Las licencias libres en la práctica

Malcolm Bain
Manuel Gallego
Manuel Martínez Ribas
Judit Rius

P08/M2114/00348

UOC
Universitat Oberta de Catalunya
www.uoc.edu

Índice

Introducción

En los módulos anteriores hemos presentado el marco legal del software en general y del software libre en particular, junto con un análisis y una discusión de las licencias de software propietarias y las licencias de software libres. Ahora, ya armados con el conocimiento correspondiente, queremos cerrar este módulo con una discusión sobre algunos mitos que han surgido alrededor del software libre y una reflexión sobre las consecuencias prácticas de las licencias libres (y de los aspectos legales del software libre en general) para los diferentes actores relacionados con él.

Asimismo, consideramos importante poner las licencias de software en relación con otras ramas del derecho, desde un punto de vista práctico.

Por lo tanto, en el primer apartado, desmitificaremos algunas creencias muy difundidas sobre diversos aspectos legales del software libre y las licencias asociadas. Luego, en el segundo apartado, comentaremos algunos temas y **efectos prácticos relativos a las licencias libres**, tales como la compatibilidad entre licencias, la gestión de las contribuciones a proyectos libres y las licencias duales.

En el tercer apartado, reflexionaremos sobre algunos **aspectos legales del software libre** que no hemos visto hasta ahora: la relación entre estas licencias y otras ramas del derecho, especialmente el derecho de la competencia y las licencias de patentes y marcas, y la interrelación entre licencias libres y el proceso de estandarización formal.

Asimismo, para cerrar el módulo, estudiaremos muy brevemente dos importantes áreas del derecho relacionadas con el software libre que no se han desarrollado hasta ahora: por un lado, la protección de **datos personales** y de la privacidad y, por otro, el control de la exportación de materiales y productos de **criptografía** o **cifrado**.

Objetivos

Con el aprendizaje de este módulo, los estudiantes podréis alcanzar los objetivos siguientes:

1. Comprobar la validez legal de varios mitos que se han establecido alrededor del software libre.

2. Discernir consecuencias prácticas y los aspectos relacionados con las licencias libres como son:
- la selección de una licencia libre,
- la compatibilidad entre las licencias,
- licencias duales o múltiples,
- la bifurcación o división de código (*forking*).

3. Conocer la relación entre las licencias de software libre y otras ramas del derecho, como las marcas, la competencia leal y el proceso formal de estandarización.

4. Conocer las obligaciones y los derechos relativos a la privacidad (bajo el régimen europeo) y su vínculo con el software libre, sobre todo, en el ámbito de la seguridad.

5. Conocer la regulación de productos de seguridad y su relación con la distribución de software libre.

1. Unos mitos legales... para desmitificar

Después de realizar el análisis de las licencias libres del módulo 6, iniciamos este módulo con un comentario y una aclaración sobre varios mitos o conceptos equivocados relativos a diversos aspectos legales del software libre.

Ejemplo

Hay otros "mitos" relativos a los aspectos tecnológicos o comerciales del software libre que no trataremos aquí: la falta de soporte y mantenimiento, la falta de seguridad, el riesgo de la bifurcación (*forking*), la posibilidad de introducir elementos dañinos en software libre, la carencia de modelos de negocio viables basado en software libre, etc.

Lectura recomendada

Para profundizar sobre este tema se recomienda leer "Dispelling myths about the GPL and free software", de J. Viega y B. Fleco (en bibliografía).

1) "El *copyleft* está en contra del derecho de autor"

Este mito se basa en la creencia de que el *copyleft* (y las licencias libres en general) crea un nuevo marco de derecho de la propiedad intelectual: el *copyleft* "en vez del" *copyright*. Al contrario, tal como se ha visto en los módulos 2 y 6, las licencias libres se fundamentan directamente en el derecho de propiedad intelectual vigente, ya sea el derecho de autor al estilo continental o el *copyright* anglosajón. Los autores de software libre usan los derechos establecidos por este marco legal (la exclusividad de explotar y/o autorizar la explotación de su obra) para conceder a los licenciatarios los derechos no exclusivos establecidos en las licencias libres y defender estos derechos de vulneraciones.

Caso "MySQL AB contra Progress Software"

MySQL AB defiende su titularidad en la aplicación de bases de datos MySQL y los derechos asociados. Inició un juicio contra Progress Software por violación de derechos de autor y de la licencia GPL sobre el programa MySQL. En Alemania, el Tribunal de Primera Instancia de Múnich ha considerado y apoyado ya dos veces acciones legales respecto a la licencia GPL (fundamentadas en el derecho de la propiedad intelectual) para hacer cumplir sus condiciones por empresas que las habían vulnerado (una distribución sin código fuente y sin una copia de la licencia).

Consideremos, por ejemplo, dos características importantes del software libre: las libertades de uso y las condiciones de *copyleft*.

- En relación a la primera, el marco legal permite a los titulares de una obra definir el alcance de los derechos de explotación cedidos a terceros. En lugar de restringir los usos del licenciante (como hacen la mayoría de las licencias propietarias), una licencia libre los amplía al máximo permitido. Ello no va contra los derechos de autor, sino que es un ejercicio de los mismos.

- Respecto del *copyleft*, un desarrollador puede crear y distribuir una obra derivada de software libre porque se lo permite, en ciertas condiciones, el titular de la obra original en que se basa. Si estas condiciones –por ejemplo, la de distribuir la obra derivada con la misma licencia (el *copyleft*)– no

se cumplen, se resolverá la licencia original y la distribución de la obra derivada será una violación de los derechos originales. El *copyleft* actúa legalmente como cláusula resolutoria.

Por lo tanto, no hay contradicción ni oposición entre los derechos de autor legislados y los derechos concedidos o reservados con una licencia libre. Es más, se puede argumentar que mientras una licencia libre respete las excepciones y los usos permitidos del usuario en nuestro marco legal, se ajustará más a derecho que muchas licencias propietarias.

"La licencia GPL no agrega nada al derecho de autor [como, por ejemplo, restricciones de uso]... El derecho de autor otorga a los titulares los poderes de prohibir el ejercicio de derechos de copia, modificación y distribución, derechos de los que nosotros consideramos que los usuarios deben beneficiarse. Por lo tanto, la GPL elimina las restricciones permitidas por el sistema de propiedad intelectual".

E. Moglen, "Enforcing the GNU GPL", Linux User, 12/08/2001, en línea en http://noglen.law.columbia.edu/publications/lu_12.html

2) "El software libre no tiene titulares o propietarios"

No hay nada más equivocado desde el punto de vista legal. El marco jurídico de la propiedad intelectual confiere derechos de autor automáticamente a los creadores del software. Y la única obligación –o casi– compartida por todas las licencias libres es la de mantener los avisos de titularidad de los creadores iniciales del software (el famoso *copyright notice*). Por lo tanto, siempre hay un titular de los derechos sobre el software y, en el caso de software libre, la titularidad está claramente indicada en los ficheros.

3) "Las licencias libres obligan a ceder sus derechos de autor"

Con la excepción de los derechos morales, que son intransferibles, los derechos de autor se pueden ceder, pero únicamente con el consentimiento explícito del titular. Las licencias libres son "no exclusivas" y no pueden "quitar" la titularidad del software a sus creadores. Las licencias libres con *copyleft* sí que obligan a los licenciatarios a usar la misma licencia (no exclusiva) para la eventual distribución de cualquier modificación u obra derivada de software original con estas licencias y a publicar el correspondiente código fuente –como condición del derecho de redistribuir la modificación–, pero no los obligan a "ceder el software" (o sus derechos sobre él).

4) "No se puede hacer un uso comercial del software libre"

Otra creencia equivocada: como hemos visto, no hay límites sobre el **uso** del software libre (libertad 0); solamente, a veces, se imponen condiciones a su **modificación y distribución** posteriores. Las licencias libres no afectan a los usuarios finales.

5) "El software libre y el software propietario son incompatibles"

Otro mito es que el software libre es incompatible con el software propietario si se ejecutan en un mismo sistema o plataforma informática. Si esto fuera cierto, ninguna aplicación propietaria, como las bases de datos de Oracle, podría ejecutarse sobre GNU/Linux, OpenBSD o los servidores web Apache. Y viceversa, aplicaciones libres como MySQL no podrían ejecutarse sobre sistemas operativos propietarios como Solaris de Oracle o AIX de IBM. Lo que sí que puede suscitar incompatibilidades es la integración o la mezcla de software con *copyleft* y software propietario, tal como comentaremos a continuación.

6) "No se pueden integrar o mezclar software libre y software propietario"

Esta afirmación sostiene que el software libre, en general, no puede mezclarse o integrarse con software propietario en una misma aplicación sin afectarlo y, por lo tanto, sin violar sus condiciones de uso. Una manera más fuerte de expresar esto es afirmar que el software libre y el software con GPL en particular es vírico e "infecta" a otras aplicaciones: cualquier aplicación que integre software GPL se convierte en software GPL. Esta afirmación es parcialmente falsa.

- **Integración por el usuario final.** Las licencias libres no restringen los usos de un software con otras aplicaciones: la posibilidad de modificación es una condición de ser libre y no hay restricciones sobre su uso. Por ello es por lo que se ha de distribuir el código fuente con el código objeto, o ponerlo a disposición del destinatario. Sin embargo, cualquier integración de software libre (permitida por la licencia libre) puede ser considerada una modificación del software propietario integrado (si se tiene el código fuente para realizarla). Dependiendo de las restricciones contenidas en la licencia propietaria, dicha modificación puede constituir una infracción de la misma, independientemente de si el programa integrado es libre, propietario o redistribuido. Éste no es un problema del software libre, sino de la licencia de software propietario.

- **Integración por un intermediario.** Donde sí que puede haber restricciones relativas a la integración de software de distintos tipos, ya sea libre o propietario, es respecto de su distribución posterior. Las **licencias permisivas** permiten mezclar y redistribuir su software con licencia propietaria. Sin embargo, las **licencias con *copyleft*** prohíben la redistribución con licencia propietaria de la "mezcla" de software con estas licencias con software propietario, práctica que ha dado en llamarse *privatización del software libre*. Ciertas licencias libres contienen cláusulas que tratan de permitir de forma parcial esta integración, como la LGPL o la MPL, que hemos comentado en el módulo 6.

Software libre y software propietario

La relación y la integración entre software libre y software propietario depende, en gran medida, de las formas de interacción tecnológica entre las aplicaciones o los componentes de las mismas. Según los diseños y los lenguajes de programación, se deberán considerar la compilación, la interpretación, la simple ejecución, las llamadas a funciones, rutinas o bibliotecas, los vínculos estáticos y dinámicos y las interfaces API, entre otros. En entornos distribuidos estas relaciones se complican, por ejemplo, con componentes CGI, ASP y otras formas de interacción con programas a distancia.

En algunos casos, no se considerará "modificado" el programa inicial y, por lo tanto, en caso de software con GPL no se aplicaría el copyleft sobre la distribución posterior. En otros, esta interacción o integración implicará una modificación permitida por el autor original del software libre en las condiciones de la licencia: sin restricciones en el caso de software con la licencia BSD o con copyleft en el caso de la GPL.

La comprensión de estas relaciones es fundamental para analizar una propuesta técnica y decidir sobre las licencias posibles o los componentes libres con los que puede interactuar o integrarse un programa.

7) "Todo el software libre es igual" (en los términos de la GPL)

Hay variaciones sustanciales entre las más de sesenta licencias libres reconocidas por la OSI para formar un criterio de lo que se considera "libre". Se debe ser mucho más cuidadoso en el uso del término *software libre*, y distinguir entre licencias libres en general, licencias con *copyleft* y licencias que no son ni libres ni abiertas. Es importante manejar con claridad los términos *fuentes abiertas*, *persistencia* o *reciprocidad* y *copyleft*, característicos de estas licencias libres.

8) "Las licencias libres obligan a publicar sus modificaciones particulares"

Ésta es una de las ideas falsas más propagadas sobre el funcionamiento de las licencias libres. Distingamos la posición de los usuarios finales y de los intermediarios (desarrolladores de programas para terceros):

- **Usuarios finales.** La mayoría de las licencias libres no obliga a los usuarios ni a distribuir sus modificaciones o adaptaciones de software libre (obras derivadas, en lenguaje legal), ni a publicarlas o a contribuir con ellas al desarrollo de la aplicación modificada. Algunas licencias requieren esto último, en casos particulares, sólo en relación con correcciones o modificaciones del código central o núcleo del programa. Como veremos, estas obligaciones no se aplican a elementos adicionales agregados al núcleo ni a cualquier extensión de la aplicación. Por lo tanto, los usuarios finales no tienen que publicar sus obras basadas en software libre.

- **Los profesionales y las empresas que desarrollan programas.** Las personas que realizan desarrollos para clientes tampoco tienen que distribuir al público (o a los autores originales) cualquier modificación de un software libre. Lo que sí que tienen que hacer es respetar las licencias libres originales, muchas de las cuales obligan a proveer del código fuente a los usuarios o clientes destinatarios o, si sólo se distribuye el código objeto, a ofrecer el código fuente a cualquier tercero (la GPLv2) o al destinatario (la MPL,

GPLv3) durante cierto período. Éste es uno de los requisitos para utilizar software libre con *copyleft*.

La licencia Apple Public

La licencia Apple Public License 1.x obligaba a remitir a Apple cualquier modificación del programa original, y ésta era una de las razones por las que no se consideraba una licencia libre.

9) "El software libre no tiene responsables ni garantías"

Hay que admitir que esto puede ser cierto, con las licencias actuales de software libre, sobre todo en condiciones de distribución gratuita del software. Sin embargo, hay dudas legales sobre la efectividad de las cláusulas de exclusión de garantías y limitaciones de responsabilidad, que no serían válidas ante consumidores, por lo menos. Esto ya se ha comentado con respecto al marco jurídico en la Unión Europea en los módulos 4 y 5.

El mito, en realidad, consiste en pensar que las licencias propietarias aceptan mayor nivel de responsabilidad, ya que muchas de ellas también intentan en términos muy similares limitar la responsabilidad del licenciante (autor o distribuidor).

Con los sistemas de distribución virtual en Internet se podría argumentar también que es difícil identificar a los licenciantes y con ello reclamar alguna indemnización. Muchos sitios de distribución de software libre, como Sourceforge, no son los titulares licenciantes, ni siquiera los distribuidores "oficiales". No obstante, en algunos casos, como en el de la FSF o en negocios basados en la distribución de paquetes de software libre como Red Hat o Suse (Novell), hay una entidad legal identificable contra quien se podría intentar una acción por responsabilidades si fuera necesario. Además, la obligación de mantener el aviso de autoría (*copyright notice*) permite identificar a los titulares de cualquier componente que pudiera ser deficiente, aunque éstos no sean necesariamente los que han distribuido el programa al perjudicado.

El mismo argumento se aplica a las garantías. Las licencias libres en sí mismas no ofrecen garantías, pero tampoco son de mucha utilidad las de las licencias propietarias, que tratan de limitar la garantía contractual, por ejemplo, a la devolución del precio de compra en caso de una avería identificada dentro de un límite de noventa días. Por un lado, hay que considerar las garantías obligatorias por ley, que se aplican tanto al software libre como al software propietario. Por otro lado, las licencias libres permiten a los distribuidores de software libre agregar cláusulas de garantía (con contraprestación económica o no), algo que se hace con muchos paquetes de distribución comercial.

2. Algunos temas legales relacionados con las licencias

Después de repasar los mitos sobre la legalidad o los efectos legales de las licencias libres en el apartado anterior, este apartado tiene como objetivo proporcionar unos comentarios prácticos sobre algunos aspectos legales de las mismas.

Cuando se quiere crear, distribuir o usar software libre, no se trata sólo de decidir qué software usar y descargarlo desde Internet o, en el caso de su desarrollo, de empezar a proveer de código al "proyecto". Aparte de los aspectos técnicos y económicos que intervienen en tal decisión, hay una multitud de temas legales importantes que cabe considerar para asegurar el éxito de cualquier actividad que implique software libre, ya sea su creación y distribución o su implantación en organizaciones públicas o privadas, y será necesario establecer las estrategias legales pertinentes.

La comprensión de temas legales más amplios relacionados con el software libre, como las consecuencias legales de las licencias o las interrelaciones entre diferentes conceptos relevantes (*copyleft*, compatibilidad, regímenes de licencias, etc.), ayudará a comprender casos particulares de aplicación del software libre y a gestionar los proyectos basados en éste. Además, ayudará a entender los debates públicos, la evolución de las licencias, los contratos de distribución e incluso los litigios pasados y futuros sobre el tema.

Los temas que trataremos en este apartado, los "efectos prácticos" de las licencias libres, tienen que ver sobre todo con la gestión de las mismas y de la propiedad intelectual e industrial en proyectos basados en software libre. En concreto, vamos a estudiar:

- Cómo seleccionar una licencia libre,
- Cómo gestionar las contribuciones a proyectos de software libre,
- La compatibilidad entre licencias,
- Las licencias duales o múltiples (*dual* o *multiple licensing*),
- El efecto de las licencias sobre la división del software libre (*forking*).

Terminaremos el apartado con el repaso de algunos problemas legales que pueden surgir en el momento de usar una licencia libre.

2.1. Seleccionar una licencia libre

Las disposiciones incluidas en las licencias de software libre resultan normalmente del compromiso entre varios objetivos, determinados por los autores o los jefes de equipo (coordinadores) de los proyectos de desarrollo libre. En particular, podemos citar los propósitos siguientes, que en cierta medida, pueden contraponerse entre sí:

- Garantizar ciertas libertades básicas comunes a todo software libre (uso, copia, modificación, redistribución, patentes, etc.);
- Imponer algunas condiciones o restricciones (el reconocimiento de autoría, la falta de garantía, el uso de marcas, etc.);
- Procurar que las modificaciones y las obras derivadas sean también libres;
- Reservarse algunos derechos (por obligación o por voluntad propia);
- Mantener el control sobre la evolución del programa.

Cada proyecto tiene, por lo tanto, sus objetivos y criterios en cuanto a licencia se refiere.

Mientras que la FSF recomienda (casi exclusivamente) el uso de la GPL, en general se recomienda usar una licencia ya existente en lugar de escribir una nueva. Esto se hace cada vez más difícil, con la proliferación de licencias libres (hasta el punto de que la OSI está tratando de reducir el número de licencias certificadas). Una tendencia general es limitarse a una de las licencias más comunes: GPL, LGPL, BSD, MPL, Apache, CPL, etc. Ello ofrece la ventaja de que aumentan las probabilidades de compatibilidad entre programas y componentes. Otra posibilidad es limitarse a una licencia "de tercera generación", como la OSL (*copyleft*/recíproca), la Apache o la AFL (permisivas), o la CDDL (intermedia), que cubren más temas relevantes (patentes, marcas, etc.) y que pueden ser más adecuadas al marco legal europeo, por una voluntad de internacionalización en el momento de su redacción.

El principal criterio de diferenciación es la existencia o no de pactos de *copyleft* o de *reciprocidad*: la obligación de que los desarrollos basados en el software original, generalmente obras derivadas, mantengan la misma licencia para la redistribución. La GPL, por ejemplo, intenta ampliar el *pool* (conjunto) de software libre disponible y maximizar la libertad de los usuarios finales: por lo tanto, impone la cláusula *copyleft*. Asimismo, la GPL tiene el efecto práctico de restringir la evolución separada de un mismo software en diferentes proyectos (*forking*) y de permitir que un equipo de coordinación mantenga cierto control sobre el programa, como veremos a continuación.

Otras licencias tienen un efecto *copyleft* reducido, que se aplica únicamente a la obra (o componente) original y a cualquier modificación de la misma. No se extiende a aplicaciones que "integran" o "usan" el componente libre. Hemos visto en el módulo 6 que la licencia LGPL es típica de esta serie, donde también figuran otras como la MPL, la CDDL y la OSL. Éstas permiten la integración o la vinculación de los componentes originales con otro código, para crear lo que se llaman *obras mayores*.

Reflexión

Algunas preguntas que puede plantearse el estudiante son las siguientes:

¿Quiero permitir la privatización de obras derivadas y modificaciones?

¿Quiero que los desarrolladores devuelvan sus modificaciones a la comunidad libre en general, o a mí, como autor inicial, en particular?

¿Quiero permitir que los licenciatarios fusionen o enlacen su programa con el mío?

¿Quiero una mayor difusión del programa y tratar de establecer un estándar?

¿Quiero obtener beneficios de mi programa a partir de su uso, comercial o de otro tipo, al tiempo que permitir el desarrollo libre?

¿La reputación es importante para mí?

¿Tengo obligaciones hacia terceros en relación con el código incorporado a mi programa?

¿Tengo un programa innovador único, o es otro editor de textos, por ejemplo, cuando ya hay "miles" de ellos disponibles, tanto libres como propietarios?

¿Mi programa debe ejecutarse con otro en particular? ¿Tiene éste restricciones?

¿Quiero incentivar a otros desarrolladores para que participen en mi proyecto y contribuyan con código u horas de pruebas?

¿Mi aplicación ha sido diseñada para ser integrada (*embedded*) en un dispositivo, junto con otro software propietario?

¿Hay una licencia "predominante" en el sector de mi software en particular (por ejemplo, un lenguaje o bibliotecas)?

¿Hay riesgo de que alguien tenga o pida una patente sobre un elemento o un aspecto del programa?

La tabla siguiente, que ya es "clásica" y aparece en casi todos los escritos sobre el tema, toma en consideración las principales licencias libres y asiste en la selección de una licencia.

Licencia	Se puede mezclar con otro software no libre y redistribuir	Se pueden privatizar las obras agregadas	Se pueden privatizar las modificaciones	Hay que licenciar patentes
GPLv2/v3				SI
LGPLv2/v3	SI			SI
MPL	SI	SI		SI
BSD	SI	SI	SI	

Otra opción es la elección de una política de doble licencia, que comentaremos a continuación. Este sistema, en el cual se distribuye el programa con diferentes licencias (normalmente una *copyleft*, la otra restrictiva), permite obtener ingresos a partir de la versión propietaria y colaborar con una "comunidad" para mejorar el programa libre. Este sistema es el que usan MySQL y Trolltech/Qt, entre otras empresas. Además, si el programa es modular, se puede prever el uso de diferentes licencias para diferentes componentes, siempre que sean compatibles entre ellas, dependiendo del grado de integración, y con la licencia sobre el programa distribuido "como un todo". Otra estrategia para sistemas cliente/servidor es el uso de una licencia libre para el cliente y una licencia propietaria para el servidor.

Lecturas recomendadas

Lecturas que asisten en la elección de una licencia:

- Z. O'Whielacronx. "Quick reference for choosing a free software license" [en línea]. <http://zooko.com/license_quick_ref.html>
- B. Perens. "The open source definition". *Open Sources* (pág. 185). <http://oreilly.com/catalog/opensources/book/perens.html
- D. K. Rosenberg. "Evaluation of public software licenses" [en línea; visitado el 29/03/2001]. <http://www.stromian.com/Public_Licenses.html>
- F. Hecker. "Setting up shop. The business of open source software" [en línea]. <http://www.hecker.org/writings/setting-up-shop.html>
- M. Perry. "Open source licenses" [en línea]. <http://fscked.org/writings/OpenSource.html>
- B. Behlendorf. "Open source as business strategy". *Open Sources*.
- R. Brooks. "Open source licenses overview" [en línea]. <http://www.vrml.org/TaskGroups/vrmlipr/open_source_overview.html>
- E. Kidd. "A history of open source" [en línea; 19 de agosto de 2000]. <http://discuss.userland.com/msgReader$19844#19889>
- Estudio POSS / IDA. *Unysis para la Unión Europea* (pág. 60-65).
- The Mitre Corporation. *Use of free and open source software (FOSS) in the U.S. Department of Defense* (versión 1.2, 28/10/2002, pág. 15).

2.2. Las licencias sobre las contribuciones y autoría

Un elemento esencial que se debe tener en cuenta a la hora de gestionar un proyecto libre y las licencias implicadas es el de los autores y las contribuciones al proyecto. La historia de Netscape muestra las dificultades que uno puede encontrar si decide liberar el software o pasar de una licencia libre a otra.

Los problemas que se pueden presentar incluyen:

- Software proveniente de una fuente no segura (puede haber sido copiado).
- Software aportado con otra licencia (una licencia incompatible).

- Software cubierto por obligaciones preexistentes, ya sea por las licencias de terceros o por compromisos que vinculan al autor-licenciante (el problema de Netscape).

- Restricciones en una licencia anterior que son incompatibles con una nueva licencia deseada. Aunque el autor siempre pueda licenciar de nuevo su código (excepto si se ha comprometido a no hacerlo), esto puede provocar problemas de división de código e incompatibilidad (el problema de Unix: hay tantas versiones que es difícil saber cuál es la licencia o el origen de una parte de código).

- Patentes otorgadas sobre un elemento del código.

Para los responsables del proyecto es necesario realizar un seguimiento cuidadoso del código aportado y mantener un registro de autoría y de versiones, para identificar y monitorizar cada componente del software. Los contribuidores suelen aportar código en diferentes formas: la licencia del proyecto, una licencia compatible con la del proyecto o una cesión más individualizada (un acuerdo sobre contribuciones). En este último caso, los responsables del proyecto podrían exigir que cualquier contribuidor otorgue una licencia total y exclusiva a la entidad que coordine el proyecto (en el derecho anglosajón, esto constituye una transferencia de titularidad, lo que no es posible en relación con los derechos morales, sometidos a derecho continental) con garantías respecto a la titularidad de los derechos sobre la contribución y, eventualmente, una licencia de patente. De esta manera, el coordinador del proyecto podrá mantener un cierto control sobre el resultado del proyecto libre y cubrirse frente al riesgo de que el código provenga de fuentes no seguras (por ejemplo, que sea copiado de otro software) o de que cualquier autor original haya solicitado una patente sobre el software o, en derecho anglosajón, revoque su licencia.

2.3. La compatibilidad entre licencias

Hemos hablado varias veces del tema de la compatibilidad de código y de licencias.

Ejemplo

La FSF exige que cualquier programador que contribuya con más de diez líneas de software a un proyecto GNU ceda de manera exclusiva el código a la FSF.

Un programa es compatible con otro si se pueden mezclar o eventualmente interrelacionar sus códigos para crear una obra derivada compuesta de elementos de cada uno de ellos y distribuir el resultado de la integración sin infringir las licencias de uno **y** de otro, de manera que se pueda cumplir con las condiciones de ambas licencias a la hora de redistribuir el resultado.

Para distribuir de manera legal un software que integre varios componentes de software libre es esencial que las licencias sobre los componentes (licencias *inbound*) sean compatibles con la licencia seleccionada para la distribución del

producto final (licencia *outbound*). Por ejemplo, como la licencia BSD permite casi cualquier acción con el software con BSD, se puede mezclar o integrar software bajo BSD con otros programas, con casi cualquier licencia (la GPL, por ejemplo) y distribuir el resultado con esta licencia sin infringir la BSD. El resultado se regirá por la licencia más restrictiva.

Las licencias con *copyleft* son incompatibles entre ellas, salvo pacto explícito (como la EUPL, comentada en el módulo 6, o la MPL, si el titular ha incluido la posibilidad de usar otra licencia): a la hora de redistribuir un programa que integre dos componentes con diferentes licencias con *copyleft*, al usar una de las licencias para la distribución del producto final se infringirá la otra. La FSF ha ofrecido una tabla muy compleja sobre la compatibilidad entre la GPLv2 y la GPLv3, en diferentes casos (http://www.fsf.org/licensing/licenses/gpl-faq.html#AllCompatibility).

Reflexión

Ya hemos clasificado las licencias compatibles con la GPL como criterio de análisis. En todos los casos de integración de código con software con GPL, la distribución del código resultante deberá someterse a la GPL (si no, no se podría mezclar con el código GPL, dado que sus obras derivadas tienen que distribuirse con la GPL).

Compatibilidad con la GPL

Varios proyectos se han esforzado por hacerse compatibles con la GPL, por ejemplo Python, Qt y Vim, e incluso Mozilla agregó la cláusula adicional para permitir la licencia dual.

Muchos exponentes del software libre recomiendan usar una licencia compatible con la GPL, sobre todo porque es la que usan casi un 75% de los proyectos de software libre (no necesariamente el 75% del software libre disponible), pero también porque los proyectos GPL generalmente tienen mayor apoyo de la comunidad de desarrollo libre. En esto existe cierta polémica, porque hay proyectos y desarrolladores que se niegan a aceptar código con GPL y otros que sólo aceptan el código con GPL o una licencia compatible.

Otro tema relevante es la compatibilidad "por enlace": aun si no se permite mezclar o integrar software con diferentes licencias, acaso se puedan enlazar. Ello puede hacerse de diferentes formas, como hemos discutido en relación con la GPLv2: por ejemplo, insertando una API entre un componente y otro, o creando enlaces dinámicos que se activan durante la ejecución.

Enlazar con software bajo la MPL

La MPL prevé expresamente que se pueda enlazar una aplicación propietaria con un programa con esta licencia. El programa propietario debe vincularse con el software con MPL por medio de una API dentro del programa con MPL. La API será parte del programa original, o una modificación hecha a medida, y por lo tanto, estará sometida a MPL.

2.4. Regímenes de licencia duales o múltiples (*dual* o *multiple licensing*)

Un programa puede ser distribuido por su titular con dos licencias libres o más, o bien con una licencia libre y una licencia propietaria, en un régimen de doble licencia. El titular no tiene restringidas las formas de licenciar su código, excepto si ha otorgado una licencia exclusiva o ha acordado compromisos de confidencialidad.

Distribución dual o múltiple

Hay varios programas que se distribuyen de esta manera:

a) **MySQL** es un motor de base de datos que se distribuye con la GPL para usos personales y con una licencia propietaria para integrarlo en productos comerciales.

b) **eZ-publish** es un programa de gestión de contenidos en Internet que también tiene una licencia doble GPL/propietaria.

c) La **MPL** permite al titular establecer si el programa se distribuye con la MPL y otras licencias (cláusula 13).

Desde el punto de vista del titular del software, la posibilidad de usar una segunda licencia permite ofrecer diferentes soluciones a diferentes grupos de interés: colaboradores (licencia libre) o clientes (licencia propietaria o libre con restricciones).

Para lograr esta estrategia, en el caso de usar una licencia propietaria como segunda licencia, es esencial que el titular del software se asegure de ser también titular de los derechos en todos los componentes incorporados en el producto. Por ello, se recomienda concentrar los derechos de autor sobre los componentes en manos de una sola entidad, como hacen la FSF y también las empresas MySQL AB y Trolltech. Esto implica obligar a los contribuidores a ceder de manera exclusiva sus derechos sobre su contribución, establecer licencias exclusivas con socios comerciales y controlar el riesgo de patentes. Ello también permite controlar el precio, la calidad y las responsabilidades sobre el software y, eventualmente, liberarlo (como ha hecho Sun con las tecnologías Java).

El éxito de la estrategia depende inversamente del potencial de terceros (ya sean comerciales o la comunidad libre) de crear un producto similar o *fork* que constituirá una competencia para el producto con licencia no libre. El uso de una licencia *copyleft* para la licencia libre impide que terceros tomen el programa libre y lo privaticen para fines comerciales en competencia con la versión comercial con licencia no libre. Otro elemento para controlar el software en el contexto de esta estrategia es la marca, una herramienta legal ya comentada en el módulo 3. Es utilizada por la Fundación Apache respecto a su software (el servidor web, Tomcat, etc.), por Sun en relación con Java™ o Jini™, y por

varias empresas de software libre profesional como Sugar (CRM), Compiere y Openbravo (ERP), Pentaho (Business Intelligence), Alfresco (gestión documental) o Zimbra (correo y *groupware*).

2.5. Licencias libres y la división de software libre (*forking*)

El concepto de *forking* o división viene de la informática multitarea: alude a la división de una tarea o proceso en dos. En este caso, por ejemplo, una tarea puede seguir activa mientras que la otra se detiene. La división o bifurcación de un programa ocurre cuando el software se modifica y esta modificación se desarrolla de manera separada, con otro equipo de coordinación, y se distribuye con otro nombre, y quizás otra licencia. Son ejemplos de ello OpenBSD y NetBSD, divisiones de la Unix BSD original.

Forking

MySQL define el forking como: "*División* [de MySQL] significa dividir el código fuente de la base de datos MySQL en un repositorio mantenido separadamente de manera que cualquier desarrollo sobre el código original requiera una operación manual para ser transferido al software dividido, o que el software dividido empiece a tener funcionalidades que no están presentes en el software original".

UNIX

El programa que más ha padecido este fenómeno es UNIX, del que surgieron casi diez variantes. Algunas variantes de UNIX fueron creadas a medida que sus autores originales (AT&T y la Universidad de California, Berkeley) iban cediendo licencias libres sobre el programa que permitían crear versiones nuevas (y propietarias): Unixware de Novell, Open Server SCO, Solaris de Oracle, AIX de IBM, etc. Esto ha dado lugar a problemas legales, por ejemplo en el caso original entre AT&T y la Universidad de California, y más recientemente entre SCO e IBM, y otros.

La posibilidad de división es relevante para los desarrolladores, porque provee de una indicación de la posible evolución técnica y legal o comercial del software. Desde el punto de vista técnico, las versiones "divididas" tienden a ser incompatibles o no interoperables con los programas originales. Desde el punto de vista legal y comercial, dichas versiones pueden erigirse como competencia del producto original y distribuirse con una licencia diferente, libre o comercial, lo que fomenta una incompatibilidad legal.

Observamos que los desarrolladores dividen un software libre porque lo pueden hacer: las licencias libres permiten modificar un software libre y redistribuir las modificaciones. Por ejemplo, la licencia BSD permite crear una obra derivada del software original y privatizar el código modificado. Este nuevo programa podría ser solamente una variante del original (por ejemplo, OpenBSD en relación con NetBSD) y, por lo tanto, se consideraría un *fork*.

Hay varias causas de división de un código libre. La principal reside en la gestión del equipo de desarrollo y en desacuerdos entre los programadores o los titulares del software (Mambo/Joomla, Compiere/Adempiere, son un ejemplo

de ello). Por ejemplo, si se identifica la necesidad de una extensión o módulo nuevo y el coordinador no está de acuerdo, es muy probable que alguien cree una versión bifurcada para integrar ese módulo.

Las licencias libres tienen una influencia directa sobre la posibilidad de división:

- Las licencias libres fomentan los *forks*: no se puede dividir el software con licencia propietaria o el software compartido o *shareware*.

- El software con una licencia que no permite el uso comercial y/o exige devolver las modificaciones al autor original no se dividirá, a causa del control centralizador que ejerce el autor (por ejemplo, Ghostscript y la licencia Aladdin).

- El software con GPL tenderá a no dividirse: hay que mantener la libertad de las obras derivadas y el proyecto original puede, por lo tanto, volver a incorporar cualquier mejora y hacerla parte de su versión.

- El software con licencias "intermedias" como la Artistic, la MPL o la LGPL podrá dividirse con más facilidad, porque permite crear variantes propietarias o libres por "agregación".

- El software con licencia BSD o similar se dividirá con mucha facilidad, porque permite distribuciones binarias sin obligación de publicar el código fuente, y por lo tanto, la versión original no tendrá acceso a los cambios para incorporarlos.

3. Licencias libres y otras ramas del derecho

El estudio de las licencias libres en elmódulo 6 se ha centrado, sobre todo, en el derecho de la propiedad intelectual y en algunos aspectos contractuales. En este apartado queremos presentar y comentar algunas dimensiones de estas licencias en relación con otras ramas del derecho: las relativas a la competencia, a las patentes y a las marcas (propiedad industrial), al secreto comercial y a los estándares.

3.1. Licencias libres y el derecho de la competencia

Antes de asumir nuevas políticas más abiertas frente al software libre, algunas grandes empresas de software alegaron que el software libre era "anticompetitivo". Argumentaban que las licencias libres obligan a las empresas a revelar sus secretos confidenciales, que la licencia GPL "infectaba" a cualquier código privado y obligaba a distribuirlo con la misma licencia, sin que el propietario pueda tener beneficios, y que el software libre era una amenaza para las empresas privadas que sustentan sus beneficios en el modelo tradicional de desarrollo (regalías o cánones pagados por la licencia, que remuneran la labor de desarrollo y no únicamente el coste de distribución física).

Es importante comentar la relación entre software libre y el derecho de la "competencia". Éste restringe o prohíbe las actividades anticompetitivas de una o más empresas, sobre todo con el objetivo de proteger a clientes y consumidores. Las reglas se aplican también a las administraciones públicas en sus relaciones con empresas públicas y a otras instituciones con derechos especiales acordados por los gobiernos.

Marco legal formal del derecho de la competencia

El marco legal formal del derecho de la competencia incluye:

- A nivel internacional, las directrices de la OECD y las reglas de la OMC y de otras organizaciones económicas internacionales.
- A nivel regional, en la Unión Europea, los artículos 81 y 82 del Tratado de la Unión Europea.
- A nivel nacional, cada país miembro de la Unión Europea ha instaurado versiones de esos dos artículos en el régimen jurídico nacional. Fuera de Europa, muchos países incorporan reglas similares (sobre todo Estados Unidos, pero también América Latina, Japón, etc.).

La cuestión principal es determinar si el uso de una licencia libre puede considerarse una práctica anticompetitiva. Brevemente, podemos afirmar que existen dos tipos de actividades ilegales:

1) Los acuerdos entre empresas que tienen el efecto de distorsionar el comercio.
2) El abuso de una posición dominante en un mercado, por parte de una o más empresas, que reduzca la competencia en el mercado.

Por un lado, una licencia es un acuerdo "vertical" entre empresas o personas. Como el licenciante está en una posición más fuerte, podría intentar imponer restricciones que tengan un efecto anticompetitivo sobre el licenciatario. Las siguientes cláusulas en una licencia de software podrían ser declaradas ilegales:

- Un pacto que prohibiese la descompilación o la corrección de errores, si tuviese el efecto de distorsionar la competencia (por ejemplo, impedir la provisión de servicios de soporte y mantenimiento).

- Pactos que obligasen a comprar un tipo de software vinculado con un hardware.

Por otro lado, los siguientes términos en una licencia de software podrían tener efectos anticompetitivos cuando el licenciante tuviese una posición dominante:

- Una cláusula que obligase a devolver al autor inicial cualquier corrección o modificación del código.
- Una cláusula que vinculase el software a la compra de otro producto.
- Una cláusula que prohibiese el uso de otro tipo de software.
- La negativa a proveer de una licencia a competidores, o a revelar detalles de una interfaz.
- La restricción del uso de un software a un solo equipo (sin provisión de más licencias).

Ya hemos visto que este tipo de cláusulas no existe en las licencias libres. Al contrario, la mayoría de estas cláusulas figurarían casi exclusivamente en licencias propietarias. Sin embargo, en la licencia pseudolibre de Apple (APSL 1.0) había una cláusula que ilustra el primer caso de la lista: la obligación de devolver las correcciones al autor original.

En otro contexto, las reglas del derecho de la competencia prohíben cualquier acuerdo que aplique a terceros (contratantes) condiciones desiguales para prestaciones equivalentes: esto constituiría discriminación. Esto afectaría a las licitaciones públicas, si la Administración pública intentara imponer uno u otro tipo de software (propietario o libre) o, más aún, si indicara una preferencia por una licencia en particular, salvo condiciones particulares (por ejemplo, un contrato de extensión de un software GPL existente). La obligación de suministrar software con una licencia libre podría considerarse como un término discriminatorio contra empresas de software propietario.

Sin embargo, en dichos casos, los gobiernos podrían beneficiarse de excepciones previstas por la ley, argumentando que el software libre favorece el progreso tecnológico, así como una participación más equitativa de los consumidores en el beneficio. Esta excepción, por ejemplo, es uno de los fundamentos del Sexto Programa de Investigación RTD de la Unión Europea, que favorece la creación de software libre y el uso de licencias libres para la distribución de los resultados de investigación y desarrollo. Actualmente las administraciones públicas tienden a insistir sobre el cumplimiento de normas y estándares internacionales (libres) para obtener software interoperable y garantizar el acceso inmediato o condicional al código fuente.

Reflexión

Como hemos visto en los últimos quince años, por los problemas legales de Microsoft ante las autoridades responsables de la competencia, la relación entre el derecho de la competencia y la propiedad intelectual (incluso las licencias) es muy compleja y discutida. A pesar de ello, lo que no hay que confundir son las prácticas anticompetitivas, como las restricciones y los abusos que hemos comentado brevemente, con la competencia entre dos modelos de desarrollo diferentes.

3.2. Licencias libres y licencias de patente

Hemos visto en el módulo 3 que una patente sobre un proceso o código informático impide no solamente su copia o modificación, tal como hace el derecho de autor, sino también su recreación o reingeniería (lo que los norteamericanos llaman *clean-room development*, es decir, un desarrollo sin acceso a versiones del código original) y el uso y la comercialización del producto o el proceso patentado.

Para minimizar el riesgo que suponen las patentes, las licencias libres más modernas (MPL, CPL, ASL 2.0, GPLv3, etc.) incluyen cláusulas para otorgar licencias cruzadas de patentes entre contribuidores y desarrolladores y pactos resolutorios para el caso en que se iniciara un litigio basado en una patente contra otro contribuidor. Esta práctica es común en el mundo del desarrollo propietario, como se ha indicado en el módulo 3 y comentado en el módulo 6 respecto de la GPLV3 y MPL, por ejemplo.

La licencia de patente otorgada debe cubrir todos los actos (reservados a los titulares de la patente) concedidos, en términos de derechos de autor, por la licencia libre: la reproducción, la distribución, etc. Éstos son muy amplios, e incluyen la fabricación, el ofrecimiento, la puesta en el comercio, la utilización y la importación o la posesión del objeto o el procedimiento patentado, o el fruto del procedimiento.

Sin embargo, estas cláusulas pueden tener sus problemas. En ciertas jurisdicciones no se puede limitar u obligar ciertos actos futuros. Por lo tanto, las cláusulas que obligan a licenciar una eventual patente pueden no ser válidas. Para evitar el uso de patentes, quizás sería mejor usar la limitación general de la GPL, que impide imponer sobre cualquier distribución o modificación restricciones mayores que las incluidas en la misma GPL. Esto tiene el efecto práctico

de impedir el uso de patentes o, si se obtiene una patente sobre un proceso incorporado en el software o una obra derivada de él, se deberá otorgar una licencia en términos no más restrictivos que la GPL.

3.3. Licencias libres y secreto comercial

Otra manera de proteger el software es con el derecho del secreto comercial, como se ha comentado en el módulo 3, que permite proteger secretos de negocio e información de las actividades comerciales, como el código fuente o la documentación técnica de un software.

Un desarrollador de software, al liberar un programa bajo licencia libre podría revelar (intencionalmente o con negligencia) a través de su programación, intencionada o negligentemente, los secretos del negocio de la empresa donde trabaja, o en la que ha trabajado y con la que sigue relacionado por obligaciones contractuales de confidencialidad. Las obligaciones de secreto pueden incluir no solamente información de la empresa en cuestión, sino también información obtenida por medio de una relación comercial con una empresa tercera, a la cual puede haber tenido acceso el desarrollador, relativa, por ejemplo, a un software propietario comprado a un proveedor, cuyo uso y ejecución estarán sometidos a obligaciones de confidencialidad relativas al diseño, a las especificaciones, a la arquitectura, etc. En estos casos, la aportación de código a un proyecto y/o la distribución del software con licencia libre pueden constituir una vulneración de los derechos del secreto de terceros, derechos que luego afectarán al uso del programa libre y a la distribución de código fuente, abiertamente legible por terceros.

Respecto a ello, debemos comentar que es muy difícil separar los "conocimientos del negocio" que están efectivamente protegidos por el secreto comercial de las "competencias" que son propias del programador y que él mismo puede usar libremente en el ejercicio de su profesión o de otras actividades. Asimismo, hay que considerar el efecto de las cláusulas de exclusión de responsabilidades contenidas en las licencias libres en relación con este tema.

SCO *versus* IBM

SCO denunció a IBM por haber incorporado sus "secretos de negocio" en la versión 2.2 de Linux. SCO alegó que dichos secretos fueron comunicados a IBM por medio de su acceso al sistema operativo SCO Unix con obligaciones contractuales de confidencialidad. En su defensa, IBM argumentó, en primer lugar, que no había secretos para revelar, ya que los procesos del software provisto por SCO ya estaban abiertamente disponibles por medio del código libre de Unix; y en segundo lugar, que no había ninguna "incorporación", tal como alegaba SCO: IBM sostuvo que no hubo ningún contacto entre el equipo de AIX que trabajaba con el software de SCO y el equipo "libre" de IBM que trabajaba sobre Linux.

3.4. Licencias y marcas

Otra rama del derecho que puede relacionarse con el software libre y las licencias es el derecho de marcas, un aspecto muy relevante para cualquier persona interesada en la comercialización o en la publicación de software libre. Ya

hemos visto en el módulo 3 que este derecho protege las marcas registradas, que indican la conexión entre una persona o empresa y sus bienes y servicios, y los diferencian de los productos de otra persona.

Como los avisos de autoría, las marcas contribuyen a proteger la calidad del software original y, por lo tanto, la reputación de los autores iniciales. Por ejemplo, Linus Torvalds ha registrado la marca Linux® en relación con el software que conocemos como Linux. Nadie más puede usar esta marca sin el consentimiento de L. Torvalds. El registro de una marca sobre un software libre sirve para sostener la autoría y gestionar la evolución del software, independientemente de los permisos acordados por la licencia.

Es importante considerar que la marca registrada es un bien intangible y un derecho separado del software, y que cualquier licencia debe incluir pactos al respecto si un distribuidor intermediario quiere usar dicha marcha. Las cláusulas de las licencias libres suelen establecer un derecho o la prohibición de uso, en ciertas condiciones (por ejemplo, si el software ha sido modificado).

El modelo de distribución y comercialización profesional del software libre depende en gran medida de las marcas (Red Hat®, SuSe®, Mandrake®, Zope®, Apache®, Java®, Sugar®, Pentaho®, MySQL®, etc.), y ciertas licencias libres incluyen términos precisos sobre su uso. Hemos visto que algunas licencias obligan a los usuarios a indicar la autoría y la marca (Apache), otras prohíben su uso sin el consentimiento expreso del titular de la marca (Zope), otras impiden su uso sobre obras derivadas. Se deberá consultar la licencia a la hora de indicar públicamente el software utilizado en alguna aplicación ("basada en Linux", "programada con PHP", "con tecnología Apache", "extensiones para MySQL").

Hay que decir que las licencias libres no son tan completas como las licencias comerciales de marca, en cuanto al uso de las marcas, su presentación, el tamaño del logotipo, las condiciones de uso, territorios, productos y servicios relacionados, etc. Las empresas de software libre (JBoss, MySQL, Alfresco, Red Hat, Openbravo, etc.) suelen establecer políticas de uso o contratos accesorios de comercialización que incluyen pactos completos sobre el uso de sus marcas registradas. Algunos ejemplos incluyen el uso de la marca Sun® (o Java®), MySQL® o Red Hat®.

3.5. Licencias y estándares

Los estándares van a ser otro campo de batalla del software libre, junto con las patentes y los sistemas de gestión de derechos de autor (DRMS y *trusted computing*).

El software libre se basa considerablemente en estándares abiertos, que permiten la interoperabilidad y la interconexión de los sistemas informáticos. Pensemos en los protocolos de Internet (HTTP, TCP/IP, etc.), las interfaces para

periféricos (como los controladores o *drivers*) y los algoritmos de cifrado (hash MD-5, SSH, etc.). Por lo tanto, mantener la apertura de los estándares es muy importante para el software libre.

Hay una tensión entre la estandarización y la propiedad intelectual e industrial, que son casi opuestas: aquélla "abre" mientras que ésta "reserva". Ha habido varias denuncias de tentativas de patentar o mantener derechos exclusivos de autor sobre tecnologías o protocolos aceptados como estándares por la W3C. Algunos actores comerciales hablan de licenciar en términos "razonables y no discriminatorios" (RAND) cualquier tecnología que sea aceptada como estándar. Pero RAND no tiene sentido para un desarrollador libre independiente.

Por otro lado, se ha observado que el sistema de desarrollo libre es un proceso de estandarización no formal en el que, al final del proceso, un software o un protocolo dado se erige en "estándar" para el sector. Asimismo, el desarrollo libre es un proceso más rápido que el proceso formal de estandarización, lo que ha fomentado, por ejemplo, un alto nivel de interoperabilidad en el contexto de Internet. SendMail, BIND, SMTP, etc. son programas o protocolos libres cuya especificación ha resultado en un estándar para la industria de Internet.

Las licencias libres favorecen los estándares abiertos. Ya hemos observado que las licencias más flexibles, como la LGPL o la BSD, permiten la difusión rápida de una tecnología y su adopción como estándar tanto en sectores propietarios como en sectores libres. Desafortunadamente, también permiten privatizar extensiones que pueden aplicarse o agregarse a los estándares y hacerlos no interoperables con el software original (por ejemplo, la versión Microsoft de Kerberos).

4. Los datos personales y la protección de la intimidad

En este apartado consideraremos la relación entre el software libre y la protección de datos personales. Primero, introduciremos el concepto de *privacidad* y su marco legal; luego, comentaremos cómo el software libre puede estar afectado por el derecho de la privacidad y viceversa, y cómo los sistemas comerciales y públicos pueden aprovechar el software libre para cumplir con este derecho.

4.1. Introducción y marco legal

Ya en los años setenta se desarrollaron varias iniciativas para definir un sistema de protección de la privacidad, que dieron lugar a la aprobación de diversos textos legales en diferentes países europeos: el **Convenio de Europa 108**, del 28 de enero de 1981, la **Directiva Europea 95/46**, relativa a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos datos (en adelante, la Directiva), y su implementación en España, la **Ley Orgánica 15/1999, de Protección de Datos de Carácter Personal** (en adelante, la LOPD), junto con otras disposiciones que desarrollan aspectos particulares, como el Real Decreto 994/1999 (el **Reglamento de Medidas de Seguridad**, actualmente en proceso de revisión).

4.2. El régimen legal de protección de datos personales

La LOPD tiene por objeto garantizar y proteger los datos personales, las libertades públicas y los derechos fundamentales de las personas físicas, especialmente su honor y su intimidad personal y familiar.

Para entender el marco legal de la privacidad, es importante considerar los siguientes conceptos básicos:

- **Datos personales:** cualquier información concerniente a personas físicas identificadas o identificables. Hay un subgrupo de datos especialmente protegidos (ideología, religión y creencias, origen racial, salud, etc.).

- **Ficheros:** todo conjunto organizado de datos de carácter personal, cualquiera que sea la forma o modalidad de su creación, almacenamiento, organización y acceso. Se extiende a ficheros no automatizados y a todo tipo de dato personal susceptible de tratamiento.

- **Tratamiento:** cualquier tipo de operaciones y procedimientos técnicos, de carácter automatizado o no, que permitan la recogida, la grabación, la conservación, la elaboración, la modificación, el bloqueo y la cancelación,

así como las cesiones de datos que resulten de comunicaciones, consultas, interconexiones y transferencias.

- **Interesado:** la persona física titular de los datos que son objeto de tratamiento. Las personas jurídicas están excluidas de protección.

- **Responsable de fichero:** la persona física o jurídica, la autoridad pública, el servicio o cualquier otro organismo que decida sobre la finalidad, el contenido y el uso del tratamiento (qué, quién, cómo, cuándo y dónde). El responsable del fichero responde administrativa, civil y penalmente de las posibles infracciones de la LOPD.

- **Encargado de tratamiento:** la persona que, sola o juntamente con otras, trata datos por cuenta del responsable del tratamiento.

- **La Agencia Española de Protección de Datos** (**AEPD**)**:** se establece una autoridad nacional con poderes de sanción para garantizar la protección de datos personales y mantener los registros de ficheros notificados.

Ficheros con datos personales

Como ejemplos podemos citar cualquier conjunto de datos, como el historial de los pacientes de un médico (a condición de que estén ordenados según un criterio lógico), o el perfil de los usuarios de un sitio web (clientes, personas registradas, etc.). No importa si el soporte de los datos tiene un formato físico o electrónico, y tampoco es relevante, en principio, si los datos son objeto de un tratamiento automatizado o no.

Ámbito objetivo. La LOPD se aplica al tratamiento de datos de carácter personal registrados en soporte físico y a toda modalidad de uso posterior de estos datos por los sectores público y privado. Están excluidos de protección los ficheros mantenidos por personas físicas en el ejercicio de actividades exclusivamente personales o domésticas (por ejemplo, una agenda personal), y los ficheros relacionados con la defensa nacional y la protección del Estado, el terrorismo y formas graves de delincuencia organizada.

Ámbito geográfico. La LOPD se aplica cuando el tratamiento sea efectuado en territorio español, en el marco de las actividades de un establecimiento del responsable del tratamiento en España o en la Unión Europea, o cuando el responsable del tratamiento no está establecido en territorio español pero es de aplicación la legislación española según las normas del derecho internacional público.

4.2.1. Principios generales

La LOPD establece varios principios generales que se deben respetar. Los principales son los siguientes:

- **La calidad de los datos y la finalidad de su tratamiento.** Los datos de carácter personal sólo se podrán recoger y tratar cuando sean adecuados,

pertinentes y no excesivos, en relación con el ámbito y las finalidades para las que se hayan obtenido. Estos criterios se determinarán en función del caso en concreto. No podrán usarse para finalidades incompatibles con aquella para la que se hubieran recogido, y cuando se haya cumplido esa finalidad, han de ser cancelados o destruidos. Asimismo, los datos deben ser actualizados y exactos.

- **Información en la recogida de los datos.** Los interesados deberán ser previamente informados de modo expreso, preciso e inequívoco de que sus datos van a ser incluidos en un fichero, de la finalidad de su tratamiento, de los destinatarios de la información y de varios otros datos pertinentes (la obligatoriedad o no de dar los datos, los derechos de acceso, rectificación, cancelación y oposición, la identidad y la dirección del responsable del tratamiento).

- **Consentimiento del interesado.** El tratamiento de los datos de carácter personal requerirá el consentimiento inequívoco, expreso o tácito, del interesado, salvo que la ley disponga otra cosa (es decir, una autorización legal, por ejemplo, por orden judicial). El tratamiento de los datos especialmente protegidos requiere el consentimiento expreso y por escrito. Hay varias categorías de datos que no requieren el consentimiento (por ejemplo, datos recogidos en fuentes accesibles al público, datos en contratos comerciales o laborales y datos recogidos con la finalidad de proteger un interés vital del interesado).

- **Secreto.** Tanto el responsable del fichero como el encargado de tratamiento, así como cualquier persona que intervenga en cualquier fase del proceso, están obligados al secreto profesional respecto de los datos, y al deber de guardarlos.

- **Comunicación y acceso a los datos.** Aunque hay varias excepciones, cualquier comunicación o cesión de los datos a terceros requiere la autorización previa del interesado.

4.2.2. Derechos y obligaciones

Como consecuencia de estos principios y de otras disposiciones de la Ley, el interesado se beneficia de varios derechos y el responsable está sujeto a una serie de obligaciones.

El interesado tiene el **derecho a recibir la información** antes mencionada en el momento de la recogida de datos, y el derecho de **acceso, rectificación y cancelación de los mismos**, con el fin de mantener la exactitud de los datos y de rectificarlos o cancelarlos cuando resulten incompletos o inexactos, inadecuados o excesivos para la finalidad. Asimismo, puede **oponerse** a su tratamiento cuando existan motivos fundados y legítimos relativos a una situación

personal concreta. Por otro lado, el interesado puede **impugnar** cualquier acto administrativo o privado que implique una valoración de su comportamiento sobre la base de un tratamiento automatizado de datos personales. Finalmente, puede **consultar gratuitamente** el registro general de la AEPD.

El responsable de tratamiento está sujeto a diversas obligaciones, las principales de las cuales son las de **notificación e inscripción** de los ficheros ante la AEPD, la de provisión al interesado de la **información** ya mencionada, y la de obtención de su **consentimiento** cuando sea necesario. Además, debe **asegurar los procedimientos** para permitir a los interesados el ejercicio de sus derechos de acceso, rectificación y cancelación. Debe **documentar** las relaciones con terceros que intervienen en el tratamiento y, en particular, designar al **encargado de tratamiento** mediante contrato. Por otro lado, el responsable del fichero debe instaurar las medidas de seguridad de índole técnica y organizativa necesarias para garantizar la seguridad de los datos objeto de tratamiento.

El encargado de tratamiento deberá también cumplir con las obligaciones incorporadas en el contrato de tratamiento: desarrollará su actividad por cuenta del responsable del fichero y tratará los datos conforme a las instrucciones que haya recibido. En caso de incumplir las obligaciones que sobre él hace recaer la LOPD, responderá de las infracciones cometidas personalmente.

Ved también

En la bibliografía hay varias referencias que aportan más información sobre la protección de datos personales.

4.2.3. Cesiones y acceso a los datos por terceros y transferencias internacionales

Una **cesión de datos** es cualquier revelación de datos a una persona distinta del interesado. Los datos de carácter personal sólo podrán ser cedidos para el cumplimiento de fines directamente relacionados con las funciones legítimas del cedente y del cesionario y, aunque con varias excepciones (como una autorización legal), siempre con el previo consentimiento informado del interesado.

Como regla general, la **transferencia internacional de datos** está permitida únicamente con destino a países que proporcionen el nivel de protección que presta la LOPD (los miembros de la Unión Europea y cualquier otro país aprobado por la Comisión Europea y, para España, la AEPD); o en algunos otros casos específicos, por ejemplo, cuando el destinatario haya firmado un contrato que garantice niveles similares de protección de datos o transferencias entre miembros de un grupo empresarial que establezca una política interna adecuada de protección de la privacidad. Las excepciones incluyen casos, entre estos, en los que la transferencia serealiza para prevención o diagnóstico médicos, la prestación de asistencia sanitaria o los tratamientos médicos, o en los que se refiera a transferencias dinerarias o sea necesaria para la ejecución de un contrato entre el interesado y el responsable del fichero.

Los países aceptados

Los países aceptados hasta hoy son Argentina, Canadá, Hungría, Suiza y la isla anglonormanda Guernsey (y Estados Unidos, con los principios de Safe Harbor).

4.2.4. Obligaciones de seguridad

El responsable del fichero y, en su caso, el encargado del tratamiento, están obligados a adoptar las medidas de seguridad para la protección de datos personales. La Directiva no predica ninguna medida en particular y los miembros de la Unión Europea han tomado distintas perspectivas sobre la cuestión, desde la autorregulación (Reino Unido) hasta las medidas detalladas obligatorias (España).

Bajo la LOPD y el Reglamento de Medidas de Seguridad se establecen tres niveles de protección en función de la información tratada. El **nivel básico** se aplica por defecto a cualquier fichero cubierto por la LOPD, mientras que el **nivel medio** se aplica a ficheros que contengan datos relativos a la comisión de infracciones administrativas o penales, a la Hacienda Pública, a los servicios financieros y a los de solvencia patrimonial y de crédito; el **nivel alto** se aplica a ficheros con datos sobre ideología, la religión, las creencias, el origen racial, la salud o la vida sexual, y a los datos recabados para fines policiales sin consentimiento del afectado.

Los diferentes niveles de seguridad comportan **obligaciones** cada vez más onerosas ser implementadas por la persona encargada de los datos (responsable de seguridad). Las obligaciones básicas incluyen la elaboración de un documento de seguridad y un registro de incidencias, y la definición de las funciones de las personas con acceso a los datos. Además, para los niveles medio y alto se debe realizar periódicamente una auditoría (técnico-legal) de las medidas de seguridad implantadas, y para el nivel alto es obligatoria la realización de copias de seguridad y el cifrado de las transmisiones de datos.

4.2.5. Sanciones

La LOPD sistematiza los posibles incumplimientos de la Ley, tipificando las infracciones relativas a la protección de datos personales. No hay una lista exhaustiva sino unos criterios que incluyen básicamente la vulneración de los derechos del interesado (el consentimiento informado, y derechos de acceso, rectificación y cancelación), la falta de colaboración con la Agencia, la ausencia de notificaciones preceptivas (como las de creación de fichero) o la creación, el tratamiento, la comunicación, la cesión y el mantenimiento de ficheros sin la observancia de las prescripciones de la Ley.

Hay tres categorías de sanciones: leves (con multas de 600 a 60.000 euros), graves (con multas de 60.000 a 300.000 euros) y muy graves (con multas de 300.000 a 600.000 euros). Se han dado a conocer varias sanciones en aplicación de la LOPD, a pesar de su supuesto secreto:

- En 2000, 180 millones de pesetas (1,1 millón de euros) a Zeppelin (Tele 5) por revelar datos de candidatos al programa *Gran hermano* (recurrido).

- En 2001, Telefónica de España y Telefónica Data fueron sancionadas por un importe de 841.420 euros, por intercambiarse datos de clientes.

- En 2002, la empresa Inlander tuvo que pagar 300.000 euros por tener instalado el servidor en Estados Unidos.

4.3. Marco legal en otras jurisdicciones

Aunque hayamos comentado con detalle el marco legal español, debemos resaltar que la mayoría de esos derechos y obligaciones desarrollados existen con pequeñas variaciones en todos los países de la Unión Europea, por efecto de la Directiva de Protección de Datos Personales. La mayor diferencia reside en las medidas de seguridad, como las que se han explicado para España.

Fuera de la Unión Europea notamos que, por efecto de estas obligaciones y, sobre todo, de la transferencia internacional de datos, la mayoría de socios comerciales de los países europeos se ven casi "obligados" a establecer marcos legales similares para la protección de la privacidad. Mencionemos Canadá, Suiza y Argentina, que han sido aprobados por la Comisión Europea, pero también Japón o Australia.

Un caso particular es el de los Estados Unidos, que tiene un nivel de protección de privacidad muy inferior al marco europeo y con una organización sectorial: sobre todo, para los bancos, los servicios financieros y el sector de la salud. Para permitir la transferencia de datos desde la Unión Europea, Estados Unidos ha establecido un régimen cuasiprivado, por medio del acuerdo de Safe Harbor, de julio de 2000. Cualquier empresa que se comprometa a cumplir las obligaciones establecidas por el Gobierno norteamericano (el Department of Commerce) y acordadas por la Comisión Europea puede recibir transferencias de datos de carácter personal desde la Unión Europea. Hasta la fecha, unas cincuenta empresas americanas han firmado el acuerdo. Otras han firmado contratos estándares obligándose a la protección adecuada.

4.4. Datos personales y software libre: la seguridad

Este breve resumen del marco legal europeo de la protección de datos indica que la mayoría de las obligaciones y derechos relativos al procesamiento de datos personales no tiene una relación inmediata con el software libre. No

por tener software libre o software propietario una persona cumple o viole esa normativa, sino que son los procesos de negocio y de tratamiento los que influyan sobre el cumplimiento: si se ha obtenido el consentimiento, si se da una oportunidad de revisión y modificación de los datos o si se respetan las políticas declaradas de privacidad.

El ámbito en el que sí que hay un vínculo importante con el software libre (o propietario) está en el área de la seguridad. A nivel Europeo, la Directiva de 1995 obliga a los "responsables de tratamiento" a implantar medidas tecnológicas y organizativas para garantizar la protección de datos y el respeto de los derechos otorgados. Aunque existan diferencias en la aplicación nacional de la Directiva en este sentido, se puede decir que hay un acuerdo general acerca de que estas medidas deben cumplirse por medio de políticas de seguridad adecuadas (en lo organizativo) y de aplicaciones informáticas seguras (en lo tecnológico).

Los riesgos de violar estas obligaciones de protección tecnológica son grandes. Una lista no exhaustiva incluirá:

- Los accesos no autorizados a partes protegidas de sistemas corporativos que contienen y procesan datos personales (por ejemplo, directorios del personal, listas de clientes o de pacientes, datos de cuentas bancarias y de tarjetas de crédito, etc.);

- La intercepción de flujos de datos en las redes públicas (y a veces en las VPN);

- La vulnerabilidad a ataques externos por troyanos u otras formas de programas dañinos;

- El robo de identidad (*spoofing*) por medio de la cosecha de datos en comunicaciones no seguras;

- La diseminación involuntaria de datos personales (por error de programación o por fallo del sistema).

Las consecuencias de un fallo de seguridad que implique la violación de la protección de datos personales son igualmente graves, en términos de multas o daños a la reputación de la sociedad en cuestión.

Error en la seguridad

En septiembre de 2003 debido a un fallo de seguridad, ya.com permitió el acceso a sus facturas con datos personales, que incluían nombres, NIF, cuentas bancarias, números de teléfono, etc. Fue penalizada por la AEPD.

4.4.1. Seguridad informática, ¿software abierto o software propietario?

En la esfera de la seguridad existe un gran debate entre los defensores del software libre y los del software propietario. Ambos grupos argumentan que su proceso de desarrollo y el resultado final –abierto o compilado– llevan a mayores niveles de seguridad.

No vamos a entrar en ese debate. Nos limitaremos a resumir sus argumentos:

- A favor del software propietario se argumenta que el secreto del código fuente impide que los atacantes descubran de manera sencilla los defectos (*bugs*) de seguridad. Asimismo es relevante que el proceso de subsanación del código es una tarea difícil e ingrata. Las empresas propietarias tienen la ventaja de poder pagar a expertos para realizar este trabajo, mientras que el desarrollo libre no tenderá a incentivar este tipo de tareas. También se argumenta que una empresa con responsabilidades se hace cargo del software propietario y coordina y concentra el trabajo del buen diseño original (referente a la seguridad) y la corrección de errores. No hay ninguna garantía de este "cuidado" en el software libre, ni para la distribución de parches. La mera publicación de código en Internet no implica que sea examinado desde la perspectiva de la seguridad. Finalmente, nada garantiza la calidad del código aportado a un proyecto libre que no tenga un coordinador de alto nivel.

- A favor del software libre se argumenta que el desarrollo libre se beneficia de "muchos ojos" para descubrir los fallos. Los programas abiertos no utilizan a sus usuarios como probadores. Sin embargo, hay pocos proyectos libres, con excepciones como Linux o Apache, que tengan muchos desarrolladores: el promedio, para la mayoría de proyectos, es de cuatro a seis personas. Pese a ello, una vez descubierto, un defecto se arregla en muy poco tiempo (horas o días), mientras que con el software propietario puede transcurrir un tiempo largo entre el descubrimiento del fallo y la distribución de su corrección. Se alega incluso que solamente la presión mediática hace que las empresas "propietarias" se esfuercen en corregir errores. Y se argumenta que el método de diseño abierto es más seguro que el método propietario al citar los RFC (*requests for comments*, propuestas de estándares en forma de sugerencia de diseño, como IPsec o S/MIME) de las organizaciones de estándares como W3C e IETF.

- Por lo que respecta al usuario, se argumenta que el código abierto permite a los administradores de sistemas abiertos eliminar código indeseado o inseguro, diseñar e incorporar sus propios parches y procesos de seguridad e integrar extensiones de seguridad de terceros.

En términos de estadísticas, tanto Windows como GNU/Linux sufren fallos de seguridad, según la organización CERT, que monitoriza e investiga estas cuestiones. No hay ventaja "numérica", a pesar de los alegatos de la comunidad libre. Desde un punto de vista científico, se argumenta que a igualdad de condiciones los riesgos de un modelo y del otro son iguales. Todavía no hay ninguna investigación que lleve a una conclusión fehaciente a favor o en contra de uno u otro tipo de software.

Web recomendada

Para un comentario sobre el informe CERT, podéis ver "Study: Linux' security problems outstrip Microsoft's", de **J. MacGuire** (en http://www.newsfactor.com/perl/story/19996.html).

R. Andersen argumenta que las simetrías teóricas entre software abierto y software propietario frente a la seguridad están rotas por varios factores que pueden favorecer a un modelo u otro. Podéis ver "Security in open systems v. closed systems. The dance of Boltzmann, Coase and Moore", en la bibliografía.

4.4.2. Conclusiones

En conclusión, es importante resaltar que las obligaciones impuestas sobre las personas que procesan datos personales hacen que se tome la seguridad muy en serio en el momento de elegir, diseñar, desarrollar y/o implementar una plataforma informática. Uno de los argumentos de mayor fuerza de OpenBSD, por ejemplo, es que es el sistema Unix es más seguro y fiable, no solamente por las herramientas de cifrado y protección de comunicaciones que incorpora, sino también porque tiene un equipo dedicado que se ha especializado en resolver los problemas de seguridad. Por lo tanto, un análisis de los aspectos de la seguridad es fundamental en la consideración de la implementación de nuevos sistemas informáticos.

5. El software libre y los controles sobre los productos de seguridad

Otra área de derecho relevante para el software en general es la de los controles de la exportación de productos de seguridad, y sobre todo, los sistemas de cifrado. En este último apartado queremos presentar el porqué de la regulación de estos productos y cómo se ve afectado el software libre por ella.

5.1. Sociedad de la información y seguridad

En la actual sociedad de la información, con la multiplicación de las operaciones diarias que se efectúan por medio de las redes informáticas y su potencial para la invasión de la vida privada, existe una necesidad creciente de mantener los datos comerciales y privados seguros y confidenciales. Sin embargo, la infraestructura de comunicaciones de nuestra sociedad se basa en redes y ordenadores inseguros, y está plagada de problemas de seguridad, ya sea por los protocolos poco fiables en términos de seguridad en los que se fundamenta (como TCP/IP, SMTP o HTTP), por varios programas perjudiciales (como los virus y otras aplicaciones dañinas) o por los errores de codificación de los programas de mayor uso, que dejan nuestros equipos y archivos abiertos a ataques e intrusiones de terceros.

Web recomendada

Podéis ver, por ejemplo, el informe "CyberInsecurity. The cost of monopoly", de Computer and Communications Industry Association, en http://www.ccianet.org/papers/cyberinsecurity.pdf (visitado el 24/09/2003). Este informe ha suscitado un debate feroz sobre los *white papers* y otros informes patrocinados económicamente.

Para paliar estas deficiencias se ha desarrollado una serie de tecnologías de seguridad, la mayoría de las cuales se basan en la criptografía o cifrado, que aparece como el único sistema de protección viable. Estas tecnologías impiden la intercepción de la comunicación y el acceso no autorizado, es decir, preservan la **confidencialidad**. Asimismo, favorecen la certeza jurídica (por ejemplo, para un contrato electrónico seguro):

- La **integridad**: asegurar que los datos no hayan sido modificados;
- La **autenticación**: establecer la identidad de las partes de una comunicación;
- El **no repudio**: garantizar que una parte no pueda negar la existencia o la recepción de un mensaje.

El cifrado asimétrico de alto nivel, basado en el sistema PKI (*public-key infrastructure*), es una de estas tecnologías, que garantiza la seguridad de una comunicación entre dos personas con claves diferentes (pública y privada) por cifrado asimétrico. Lo que se considera cifrado de alto nivel (*strong encryption*) variará según los avances tecnológicos. Hasta una época reciente, el cifrado con 40 bits se consideraba suficientemente alto como para justificar restricciones sobre la exportación de los productos que lo incluían. Hoy, el cifrado a 128 bits se considera el mínimo para garantizar una comunicación segura.

Web recomendada

El cifrado de alto nivel se usa en los servicios bancarios en línea. Podéis consultar como ejemplo www.banesto.es/banesto/home2/informacion_ssl128.htm (visitado el 24/09/2008).

En cuanto a tecnologías para asegurar la confidencialidad y la seguridad de las comunicaciones, mencionemos brevemente los **algoritmos de cifrado** (RSA, inventados en los laboratorios del MIT en 1978 y protegidos por patente –en Estados Unidos– hasta septiembre de 2000; Triple DES, IDEA y Blowfish); tecnologías de seguridad para el **correoelectrónico** (S/MIME, PGP y OpenPGP, el estándar actual de cifrado para correo basado en PGP de Phil Zimmermann); la protección de las **transferencias por Internet** (SSH y SSL, que permiten crear canales de comunicación seguros entre el servidor y los navegadores), y tecnologías de seguridad en **redes privadas o públicas de tipo VPN** (IPSec, *point to point tunelling protocol*, etc.).

Web recomendada

PGP tiene una historia muy interesante que merece una lectura. Para ver una historia de PGP, podéis consultar http://www.cypherspace.org/~adam/timeline/ (visitado el 24/09/2008).

5.2. Los controles sobre los productos de seguridad

El cifrado no solamente protege la confidencialidad de los datos contra el robo, la intercepción y el acceso no autorizado, sino que también impide que las autoridades gubernamentales (policía, aduana) intercepten y accedan a las comunicaciones. Argumentando que se necesita controlar estas comunicaciones para mantener el orden público y para la lucha contra el terrorismo, varios gobiernos han impuesto distintos tipos de controles sobre el uso de los productos de cifrado.

Lectura recomendada

No tratamos aquí la cuestión del acceso a las claves (*key recovery*), porque no es directamente relevante para el software libre. Podéis consultar en la bibliografía los textos sobre el tema: **H. Abelson, y otros**, *Los riesgos del cifrado*; y **Unión Europea**, "*Green Paper* sobre servicios de cifrado en el mercado interno".

Interpretación de la revista Internautas

Interpretación de la revista Internautas. "El Gobierno quiere controlar la criptografía para mantener sus posibilidades de escuchas (esa intención ha sido traicionada por las propuestas fallidas de adopción del chip Clipper por la Administración). La posición del Gobierno de los Estados Unidos es que las técnicas criptográficas pueden ser usadas por malhechores para madurar sus planes ilegales y la policía tendría mucho más difícil seguir sus huellas. Terroristas internacionales podrían usar correo electrónico cifrado para planear alguna mala jugada en territorio norteamericano y el FBI ni se enteraría". Accesible en línea en:
http://www.internautas.org/NOTICIAS/DIC98/11.htm

Los instrumentos de mayor importancia relativos al control de la distribución de productos de cifrado son:

- Las **Directrices de la OCDE, de 1997**, que no son vinculantes, enfatizan la necesidad de mantener disponibles productos de cifrado para la privacidad y la confidencialidad de los datos, sujetos a medidas proporcionales y efectivas para mantener el orden público.

- El **Acuerdo Waasenaar, de 1995** (modificado en 1998 y 2000), un tratado internacional ratificado por treinta y tres países, para la imposición de controles sobre la exportación de armas convencionales y bienes y tecnologías de doble uso. El acuerdo determina una lista de productos y tecnologías controlados, entre los cuales figuran tecnologías de criptografía (en el General Software Note). En 1998, debido a varias campañas a favor de la privacidad y la criptografía, se eliminaron de la lista algunos productos relativos a la autenticación (para la firma digital) y los algoritmos de cifrado de menos de 56 bits. Sin embargo, se amplió la lista con cualquier producto de hardware y software con algoritmos de más de 64 bits. En 2000 se eliminó el límite de 64 bits para el software y el hardware para gran público. El Acuerdo Waasenaar no es derecho directamente aplicable; cada país lo implanta como quiere.

- El **Convenio Europeo sobre el Cibercrimen**, del Consejo de Europa de 2001.

- Los **reglamentos de la Unión Europea** que implementan los acuerdos de Waasenaar (el 1334/2000, sobre los materiales de doble uso, modificado por los reglamentos 458/2001 y 2432/2001). Permiten el libre movimiento de productos de criptografía dentro de la Unión Europea e imponen restricciones sobre su exportación a terceros países (con licencia general o específica). Se mantiene la misma exención que en el Acuerdo de Waasenaar para productos "en el dominio público", y en 1999 se eliminaron los límites sobre software de cifrado para el gran público.

Dentro de la Unión, hay un debate importante entre los defensores de la libertad y la privacidad, que quieren eliminar los controles, y los gobiernos –sobre todo los de Reino Unido y Francia– que quieren mantener ciertas posibilidades de acceso a las comunicaciones.

Web recomendada

Para más información sobre la posición de la Unión Europea, podéis ver el "*Green Paper* sobre servicios de cifrado", COM(96)76, 06/03/1996, en http://europa .eu/documents/comm/green_papers/pdf/com96_76_en.pdf (visitado el 24/09/2008).

A nivel nacional, hay varios regímenes: establecemos una tabla que resume de manera breve la regulación de productos de cifrado en algunos países.

	Régimen	Controles sobre exportación	Exenciones	Condiciones para exención
EE.UU.	Interno + Waasenaar (v. antes de 1998)	EAR - Licencias o notificaciones obligatorias	Dominio público y otros (ver abajo) Gran público	Previa notificación al Gobierno (BIS) (ver adelante)
Canadá	Waasenaar (v. antes de 1998)	Paralelo con EE.UU.	Cifrado hasta 56 bits Exportación a EE.UU. Dominio público Gran público	Control de la reexportación de productos de EE.UU.

	Régimen	Controles sobre exportación	Exenciones	Condiciones para exención
Francia	Waasenaar (v. antes de 1998) + UE	Licencia obligatoria	Gran público Cifrado hasta 40 bits	
Reino Unido	Waasenaar + UE + internos	Licencia obligatoria	Cifrado hasta 56 bits Dominio público Gran público	
España	Waasenaar + UE	Licencia obligatoria (ver recuadro más adelante)	Dominio público Gran público Cifrado hasta 56 bits	

Web recomendada

Para ver más detalles, hay un análisis detallado en "Crypto law survey", en http://rechten.uvt.nl/koops/cryptolaw/index.htm.

Los Estados Unidos han presenciado a un gran debate público sobre los controles de la diseminación de la criptografía frente a la protección constitucional de la libertad de expresión. Tres decisiones judiciales, Bernstein I, II y III, relativas a un programa que usaba criptografía *snuffle*, declararon inconstitucionales algunas partes del sistema de control de exportaciones (ITAR) de entonces. La Administración de Clinton finalmente instauró el régimen actual de EAR (*export Administration regulations*), que mantenía controles similares. En enero y octubre de 2000, para implementar Waasenaar, se relajaron varios requerimientos. El régimen actual es el siguiente:

- La exportación de cualquier producto de criptografía a usuarios no gubernamentales se debe realizar con licencia, excepto a siete países "terroristas" adonde se prohíbe la exportación: Libia, Corea del Norte, Cuba, Irán, Irak, Sudán y Siria.

- Se permite la exportación de productos para "gran público" después de una revisión técnica del Gobierno. Para la Unión Europea y otros estados "amigos" no hace falta esperar a la revisión.

- Se permite la exportación de software con código fuente a disposición del público (*open source* y *community source*) sin restricciones y sin revisión técnica. Solamente se debe notificar al Bureau of Industry and Security (BIS), incluyendo una copia del código fuente (o su URL). Los países "terroristas" quedan prohibidos. Se puede "colgar" el código en Internet sin tener que controlar quién lo descarga.

Web recomendada

Para más detalles sobre el régimen norteamericano, podéis ver Commercial Encryption Export Controls: http://www.bis.doc.gov/encryption

5.3. Controles de exportación y software libre

En relación con el software libre, el régimen de protección de los productos llamados "de seguridad" tiene varias implicaciones.

Por un lado, queda sujeta a un control gubernamental la distribución de tecnologías de cifrado de alto nivel (mayor de 56 bits) fuera de su país, incluyendo la distribución de software por Internet. Pero, por otro lado, la definición de *dominio público* en estos acuerdos no es una definición legal desde el punto de vista de la propiedad intelectual, sino que alcanza a tecnologías o software que han sido puestos a disposición del público (divulgados) sin restricciones para su diseminación posterior (las restricciones que implica el derecho de propiedad intelectual no se consideran restricciones). Esto conlleva que gran parte del software libre se considere "en el dominio público" y exento de control.

Asimismo, aunque el régimen norteamericano se haya liberalizado parcialmente, sigue siendo suficientemente complejo y peligroso como para que los desarrolladores de software libre y los responsables de su distribución sigan restringiendo la disponibilidad de productos de cifrado de alto nivel desarrollados en Estados Unidos, o exigiendo que se desarrollen fuera de dicho país (ni siquiera se permite la contribución de código de desarrolladores de Estados Unidos). Además, una lectura pesimista del reglamento indica que cualquier programa notificado al BIS **y sus versiones siguientes, modificaciones y obras derivadas** caerán bajo las reglas de exportación EAR para siempre. Por lo tanto, corren el riesgo de una modificación de las reglas por parte del gobierno de los Estados Unidos a favor de controles más restrictivos. Esto afectaría a todas las copias del programa distribuidas y a sus modificaciones y derivados y a cualquier aplicación tercera que incorpore el programa. No es de extrañar que no haya proyectos de software libre con criptografía de alto nivel distribuidos desde Estados Unidos. El OpenBSD, por ejemplo, se distribuye desde Canadá.

Web recomendada

La interpretación de la FSF sobre el Acuerdo de Waasenaar es interesante. Se puede consultar en www.gnu.org/philosophy/wassenaar.html.

6. Conclusiones

Aquí termina el tercer bloque del módulo de aspectos legales del software libre –el bloque de las licencias libres. En este módulo hemos tratado de situar los aspectos legales de las mismas y del software libre en general, en un contexto más práctico, así como de vincularlas con otras áreas del derecho no vistas hasta ahora (como los datos personales).

En relación con la vida profesional del estudiante, nos parece importante que pueda evaluar las consecuencias comerciales y tecnológicas de los elementos legales que venimos describiendo y comentando: qué se puede hacer en relación con la eficacia o la ineficacia legal de las licencias, cómo se pueden aprovechar las posibilidades de licencias múltiples, qué documentación o *checklist* podría ser de utilidad, qué estrategia o táctica hay que adoptar frente a una duda legal sobre el software libre y qué proceso de decisión es el más adecuado.

Será interesante para el estudiante retomar las preguntas formuladas y los ejemplos dados en el módulo 1, para ver cuántos puede contestar y resolver, así como en cuántos de ellos es capaz de identificar la dificultad legal y de establecer una estrategia para superarla. Asimismo, en los estudios de caso que analizaremos en los módulos siguientes, veremos un cierto número de situaciones en las que el conocimiento adquirido aquí será de gran utilidad.