacionales

- Licence Libre du Québec – Permissive (LiLiQ-P) version 1.1 (LiLiQ-P-1.1)
- Licence Libre du Québec – Réciprocité (LiLiQ-R) version 1.1 (LiLiQ-R-1.1)
- Licence Libre du Québec – Réciprocité forte (LiLiQ-R+) version 1.1 (LiLiQ-Rplus-1.1)

Licencias de propósitos especiales

- BSD+Patent (BSD-2-Clause-Patent)
- Educational Community License, Version 2.0 (ECL-2.0)
- IPA Font License (IPA)
- NASA Open Source Agreement 1.3 (NASA-1.3)
- OSET Public License version 2.1 (OSET-PL-2.1)
- SIL Open Font License 1.1 (OFL-1.1)
- Upstream Compatibility License v1.0

Licencias miseláneas

- Adaptive Public License (APL-1.0)
- Artistic license 2.0 (Artistic-2.0)
- Free Public License 1.0.0 (0BSD)
- Open Software License (OSL-3.0)
- Q Public License (QPL-1.0)
- Universal Permissive License (UPL)
- zlib/libpng license (Zlib)

Licencias que son redundantes con las más populares

- Academic Free License (AFL-3.0)
- Attribution Assurance Licenses (AAL)
- Eiffel Forum License V2.0 (EFL-2.0)
- Fair License (Fair)
- Historical Permission Notice and Disclaimer (HPND)
- Lucent Public License Version 1.02 (LPL-1.02)
- The PostgreSQL License (PostgreSQL)
- University of Illinois/NCSA Open Source License (NC-SA)
- X.Net License (Xnet)

Licencias no reutilizables

- Apple Public Source License (APSL-2.0)
- Computer Associates Trusted Open Source License 1.1 (CATOSL-1.1)
- CUA Office Public License Version 1.0 (CUA-OPL-1.0)
- eCos License version 2.0
- EU DataGrid Software License (EUDatagrid)
- Entessa Public License (Entessa)
- Frameworkx License (Frameworkx-1.0)
- IBM Public License (IPL-1.0)
- LaTeX Project Public License (LPPL-1.3c)
- Motosoto License (Motosoto)
- Multics License (Multics)
- Naumen Public License (Naumen)
- Nethack General Public License (NGPL)
- Nokia Open Source License (Nokia)
- OCLC Research Public License 2.0 (OCLC-2.0)
- PHP License (PHP-3.0)
- Python License (Python-2.0)
- CNRI Python license (CNRI-Python) (CNRI portion of Python License)
- RealNetworks Public Source License V1.0 (RPSL-1.0)
- Ricoh Source Code Public License (RSCPL)
- Sleepycat License (Sleepycat)
- Sun Public License (SPL-1.0)
- Sybase Open Watcom Public License 1.0 (Watcom-1.0)
- Vovida Software License v. 1.0 (VSL-1.0)
- W3C License (W3C)
- wxWindows Library License (WXwindows)
- Zope Public License (ZPL-2.0)

Licencias suspendidas

- Apache Software License 1.1 (Apache-1.1)
- Artistic license 1.0 (Artistic-1.0)
- Common Public License 1.0 (CPL-1.0)
- Eclipse Public License 1.0 (EPL-1.0)
- Educational Community License, Version 1.0 (ECL-1.0)
- Eiffel Forum License V1.0 (EFL-1.0)
- Lucent Public License (“Plan9”), version 1.0 (LPL-1.0)
- Mozilla Public License 1.0 (MPL-1.0)
- Mozilla Public License 1.1 (MPL-1.1)
- Open Software License 1.0 (OSL-1.0)
- Open Software License 2.1 (OSL-2.1)
- Reciprocal Public License, version 1.1 (RPL-1.1)

REFERENCIAS

Licencias que han sido voluntariamente suspendidas

- Intel Open Source License (Intel)
- Jabber Open Source License
- MITRE Collaborative Virtual Workspace License (CVW)
- Sun Industry Standards Source License (SISSL)

Licencias sin categoría

- Boost Software License (BSL-1.0)
- CeCILL License 2.1 (CECILL-2.1)
- Common Public Attribution License 1.0 (CPAL-1.0)
- European Union Public License (EUPL-1.1)
- GNU Affero General Public License v3 (AGPL-3.0)
- ISC License (ISC)
- Microsoft Public License (MS-PL)
- Microsoft Reciprocal License (MS-RL)
- MirOS Licence (MirOS)
- Non-Profit Open Software License 3.0 (NPOSL-3.0)
- NTP License (NTP)
- Reciprocal Public License 1.5 (RPL-1.5)
- Simple Public License 2.0 (SimPL-2.0)
- Open Group Test Suite License (OGTSL)

Apendice 2: Tratados de PI firmados por Argentina

- Convenio de Berna para la Protección de las Obras Literarias y Artísticas. Fuente : <http://www.wipo.int/treaties/es/ip/berne/>.
- Convenio de Bruselas sobre la distribución de señales portadoras de programas transmitidas por satélite. Fuente: <http://www.wipo.int/treaties/es/ip/brussels/>
- Tratado de Marrakech para facilitar el acceso a las obras publicadas a las personas ciegas, con discapacidad visual o con otras dificultades para acceder al texto impreso. Fuente: <http://www.wipo.int/treaties/es/ip/marrakesh/>
- Tratado de Nairobi sobre la protección del Símbolo Olímpico. Fuente: <http://www.wipo.int/treaties/es/ip/nairobi/>
- Convenio de París para la Protección de la Propiedad Industrial. Fuente: <http://www.wipo.int/treaties/es/ip/paris/>
- Convenio para la protección de los productores de fonogramas contra la reproducción no autorizada de sus fonogramas. Fuente: <http://www.wipo.int/treaties/es/ip/phonograms/>
- Convención de Roma sobre la protección de los artistas intérpretes o ejecutantes, los productores de fonogramas y los organismos de radiodifusión. Fuente: <http://www.wipo.int/treaties/es/ip/rome/>
- Tratado de la OMPI sobre Derecho de Autor. Fuente: <http://www.wipo.int/treaties/es/ip/wct/>
- Tratado de la OMPI sobre Interpretación o Ejecución y Fonogramas. Fuente: <http://www.wipo.int/treaties/es/ip/wppt/>