

Software libre y sistemas de información geográfica: conceptos, definiciones y aspectos legales.

Malcolm Bain ⁽¹⁾, D. Megías Jimenez ⁽²⁾ y Antoni Perez-Navarro ⁽²⁾

⁽¹⁾ LegisTICs SCP, Consultor del Máster de Software Libre de la UOC y Profesor del Posgrado "Desarrollo en Software Libre" de la Universitat Politècnica de Catalunya, {mbain}@logistics.net

⁽²⁾ Estudis d'Informàtica, Multimèdia i Telecomunicació, Universitat Oberta de Catalunya, Rambla del Poblenou, 156, 08018 Barcelona, {dmegias,aperezn}@uoc.edu.

RESUMEN

Bajo el nombre de software libre (no confundir con gratuito) se esconde todo un abanico de formas de desarrollo, licencias, derechos, etc. que abre diversos interrogantes a quien quiere trabajar con él: ¿Quién es responsable de un proyecto en software libre? ¿Cómo se desarrolla? ¿Quién es el propietario del software, si es que hay alguno? y muchos otros. En este artículo se da respuesta a algunas de estos interrogantes y se muestran las diversas opciones legales que rodean al mundo del software libre: licencias, derechos, estándares abiertos, etc. Todos estos elementos se exponen desde el punto de vista de los sistemas de información geográfica, tocando temas que van desde el desarrollo y mantenimiento de las plataformas SIG de base hasta la implementación de aplicaciones sobre estas plataformas.

Palabras clave: Jornadas, SIG, software libre, Girona, licencias, GPL.

ABSTRACT

This article covers some of the principal legal aspects of free and open source software applied to Geographic Information Systems. Starting out from basic concepts in FOSS licensing, it aims to deal with questions such as "who is responsible for FOSS projects?", "How are they developed?", "who is the owner of the software?" and discusses the legal issues and problems that may arise during the course of creating, distributing or deploying a FOSS-based GIS.

Key words: GIS, free software, licenses, legal issues.

INTRODUCCIÓN

Bajo el nombre de software libre (no confundir con gratuito) se esconde todo un abanico de formas de desarrollo de software, licencias, derechos, etc. que abre

diversos interrogantes a quien quiere trabajar con él: ¿Quién es responsable de un proyecto en software libre? ¿Cómo se desarrolla? ¿Quién es el propietario del software, si es que hay alguno? y muchos otros. En este artículo se da respuesta a algunas de estos interrogantes y se explican diversos aspectos legales que rodean al mundo del software libre (licencias, derechos, estándares abiertos, etc.) aplicado al contexto de los sistemas de información geográfica (SIG).

Primero, definimos y comentamos los conceptos principales del software libre, antes de profundizar en algunos temas legales en relación con la propiedad del software, la compatibilidad de licencias y los modelos de desarrollo y distribución.

CONCEPTOS GENERALES

La principal diferencia entre el software libre y el software propietario, privativo o cerrado, reside en los derechos y obligaciones especificados en la licencia bajo la cual se distribuye el software en cuestión. Aquellos otorgados por las licencias de software libre (“licencias libres”) suelen ser directamente opuestos a los otorgados y reservados por una licencia de software propietaria (“licencia propietaria”), sobre todo en cuanto al uso, la distribución y la modificación del software.

Una licencia de software es el mecanismo legal por el cual el propietario de un software (el “titular”) autoriza a terceros utilizar el programa bajo ciertas condiciones – sin llegar a venderlo. Cuando uno “compra” un software, entonces, en realidad uno compra solamente unos derechos de uso de acuerdo a la licencia aplicable. Esta licencia permite al titular imponer restricciones o condiciones sobre la explotación del programa, como por ejemplo prohibiciones en cuanto a la realización de modificaciones o redistribuciones. De esta manera, el titular del software puede establecer un modelo de negocio basado en la venta de licencias, porque cada usuario no lo puede conseguir libremente (legalmente) sino que necesita conseguir (y comprar) una licencia de uso. Las licencias propietarias o privativas, a menudo denominadas “EULA” (*End User License Agreements*, es decir, licencias de usuario final), de forma habitual tienden a restringir los derechos de los usuarios a su mero uso y el propietario distribuye el software en código binario, que es muy difícil modificar.

Una licencia de software libre, al contrario, debe garantizar al usuario las cuatro libertades enunciadas por la *Free Software Foundation* [1]: el usuario / licenciataria debe gozar de:

0. La libertad de ejecutar el programa.
1. La libertad de estudiar el programa y adaptarlo a sus necesidades (modificarlo).
2. La libertad de distribuirlo.
3. La libertad de mejorarlo y distribuir las mejoras.

Esto se traduce, en términos de contrato de licencia, en otorgar al usuario los derechos de copia, modificación (mejorar, traducir, adaptar, integrar con otros programas) y distribución/redistribución del programa y del código derivado del mismo. Asimismo, para poder ejercer estos derechos, el usuario debe tener acceso al código fuente del programa. Notemos que las libertades 1, 2 y 3 son justamente los actos prohibidos por las licencias propietarias y hasta la libertad 0 se ve restringida por ellas a veces, con respecto a usuarios, países o usos.

Gozando de estas libertades (o derechos), el usuario de software libre en la práctica puede descargarlo de internet, instalarlo, ejecutarlo sin condiciones, modificarlo en función de sus necesidades y volver a distribuirlo (junto con las modificaciones) a terceros.

Es importante destacar el hecho de que mucho software libre es software comercial, es decir software que tiene un precio para su compra, se usa en proyectos comerciales y por empresas que ofrecen servicios comerciales a terceros. La diferencia con el software propietario es que el usuario de este software libre (comercial) recibe no solamente un derecho de uso sino también un derecho de acceso al código fuente y los antes mencionados derechos de copia, modificación y redistribución.

UN ABANICO DE LICENCIAS

El concepto de software libre incluye todo un mundo de diferentes tipos de software, desde los sistemas operativos hasta aplicaciones de infraestructura y de usuario, y este software se distribuye bajo diferentes licencias. Aunque haya una serie de criterios para determinar lo que es una licencia libre, hay varias licencias libres "estándares" que cumplen con estos criterios.

La licencia libre más famosa, la *General Public License*¹ o GPL, es emblemática dentro del movimiento libre y concretiza el concepto de "copyleft". El *copyleft* (un juego de palabras sobre "copyright", los derechos de autor en inglés) es un mecanismo por el cual una licencia no solamente debe ofrecer las libertades y derechos del usuario antes mencionadas, sino que también debe obligar a mantener estas libertades para usuarios posteriores del software: cualquier redistribución del programa y de cualquier modificación del mismo debe estar bajo esta misma licencia, sin restricciones adicionales.

Hay otras 55 licencias que se consideran parte del movimiento de software "libre". De hecho, el número de licencias ha ido creciendo, a medida que diferentes proyectos o personas han redactado licencias con condiciones que corresponden a sus objetivos estratégicos, tecnológicos o filosóficos. Este aumento ha ocurrido sobre todo desde la formalización del concepto de licencias de "código abierto" (*open source software*). Resaltando más la necesidad de disponer del código fuente de un programa, el software de código fuente abierto se distribuye bajo licencias que cumplen criterios determinados por el *Open Source Initiative*², una organización fundada para dar coherencia y voz a la comunidad de software "open source" y darle una imagen más amena para el mundo empresarial y comercial. Estos criterios son similares a las cuatro libertades ya mencionadas y resaltan los derechos de uso sin discriminación, modificación ni redistribución. Sin embargo, no obligan a imponer obligaciones de *copyleft*. Las licencias de código abierto incluyen algunas que permiten cualquier uso del software, hasta incorporarlo en software propietario - lo que sería imposible con software *copyleft*.

Según cada licencia, la mayor diferencia se sitúa en las condiciones sobre la redistribución, una actividad fundamental para integradores y consultores de software ya que redistribuyen el software a sus clientes. El abanico de obligaciones respecto de la redistribución va desde unas mínimas, que obligan únicamente a mantener el aviso de autoría y la limitación de garantías y de responsabilidades (el *disclaimer*), hasta el "máximo" (en cierto sentido) de la cláusula *copyleft* de la GPL que, como hemos comentado, obliga a distribuir cualquier modificación y obra derivada bajo la misma licencia GPL (junto con el acceso al código fuente).

De manera resumida, podemos decir:

- Si el programa se ha licenciado bajo una licencia "permisiva" o académica, por usarse preponderantemente en entornos universitarios (de tipo BSD – *Berkely software Distribution* o *Apache*), generalmente no hay condiciones sobre la

1 <http://www.gnu.org/copyleft/gpl.html>

2 <http://www.opensource.org/>

redistribución, lo que permite mezclarlo con cualquier otro software y hasta “privatizar”lo.

- Si el programa se ha licenciado bajo la licencia GPL u otra licencia "con *copyleft*", entonces la redistribución ha de realizarse bajo la misma licencia, y ofrecer acceso al código fuente al destinatario del software. Esto asegura que el programa quede libre y no pueda ser privatizado.
- Algunas licencias “intermedias”, como la LGPL (*Lesser GPL*) o la MPL (*Mozilla Public License*) permiten agregar y/o vincular código nuevo al código original y distribuir el conjunto bajo una licencia nueva (propietaria o libre). Sin embargo, la parte original tendrá que ofrecerse bajo su licencia original, normalmente con el código fuente. Es una forma de *copyleft* parcial.

En la tabla siguiente, se puede ver una clasificación preliminar de los diversos tipos de licencia.

Tabla1: Tipos de licencia libre

Licencia	Se puede mezclar con otro software no libre	Se pueden privatizar las obras agregadas	Se pueden privatizar las modificaciones	No tiene restricciones sobre el uso ni la distribución	<i>Copyleft</i> : hay que redistribuir bajo la misma licencia
GPL					X
LGPL	(X)				(X)
MPL	(X)	X			(X)
BSD	X	X	X		
Dominio Público	X	X	X	X	

Fuente: Aspectos legales y de explotación del software libre, UOC/FUOC [2]. (X) significa “parcialmente”

En el mundo de los SIG libres, se usan varias licencias, aunque predomina la GPL. El gvSIG³, GRASS⁴ y el SAGA-GIS⁵ usan esta licencia, pero SAGA aplica la licencia un poco más permisiva, la LGPL, para su API, para permitir a los usuarios distribuir bajo licencia propietaria sus propios módulos que interactúan con SAGA. GeoTools⁶ también usa esta licencia LGPL para su *toolkit*. El Geospatial Data Abstraction Library (GDAL⁷), por otro lado, usa una licencia permisiva MIT que permite a terceros incorporar esta librería en sus productos cerrados. Otro ejemplo, el editor cartográfico OrbisCAD⁸, se distribuye bajo la licencia permisiva, la licencia Apache.

EFFECTOS DE LA LIBERTAD

Como hemos visto, los derechos otorgados a los usuarios de software libre, sean usuarios finales o intermediarios como los consultores o integradores de software, permiten instalar y ejecutar el programa, copiarlo, modificarlo y redistribuirlo junto con las modificaciones. Asimismo, el usuario tiene derecho a obtener el código fuente de

3<http://www.gvsig.gva.es/>

4<http://grass.itc.it/>

5<http://www.saga-gis.uni-goettingen.de/html/index.php>

6<http://www.geotools.org/>

7<http://www.gdal.org/>

8<http://orbiscad.sourceforge.net/>

la aplicación, tanto para corregir errores como para ejercer sus derechos de modificación y adaptación. Estos derechos tienen varias consecuencias a nivel de desarrollo, distribución y uso del software libre, que comentamos en el contexto de los SIG.

Primero, permiten a muchas personas o entidades participar en la creación de un programa libre. Esto cambia la dinámica y la gestión del desarrollo, ya que los usuarios finales del producto pueden participar no solamente como “testers” o para corregir errores, sino también para ofrecer recomendaciones de mejoras, adaptaciones y hasta nuevos módulos o componentes que corresponden a necesidades particulares. El proyecto ILWIS⁹, por ejemplo, ha decidido liberar el software ILWIS (dentro de la iniciativa 52° North) “para crear mejores oportunidades para la reutilización e implantación de las funcionalidades ofertadas por los SIG dentro de una comunidad más amplia”. Cambia también el régimen de propiedad, porque los propios contribuidores suelen mantener la propiedad del software, contribuyéndolo al proyecto bajo licencia libre o acuerdo sobre contribuciones (que permite al responsable del proyecto redistribuir la contribución bajo la licencia del proyecto).

También este modelo “libre” suele implicar que el software sea más modular, puesto que diferentes personas (en distintas partes del mundo) pueden crear diferentes partes del software. Un software libre no es necesariamente un solo programa, sino una serie de componentes, núcleos para un sistema operativo u otro, “plug-ins”, módulos adicionales, interfaces y extensiones, según vayan contribuyendo los participantes en la comunidad o proyecto en cuestión. Consideramos el proyecto gvSIG¹⁰, que tiene tres partes diferenciadas: ANDAMI, FMAP y GUI. Desde la perspectiva del desarrollador, hay otras tres divisiones: Las librerías genéricas que pueden ser empleadas en cualquier otro proyecto SIG (FMap, Cresques, WMSCClient), un *Workbench* extensible utilizable para cualquier tipo de aplicación MDI, esté o no relacionada con el SIG (ANDAMI) y un *Plugin* que usa las librerías y que convierte al *Workbench* en la aplicación gvSIG (gvSIG).

Otro efecto: la multiplicidad de componentes y desarrolladores significa que la única manera de crear un software que funcione correctamente y sea interoperable es realizarlo de acuerdo con estándares aceptados internacionalmente dentro del sector, en cuanto a interfaces, protocolos, formatos, etc.

El software libre por lo tanto suele ser más modular, más distribuido y más estandarizado. Estos cambios en el modelo de creación de software constituyen un aspecto importante: habitualmente no hay proyectos libres monolíticos, gestionados por un “centro” o un jerarquía, sino comunidades horizontales de contribuidores, coordinados por equipos (más o menos grandes) de líderes que empaquetan y distribuyen las versiones integradas (*releases*). ¿Menos control, menos calidad? No necesariamente: los *gatekeepers* – las personas que controlan el código que compone el software – ejercen un control estricto sobre lo que entra en una *release*, normalmente sujeto a controles de calidad técnica y legal.

Desde la perspectiva del integrador o usuario final sofisticado (que realiza sus propias adaptaciones), la libertad de uso de las librerías de software libre permite su reutilización en software tanto libre como propietario, tanto proyectos universitarios o públicos como comerciales. IDRISI¹¹ usa librerías Gstat, por ejemplo. GeoTools¹² se distribuye bajo licencia permisiva MIT para permitir su integración en varias herramientas SIG.

9<http://www.itc.nl/ilwis/>

10<http://www.gvsig.gva.es/>

11<http://www.clarklabs.org/>

12<http://geotools.codehaus.org/>

Finalmente desde la perspectiva del usuario, una faceta importante del software libre es la libertad que tiene el usuario frente a sus proveedores: el usuario de software libre, sobre todo si es software bajo una licencia *copyleft*, tiene acceso al código fuente y derechos de copia y modificación. Esto le permite cambiar de proveedor, o modificar y mejorar el software de manera interna, en caso de desacuerdo con el proveedor original o de su desaparición. En general, esto tiene como efecto positivo el de obligar a los intermediarios de software a mejorar sus servicios y su atención al cliente para no perder el cliente: éste, teniendo el código fuente y derechos de explotación del software mayores que el mero uso, no está más cautivo de su proveedor y puede romper la relación comercial y elegir otro consultor o contratar el personal necesario para hacer desarrollos internos.

Finalmente, por el efecto de la distribución libre del software, éste suele ser gratuito. No sirve de nada poner un precio alto a un software, si mañana cualquier usuario puede legítimamente publicar el código (fuente y binario) en Internet para su descarga, casi sin coste para cualquier internauta. Esto no impide que haya mucho software libre en Internet que sea de pago. Redhat, por ejemplo, puede cobrar por su distribución de GNU/Linux, principalmente porque ofrece productos y servicios adicionales: garantías de funcionamiento, servicios de ayuda y soporte, aplicaciones administrativas o de instalación, etc.

Retos

Estas libertades y derechos no significan que todo lo relacionado con el software libre esté libre de riesgos o dificultades legales. Desde una perspectiva legal, podemos comentar dos de los principales retos actuales: las incompatibilidades entre licencias y el problema de las licencias sobre contenidos. Una pregunta frecuente es ¿cómo integrar varios módulos o herramientas de software libre en “soluciones” más complejas, sin infringir las licencias sobre cada componente? El usuario final, que no redistribuye su software, tiene libertad (sin condiciones) para el uso y la modificación/adaptación del software para crear sistemas complejos. El integrador/consultor, al contrario, distribuye los programas como “soluciones” a sus clientes. Como se ha mencionado más arriba, es a la hora de la redistribución cuando surgen las obligaciones de las licencias libres y éstas pueden entrar en conflicto las unas con las otras: si hay que usar una licencia u otra para el código, si hay que proveer el código fuente, si se puede privatizar, si se debe mantener una marca u otro signo distintivo sobre el software, etc.

Por otro lado, se han creado una serie de licencias libres para contenidos (textos, datos), que no son necesariamente compatibles con las licencias de software. Por lo tanto, de manera similar al punto anterior, su integración y distribución junto con sistemas informáticos complejos, como los SIG o traductores, dependerá del grado de compatibilidad entre las respectivas licencias. Afortunadamente, la licencia sobre el SIG normalmente no afectaría la licencia sobre datos y viceversa, ya que no son elementos “integrados” sino compatibles por su formato (respetando un estándar).

ASPECTOS LEGALES PARTICULARES PARA SIG

Estos aspectos y dificultades legales surgen también en el mundo de los SIG. Comentamos tres en particular: la modularidad y los conflictos de licencia, la selección de licencia para un software nuevo o un SIG para un cliente, y la propiedad del software libre y de los desarrollos sobre software libre.

En los SIG confluyen diversos elementos, cada uno de los cuales puede verse afectado por licencias distintas. A grandes rasgos, podríamos dividir un SIG en tres grupos. En el primero está el programa base, que correspondería al SIG genérico de base (ejemplos: Geomedia, ArcGIS, Smallworld, gvSIG, etc.). Luego están las

I Jornadas de SIG Libre

aplicaciones de usuario, o desarrollos personalizados, que corresponderían a las diversas personalizaciones desarrolladas sobre el SIG de base para adaptarlo a las necesidades propias del usuario. Un ejemplo sería desarrollar una aplicación de distribución de aguas, de electricidad o de gas sobre alguna de los SIG “bases” anteriores. Finalmente están los datos de base, que contienen los datos geoespaciales sobre los que se ejecuta el SIG. Así por ejemplo, mientras que una compañía de electricidad podría tener, como información propia, los cables y los datos de los clientes, éstos estarían representados sobre una cartografía de base con las manzanas y las calles.

Generalmente, estas tres partes se distribuirían bajo licencias distintas. Así pues, sería perfectamente posible que una compañía partiera de un SIG libre y sobre él llevara a cabo sus propios desarrollos y contratara otros desarrollos y datos de base para terceros. Estos desarrollos podrían ser propietarios o libres, según la licencia que se les aplicara. Para la persona que integra los componentes, los adapta y los distribuye al cliente final, será importante garantizar que no haya ningún conflicto entre las licencias sobre los componentes.

Por otro lado, entre las preguntas fundamentales que debe hacerse un responsable del desarrollo de un SIG libre (o el proveedor de una aplicación basada en un SIG libre), es qué tipo de licencia libre puede usar para su producto, y por lo tanto, qué componentes puede integrar dentro del mismo. Un desarrollo a medida para un cliente, integrando conocimientos específicos del proveedor y sus “secretos de negocio” podría llegar a publicarse si se usaran componentes bajo una licencia con *copyleft*, por ejemplo en el SIG de base. Sin embargo, no será siempre así: según la naturaleza de las licencias sobre los componentes, sería posible tener desarrollos tanto propietarios como libres: los datos cartográficos de dominio público o bajo licencias permisivas de tipo BSD, junto con librerías u otros componentes “base” bajo licencias similares, permiten usar el software para crear aplicaciones propietarias. Esta dificultad no debe impedir el uso de software libre: al contrario, su existencia promueve la reutilización de componentes de alta calidad disponibles para toda la comunidad. Lo que sí impacta es al modelo de negocio de este responsable o proveedor, la fuente de su ventaja competitiva, y su relación con sus clientes.

La titularidad del software es otro tema importante, fuente de negociaciones a la hora de concluir acuerdos sobre el desarrollo de un SIG. Si el cliente desea ser propietario de cualquier desarrollo realizado por un consultor o integrador, deberá ser consciente de que si se usan componentes libres, solamente adquirirá propiedad sobre los desarrollos hechos a medida. Los componentes quedan bajo la titularidad del autor o titular original y esto debe estar claro en la relación entre consultor y cliente. Las libertades concedidas en una licencia libre son amplias, pero no constituyen propiedad en el software y el cliente deberá respetar las obligaciones en el caso de cualquier redistribución posterior. Si estos desarrollos son sobre software *copyleft* bajo la licencia GPL, entonces aunque el cliente pueda recibir la propiedad del desarrollo (que no del componente original), su redistribución debe realizarse de acuerdo con esta licencia.

Finalmente, debemos comentar brevemente otro tema fundamental en el contexto de los SIG, que es el acceso a los datos geoespaciales de base. Existen una serie de bases de datos libres (y gratuitas), como por ejemplo el “Global Self-consistent, Hierarchical, High-resolution Shoreline Database” (GSHHS)¹³, muchos de los cuales están distribuidos bajo licencias permisivas (para maximizar su uso) o son de dominio público (permitiendo su uso por cualquiera para cualquier fin).

13<http://www.ngdc.noaa.gov/mgg/shorelines/gshhs.html>

CONCLUSIONES

Aunque los SIG más usados y desarrollados sean software propietario, hay cada vez más SIG – o componentes para SIG – que se distribuyen bajo licencias libres y que empiezan a crear un “ecosistema” de software libre para SIG. Participan en este movimiento tanto organizaciones universitarias (a menudo la primera fuente de software libre), como entidades comerciales y, por supuesto, individuos. Recientemente se ha creado una fundación llamada “*Open Source Geospatial Foundation*”¹⁴ cuya misión es apoyar y crear software SIG libre de alta calidad. Por lo tanto a pesar de las diferencias y especificidades de este sector, se podría preguntar si los SIG libres no van a tener un futuro tan positivo como el resto de software libre, que se encuentra en plena expansión. En este artículo se ha intentado resaltar algunos de los aspectos legales relevantes para los SIGs libres – tanto proyectos como software – y destacar algunos retos y puntos importantes a tener en cuenta a la hora de crear un software SIG libre o implementar una solución basada en uno. La multiplicidad de licencias y tipos de colaboraciones significa que se debe considerar cuidadosamente el impacto de los aspectos legales sobre los modelos de desarrollo, los modelos de negocio, los servicios a brindar y las relaciones con los eventuales clientes de estos sistemas.

REFERENCIAS

- ◆ STALLMAN, R (1996) *La Definición de Software Libre*. <http://www.gnu.org/philosophy/free-sw.es.html>
- ◆ BAIN,M; GALLEGO M; MARTINEZ RIBAS, M; RIUS, J (2004) *Aspectos legales y de explotación del software libre*. Barcelona, FUOC.

¹⁴<https://www.osgeo.org/>