

Introducción al software libre en general y a los SIG libres en particular

D. Megías⁽¹⁾, A. Pérez-Navarro⁽¹⁾, y M. Bain⁽²⁾

⁽¹⁾ Estudis d'Informàtica, Multimèdia i Telecomunicació, Universitat Oberta de Catalunya, Rambla del Poblenou, 156, 08018 Barcelona, {dmegias,aperezn}@uoc.edu.

⁽²⁾ LegiTICs 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@legistics.net.

RESUMEN

En este documento se analizan los conceptos clave relacionados con el software libre: ¿qué es el software libre? ¿qué es el software propietario? ¿qué es el software gratuito? ¿qué es una licencia copyleft? A continuación se muestra cómo las aplicaciones horizontales libres se desarrollan de forma distinta a las verticales libres, entre las que se encuentran los SIG. Seguidamente se detallan las características de los sistemas SIG, centrándose en los SIG libres. El hecho de ser una aplicación vertical condiciona y explica buena parte de las particularidades de los SIG libres. Estas particularidades se analizan a partir de las impresiones de profesionales de los SIG. Todo ello lleva a la conclusión que los SIG libres no pueden depender sólo de una comunidad de desarrolladores y necesitan un respaldo económico importante.

Palabras clave: *Software libre, Copyleft, Aplicaciones verticales y horizontales, SIG.*

INTRODUCCIÓN

Hace unos años el software libre era algo propio de sectores muy concretos, para usuarios con conocimientos informáticos muy avanzados y con interfaces de usuario poco o nada desarrolladas. Para la sociedad, en general, software libre era sinónimo de gratis y GNU/Linux era un sistema operativo del que poco o nada se había oído hablar.

La situación actual es otra: algunas aplicaciones libres se han puesto a la par o incluso han superado sus correspondientes propietarias. Cada día más empresas apuestan por desarrollos de software libre y GNU/Linux, por ejemplo, ha empezado a estar presente ya, no sólo en entornos profesionales, sino también en muchos hogares. A pesar de todo ello, aún hay un cierto desconocimiento, dudas y confusiones en lo que respecta al software libre.

Los SIG también se han visto inmersos en esta evolución del software libre. En pocos años se ha pasado de una situación en que los SIG libres no eran una opción, a la presencia en el mercado de diversas soluciones totalmente válidas y funcionales.

Aún así, da la impresión de que no acaban de despegar o que este despegue está siendo más lento de lo que cabría esperar.

Estos elementos, el software libre en general y los SIG libres en particular, son los que se tratan en el presente artículo. Éste se estructura en dos partes: en la primera parte se trata el software libre y los conceptos clave relacionados con el mismo. En la segunda, se analiza el porqué de este lento despegue de los SIG libres.

EL SOFTWARE LIBRE

En este primer apartado se presenta una introducción al software libre. Se comienza diciendo qué significa que un software sea libre y resolviendo algunas dudas sobre este concepto. A continuación se explican las diferencias entre el software libre y el privativo o cerrado. Una vez entendido qué es el software libre y qué lo distingue del propietario, se explica un concepto clave en el mundo del software libre como es el de *copyleft*. Finalmente, se muestra el papel que juega el software libre en nuestra sociedad.

Definición de software libre

El software se ha convertido en un factor estratégico en las últimas décadas para muchos sectores económicos. El surgimiento del software libre (*Free Software*) y su penetración en los sectores fundamentales del mercado tecnológico está produciendo unos cambios drásticos en la economía del desarrollo y del uso del software.

Se entiende por software libre a aquél que garantiza al usuario final las cuatro **libertades** siguientes [1]:

0. Libertad para ejecutar el programa en cualquier sitio, con cualquier propósito y para siempre.
1. Libertad para estudiarlo y adaptarlo a nuestras necesidades. Esto exige el acceso al código fuente.
2. Libertad de redistribución, de modo que se nos permita colaborar con vecinos y amigos.
3. Libertad para mejorar el programa y publicar las mejoras. También exige el código fuente.

El software libre proporciona libertad para aprender, libertad para educar, libertad para competir y libertad para opinar. El término *free* en inglés no debe, sin embargo, confundirse con la acepción de “gratis”. De hecho, los sistemas de software libre pueden, perfectamente, ser aplicaciones comerciales de manera que hay que pagar por su adquisición o bien por ciertos servicios de valor añadido, como mejoras o soporte técnico. Por lo tanto, no debe usarse el término “software comercial” como opuesto al software libre, dado que el software libre puede ser usado comercialmente (cosa que sucede a menudo).

El software libre desplaza la estrategia del mercado del software del producto hacia los servicios, lo que puede permitir a Europa y a otros territorios (como por ejemplo a Iberoamérica) convertirse en líderes mundiales en el mercado del software.

Definición de software propietario, privativo o cerrado

A los programas que no respetan completamente este “código de libertades” se los conoce como software propietario, cerrado o privativo. Por ejemplo, un programa que

garantice las cuatro libertades, pero que prohíba el desarrollo de una aplicación comercial a partir de su código, no se consideraría software libre, dado que impondría una restricción en la libertad 3.

La forma más habitual de software privativo corresponde a un programa que se distribuye únicamente en código binario (ejecutable por el ordenador, pero no inteligible para los seres humanos) y que explícitamente prohíbe su redistribución o copia (libertad 2), su modificación y la publicación de mejoras (libertad 3). Este tipo de programas, al no distribuir el código fuente, tampoco permite estudiarlos ni adaptarlos a las propias necesidades (libertad 1). Las restricciones del software propietario pueden alcanzar, incluso, a la libertad 0, la más básica. Algunas aplicaciones incluyen cláusulas en sus licencias que impiden usarlas en determinados países o con algunos fines concretos (como criticar al fabricante del programa), lo que constituye un ejemplo de limitación en la primera de las libertades que el software libre sí garantiza a los usuarios.

El concepto de *copyleft*

De manera informal, “*copyleft*” vendría a ser lo opuesto al concepto de “*copyright*”. Las licencias de tipo *copyleft*, como todas las licencias, usan los derechos de autor para imponer ciertas restricciones en el uso de un programa (o, en general, de una obra). En este caso, las restricciones consisten en imponer que cualquier programa que se derive del original publicado bajo una licencia *copyleft* se publique, a su vez, con una licencia compatible con la primera, que otorgue los mismos derechos y libertades que proporciona la licencia del programa original. Esto es, los programas derivados deben permitir a los usuarios copiar, modificar y realizar nuevos programas derivados [2]. Por este motivo, a este tipo de licencias se las conocen como “recíprocas” y se dice que incluyen una “cláusula viral” que afecta a todas los programas que se deriven del original. De un modo más resumido, podemos decir que las licencias de tipo *copyleft* evitan que un programa que se distribuya como libre acabe originando un programa propietario o cerrado en el futuro como fruto de un abuso de las libertades que otorga el software libre.

No todas las licencias de software libre incluyen este tipo de cláusulas virales. Existen algunas licencias, denominadas “permisivas”, que no impiden que un programa derivado se publique con una licencia no libre (o propietaria). De todos modos, las licencias de tipo *copyleft* gozan de gran popularidad en las comunidades de software libre, siendo la GNU General Public License¹(GPL) la más utilizada en los proyectos de desarrollo de software libre.

¿Qué necesidades cubre el software libre?

El software libre ha adquirido un grado muy elevado de popularidad en los últimos años. Su uso se ha ido extendiendo desde las comunidades de desarrolladores de software libre, a menudo integradas por expertos (*hackers*) con un elevado grado de conocimientos técnicos, a todo tipo de usuarios, empresas y administraciones públicas. Además, las ventajas que representan las cuatro libertades garantizadas por el software libre, lo han convertido en una herramienta muy poderosa para la educación, motivo por el cual la penetración del software libre en el mundo académico universitario ha experimentado un aumento considerable en la última década.

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

Hoy en día, existe una plétora de aplicaciones libres disponibles para los usuarios, que prácticamente ven cubiertas la totalidad de sus necesidades generales con estos programas. Por citar algunos ejemplos, algunas aplicaciones muy populares son el GNU/Linux, Gnome², KDE³, OpenOffice.org⁴, Mozilla Firefox⁵, Mozilla Thunderbird⁶ y Apache⁷.

El sistema operativo GNU/Linux se ha erigido como una potente herramienta tanto en el *back-office* (o servidor) como en el escritorio. Este sistema operativo, en constante evolución, destaca por su estabilidad y eficiencia, y distribuciones como Ubuntu⁸ han permitido a todo tipo de usuarios familiarizarse con uno de los proyectos estrella del software libre. Otras aplicaciones de gran éxito son los escritorios gráficos Gnome y KDE, ambos presentes en la práctica totalidad de las distribuciones actuales de GNU/Linux. Más allá del sistema operativo y de los escritorios gráficos, encontramos una familia de aplicaciones “horizontales” que constituyen las herramientas básicas para todo usuario, incluyendo la *suite* ofimática OpenOffice.org, el navegador Mozilla Firefox o el cliente de correo Mozilla Thunderbird. En el lado del servidor, existen centenares de aplicaciones de éxito, algunas tan relevantes como el servidor de web Apache. En definitiva, casi todas las necesidades de los usuarios no especializados están cubiertas con aplicaciones libres que han demostrado encontrarse entre las mejores en su ámbito.

Por otro lado, nos encontramos con las aplicaciones denominadas “verticales”, aquéllas que cubren necesidades esporádicas de pequeños sectores. Este segundo tipo de aplicaciones, se caracteriza por tener un conjunto de usuarios potenciales muy reducido (en comparación con los millones de usuarios de OpenOffice.org, por ejemplo). Éste sería el caso de los sistemas de información geográfica (SIG), un tipo de aplicación que va dirigida a un sector empresarial bastante delimitado y no al público en general. En lo referente a las aplicaciones verticales, los desarrollos de software libre son menos abundantes. Las aplicaciones horizontales tienen una comunidad de usuarios tan grande que es muy simple para estos proyectos captar desarrolladores (voluntarios o remunerados) que aporten su experiencia y contribuyan al proyecto. Por otro lado, en las aplicaciones verticales, la comunidad de usuarios que hay detrás de ellas es muy reducida, lo que repercute en una evidente dificultad para captar e involucrar desarrolladores en proyectos libres de este ámbito. De hecho, a menudo son los propios usuarios de las aplicaciones los que ponen en marcha los proyectos de software libre y esto limita mucho las posibilidades de iniciar proyectos de aplicaciones verticales.

El hecho de que las aplicaciones SIG se hallen entre las denominadas verticales no implica, sin embargo, que no existan algunas opciones de SIG libres de gran calidad. A lo largo de este artículo y, en general, en estas Jornadas, se presentan algunas de las soluciones SIG libres más destacadas.

Respecto a la dificultad del desarrollo de aplicaciones verticales libres, algunos proyectos de muestran cómo es posible solventar este obstáculo con la creación de consorcios formados por usuarios de un determinado tipo de aplicación vertical que unen sus esfuerzos (y, a menudo, aportan capital) para el desarrollo cooperativo del

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

3 <http://www.kde.org>

4 <http://www.openoffice.org>

5 <http://www.mozilla-europe.org/es/products/firefox/>

6 <http://www.mozilla-europe.org/es/products/thunderbird/>

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

8 <http://www.ubuntu.com>

proyecto. En algunos casos, estos consorcios cuentan con participación de entes públicos. Un ejemplo de ello es el acuerdo suscrito por la mayoría de las universidades catalanas con el antiguo Departament d'Universitats, Recerca i Societat de la Informació (DURSI) de la Generalitat de Catalunya para desarrollar un entorno virtual de aprendizaje (campus virtual) libre. Sin duda, ésta sería una buena fórmula para iniciar proyectos verticales de software libre, entre los cuales podría enmarcarse el desarrollo de un SIG libre de altas prestaciones que cumpla con los requisitos necesarios para los integrantes de un determinado consorcio.

LOS SIG LIBRES

Hasta aquí se han visto los diversos elementos que hacen referencia al software libre y cómo los SIG se enmarcan en este ámbito. Es en estas aplicaciones en las que se centra este apartado. Para ello se empieza definiendo qué es un SIG. A continuación, se describe la evolución social que estas aplicaciones han tenido los últimos años. Este punto es clave para entender el porqué del despegue de los SIG libres que se está produciendo actualmente. El último apartado se centra específicamente en los SIG libres. Este apartado se ha elaborado a partir de las impresiones de profesionales de estos sistemas, es decir, cómo ven estos profesionales los SIG libres actuales.

¿Qué es un SIG?

Los SIG son un tipo de aplicaciones muy particulares que aglutinan elementos de diversos ámbitos, como la física, las matemáticas, la informática, la cartografía y la topografía, entre otros. De hecho, si se pueden encontrar muchas y muy variadas definiciones[3]:

1. Base de datos computerizada que contiene información espacial [4].
2. Sistema basado en el ordenador que proporciona los siguientes cuatro conjuntos de capacidades para la manipulación de datos georeferenciados: entrada de datos, gestión de los datos (almacenamiento y recuperación), manipulación y análisis y salida de datos [4].
3. Sistema de hardware, software y procedimientos diseñados para realizar la captura, almacenamiento, manipulación, análisis, modelado y presentación de datos referenciados espacialmente para la resolución de problemas complejos de planificación y gestión [4].
4. Software utilizado para automatizar, analizar y representar datos gráficos georeferenciados y organizados según un modelo tipológico [5].
5. Modelo informatizado del mundo real, descrito en un sistema de referencia ligado a la tierra, creado para satisfacer unas necesidades de información específicas respondiendo a un conjunto de preguntas concretas [5].

Vemos que algunas de estas definiciones son más restrictivas que otras: algunas entienden por SIG sólo el software mientras otras incluyen en la definición también el hardware y los datos. En sentido amplio, podríamos decir que un SIG se compone de los elementos siguientes:

- Hardware: que hace referencia tanto al ordenador como a elementos de entrada de datos (tablas digitalizadoras, escáneres, etc.) y de salida (impresoras, *plotters*, etc).
- Software: a menudo se identifica esta parte con el SIG, como se ha visto en alguna de las definiciones anteriores. Un software de SIG se compone de: un

manipulador de información geográfica, un gestor de base de datos, una herramienta de gestión y una interfaz gráfica. Normalmente estos elementos se encuentran en un programa base, sobre el cual se llevan a cabo desarrollos específicos.

- Datos: pueden ser alfanuméricos, gráficos o gráficos y alfanuméricos. Incluyen tanto los datos de base (como la cartografía) como los datos de negocio.
- Usuarios: hay que tener en cuenta que un SIG ofrece muchas posibilidades y es el usuario el responsable de sacar partido de ellas.
- Reglas de negocio: un SIG se integra normalmente en una estructura más amplia y se rige por las reglas de negocio de esa estructura.

En lo que a software libre se refiere, los puntos afectados son los que hacen referencia al software y a los datos. A pesar de que el debate para la liberación de los datos está muy activo actualmente, el presente artículo se centrará en la parte del software.

Evolución social de los SIG

Como se ha dicho anteriormente, el software SIG normalmente está formado por un programa base que aporta todos los componentes de software propios de los SIG y por desarrollos a la carta para adaptar el programa a las reglas de negocio y a las necesidades de los usuarios. Las licencias de SIG de base propietarios, como ArcGIS de ESRI⁹, la familia Geomedia d'Intergraph¹⁰ o Smallworld de General Electric¹¹ tienen un coste elevado. Los desarrollos SIG también son caros, por un lado porque hacen falta profesionales con conocimientos muy específicos y, por el otro, porque no son desarrollos estándar sino que normalmente llevan implícita cierta investigación. Finalmente, la cartografía propietaria y su mantenimiento también es cara.

Hasta aproximadamente el año 2000, el escenario propietario era prácticamente el único y, debido al elevado coste de implantar un SIG, este tipo de aplicaciones era prácticamente un producto de lujo al alcance, básicamente, de medianas y grandes empresas o instituciones públicas. En muchos casos, además, los SIG no jugaban un papel central, sino que eran aplicaciones aisladas del resto de sistemas. Pero a principios de esta década se inició un proceso de cambio que sigue en marcha. Los aspectos en los que se ha focalizado este proceso podrían resumirse en los siguientes puntos:

- **Interoperabilidad de las aplicaciones:** las compañías e instituciones que disponían de SIG aislados han decidido integrarlos en el resto de aplicaciones. Ya no sólo es necesario disponer de un programa de calidad y eficiente con datos de calidad, sino que se requiere que ese programa sea compatible con el resto de sistemas informáticos de sistemas de la compañía, de manera que no haga falta duplicar información. Así, a los SIG de ahora se les pide, por ejemplo, que permitan almacenar los datos en bases de datos como Oracle, Microsoft SQL Server o MySQL y, además, que puedan importar y exportar información a formatos gráficos estándar de forma rápida y eficiente. En este sentido, los SIG que ofrecían bases de datos propietarias, a pesar de la extraordinaria calidad de

9 <http://www.esri.com/> [Febrer 2007]

10 <http://www.intergraph.com/> [Febrer 2007]

11 http://www.gepower.com/prod_serv/products/gis_software/en/index.htm [Febrer 2007]

algunos, han perdido terreno en favor de otros que son más fácilmente interoperables.

- **Acceso web:** es necesario acceder a la información desde cualquier punto e, incluso, publicar alguna información en Internet. Sería lo que podríamos llamar “el efecto Internet”. En este sentido los programas SIG tradicionales han ofrecido diversas soluciones, con más o menos fortuna, para permitir el acceso web a los SIG.
- **Liberación de la cartografía:** la cartografía de base era un elemento que encarecía considerablemente los SIG. Sin embargo, el abaratamiento de ésta, e incluso su gratuidad en algunos casos, han reducido considerablemente la inversión inicial necesaria para implantar un SIG.
- **La eclosión social de los SIG:** quizás el cambio más importante que se ha producido en los SIG en los últimos años haya sido su eclosión social. Como se ha dicho anteriormente, hasta hace pocos años los SIG eran elementos propios de algunas grandes empresas, pero esto ha cambiado como lo demuestran, primero, las páginas web que indican la mejor ruta de un punto a otro sobre un mapa y, más adelante, los navegadores GPS y aplicaciones como Google Maps o Google Earth. Todos estos sistemas, sin ser SIG en sí mismos, han convertido en cotidianas palabras como cartografía, coordenadas UTM o posición GPS.

La conclusión que se puede extraer de todos estos puntos es que, por un lado, los SIG actuales han de ser compatibles con las bases de datos y los formatos estándar y poder trabajar desde Internet y, por el otro, que el mundo SIG está dejando de ser patrimonio exclusivo de empresas e instituciones.

Los SIG y el software libre

Mientras en el mundo de los SIG se producían los cambios indicados en el punto anterior, el software libre iba evolucionando. Proyectos como Ubuntu, Mozilla u OpenOffice.org, por citar sólo algunos ejemplos, han dado credibilidad al software libre y, poco a poco, van ganando terreno ante soluciones propietarias. Ha sido también en estos años en los que han aparecido soluciones de SIG libre de cierta envergadura, como las que se verán en estas jornadas.

Los cambios indicados en el apartado anterior, junto con la aparición de herramientas de más y mejores SIG libres ha llevado a la proliferación de los SIG y a que cada vez haya más empresas e instituciones que dispongan de su propio SIG.

Aún así, todavía reina un cierto escepticismo respecto a ellos en el mundo de los profesionales SIG. Normalmente las empresas e instituciones no desarrollan sus propios SIG, sino que contratan a consultoras especializadas para que lleven a cabo el desarrollo. Para escoger la plataforma SIG, los usuarios pueden escogerla ellos mismos o bien contratar a una consultora para que lleve a cabo el estudio correspondiente y elija la mejor solución. Una vez determinada la plataforma, el proyecto puede llevarlo a cabo la misma consultora u otra. En algunas ocasiones, el proyecto sale a concurso y las consultoras que se presentan proponen en su oferta una u otra plataforma. Tampoco hay que olvidar que a menudo las empresas son conservadoras y prefieren esperar a que la tecnología esté más consolidada a la hora de implantarla.

La impresión general es que los SIG libres aún están lejos de las soluciones propietarias que ofrecen empresas como ESRI, Intergraph o General Electric, a pesar

I Jornadas de SIG Libre

de que para algunas soluciones concretas pueden estar a la par. La pregunta es ¿se decantarán las consultoras por el software libre en igualdad de condiciones con el propietario? La respuesta a esta pregunta no es única, y los SIG libres provocan en el mundo profesional la misma división que provoca el software libre en general. Si preguntamos a profesionales del SIG qué piensan de esta tecnología encontramos argumentos a favor y en contra.

Los principales argumentos en contra que se achacan a los SIG libres son los siguientes:

- Hay demasiados productos y con bastantes limitaciones, inacabados o, simplemente, con muchos fallos. Muchos quedan en estado larvario. Faltan proyectos serios y continuados del estilo de gvSIG, por poner un ejemplo.
- Da la impresión que falta unificar criterios y que a veces se hace lo mismo varias veces.
- La interfaz de usuario está, por lo general, poco trabajada (aunque no siempre).
- Ofrecen sólo funcionalidades muy básicas, y para hacer algo sofisticado hay que partir desde cero. Tienen ciertos aspectos muy desarrollados pero otros ni siquiera existen. Así, dan la impresión de quedarse pequeños al menor intento de ampliación o de personalización, lo que para una empresa integradora de SIG es muy costoso. La impresión es que llevar a cabo los desarrollos adicionales propios de los SIG suele ser muy complicado.
- Suelen ser útiles sólo para cosas muy concretas y básicas y quizás, complementando a alguna de las plataformas más usadas. En cualquier caso, es obligado verificar muy bien para qué sirve y qué funciona correctamente.
- A menudo no hay una organización detrás que responda del software.

Los defensores argumentan que:

- El software propietario también puede generar los mismos problemas de falta de continuidad que el software libre. No es extraño que una empresa abandone un producto y deje de dar soporte a los usuarios del mismo.
- Los componentes de descubrimiento y desconocimiento de una aplicación libre son las mismas que lleva implícitas cualquier aplicación (libre o no). En este sentido, el problema sería que los SIG clásicos están muy implantados y hay una cierta resistencia al cambio.
- El problema de escoger un SIG libre entre los muchos que hay también se da en el caso de los SIG propietarios.
- Depender de una comunidad de desarrolladores es positivo: estará desarrollado lo más estándar y lo que falte puede desarrollarlo uno mismo.
- No hay duda que una de las principales bazas es la interoperabilidad y la utilización de estándares que, como hemos visto, es algo que actualmente se considera muy importante.

Preguntados los expertos sobre si utilizarían o no SIG libres, las actitudes también son diversas:

- Algunos no lo implementarían. El motivo que arguyen: simplemente porque sólo escogen productos por los que pagar y a quien culpar.
- Otros utilizarían SIG libres sin problemas.
- Para algunos, las condiciones para utilizar SIG libre son:
 - en caso que haga falta un producto horizontal, que requiera más conocimientos técnicos que funcionales.
 - en el caso del soporte decidido de alguna empresa, como los casos de Mapguide (Autodesk)¹², uDIG (Refractions)¹³; o de alguna institución relevante, como una universidad (el caso de MapServer¹⁴) de una administración pública (el caso de gvSIG¹⁵).

En cualquier caso, hay dos puntos a destacar de entre todas estas impresiones:

- El hecho de disponer o no del código a menudo no es un argumento decisivo: ciertas empresas desarrolladoras de software no tienen problemas en facilitar su código a sus clientes y otras ofrecen un soporte técnico capaz de suplir la disponibilidad del código.
- Es importante tener alguien que responda por el SIG. En este sentido casi se busca “vincularse” con alguna empresa, siempre y cuando dé buen servicio. Quizás el argumento más extendido entre los integradores en cuanto al SIG libre es que no hay nadie detrás que responda por él.

Vemos, pues, que las actitudes están muy dispersas, en lo que sí que hay bastante acuerdo es en los SIG libres han avanzado mucho en muy poco tiempo, con proyectos serios y bien financiados, pero la impresión es que todavía no están maduros y que aún queda camino por recorrer.

Es importante notar que éstas son impresiones extraídas de conversaciones con profesionales que integran diversos SIG (es decir, que no están ligados a una única plataforma) y que, en un momento u otro, han tenido que interactuar con algún SIG libre. Es decir, en ningún caso estamos diciendo que los argumentos a favor o en contra sean ciertos, sino simplemente que son impresiones que tienen los profesionales.

En este sentido se puede ver que hay un cierto desconocimiento de lo que significa el software libre: no se puede decir que no haya un responsable del software libre ni que dependa únicamente de la voluntad. Puede ser cierto en algunos casos, pero no siempre.

Lo que sí parece claro es que los profesionales perciben lo que ya se apuntaba anteriormente en el texto: los SIG libres se encuentran con los problemas propios de una aplicación vertical, y en general sólo se han consolidado proyectos con un fuerte respaldo económico y movidos por la filosofía del software libre.

12<http://usa.autodesk.com/adsk/servlet/index?id=7171990&siteID=123112>

13<http://udig.refractions.net/confluence/display/UDIG/Home>

14<http://mapserver.gis.umn.edu/>

15<http://www.gvsig.com/>

CONCLUSIONES

En este artículo se define qué es el software libre y qué lo distingue del software propietario y del gratuito. También se explica qué son las licencias *copyleft* y qué papel juega el software libre en nuestra sociedad. También se analizan conceptos que, a menudo, se confunden cuando se habla de software libre y estas confusiones son las que en ocasiones llevan a un usuario a decantarse por una solución propietaria en lugar de una libre.

Quizás la confusión más generalizada es que se depende de una comunidad de desarrolladores sin que haya una figura detrás que responda por el software. Hemos visto que esto no es así: no hay que confundir propietario con comercial. Una empresa puede desarrollar perfectamente software libre, ofrecer el mantenimiento correspondiente y cobrar por todo ello.

Al resolver la confusión anterior se ha resuelto otra: que en el software libre depende de una comunidad de desarrolladores. No es que se dependa de una comunidad de desarrolladores, sino que, según la licencia, cualquiera se puede beneficiar de los desarrollos de cualquiera.

Es importante notar lo dicho en el párrafo anterior: "según la licencia". Y es que una confusión muy común es creer que para que un desarrollo sea libre, es obligatorio hacerlo público. No hay que olvidar que para muchas empresas hacer público un código representa ofrecer a la competencia su hecho diferencial. Hay muchos tipos de licencia, más o menos restrictivas, que cumplen con los cuatro requisitos del software libre y que no obligan, necesariamente, a hacer públicos los desarrollos. Además, una empresa puede escoger proteger una parte del código si la licencia lo permite. En cualquier caso, no hay que olvidar tampoco que, en un mundo de software libre, el coste de liberar un código, puede ser compensado por el beneficio del código liberado por la competencia. En otro artículo presentado en estas Jornadas se profundiza en estos temas.

Además, no hay que olvidar que los modelos de negocio del software libre centran la mayoría de los posibles ingresos en los servicios (y no en la venta de licencias que restringen las libertades). De esta manera, una empresa que contribuya a desarrollos de software libre puede obtener sus beneficios ofreciendo servicios de personalización y consultoría sobre un código que nadie conoce mejor que ella, por muy libre que sea. Otra confusión muy común es creer que el software libre es gratis. Por lo dicho en los párrafos anteriores queda claro que esto no es así.

En cuanto a la función social del software libre, se ha visto que éste está cada vez más a la orden del día. Sin embargo, en este punto se han distinguido dos tipos de aplicaciones: horizontales y verticales. Las primeras son aplicaciones necesarias para cualquier actividad y tienen un elevado número potencial de usuarios y desarrolladores; las segundas son muy específicas y el número de usuarios y desarrolladores potenciales es muy reducido. El tipo de aplicación se ve reflejado en el tipo de desarrollo libre: las primeras pueden depender de una comunidad de desarrolladores mientras las segundas necesitan, en muchos casos, de un fuerte respaldo económico para su desarrollo exitoso.

Los SIG se encuentra entre las aplicaciones verticales por lo que, a todas las dudas y confusiones propias del software libre, se suman los problemas habituales de los desarrollos libres en aplicaciones verticales. Sí que es cierto, como se ha comentado, que los SIG están eclosionando a nivel social. Pero también es cierto que

sólo en lo que a usuario final de consulta se refiere, por lo que en ningún caso se puede hablar de que los SIG se estén transformando en una aplicación horizontal: las aplicaciones y desarrollos SIG siguen correspondiendo a empresas e instituciones. De hecho, los SIG libres que más están calando son aquellos que cuentan con alguna empresa o institución que los respalde, y que se corresponde con el modelo de desarrollos libres de aplicaciones verticales.

Otro elemento importante del artículo ha sido recoger las impresiones de integradores SIG en cuanto al SIG libre. No hay que olvidar que son los profesionales y usuarios los que van a trabajar con él y es su percepción la que, al fin y al cabo, puede decantar la balanza a favor de una solución u otra. En este sentido podemos extraer dos conclusiones importantes. Por un lado, que argumentos propios de muchos desarrollos horizontales, como no depender de una empresa o disponer del código, no aparecen al hablar de los SIG libres. De hecho, al ser una aplicaciones verticales presentes sobre todo en empresas e instituciones, se busca sobre todo calidad, servicio y beneficio. Por otro lado, al margen de alguna pequeña confusión, los profesionales están siguiendo de cerca el software libre y muchos de ellos, quizás los más atrevidos, ya los ven como alternativas reales al software propietario. Algo que hace sólo algunos años era impensable.

Como conclusión final decir que al margen de discusiones e impresiones, el software libre es la apuesta por una filosofía y una forma distinta de evaluar los beneficios. En el caso de los SIG libres está apuesta se percibe de forma aún más perceptible puesto que las principales soluciones disponibles actualmente han sido consecuencia de la decisión política de una empresa o institución.

Para acabar, comentar que en este artículo se han soslayado, sin embargo, los temas legales relacionados con el software libre en general y los SIG en particular, o cómo se gesta un proyecto de software libre. En un segundo artículo presentado en estas Jornadas se profundiza en estos temas y se analiza en profundidad qué significan realmente, las cuatro libertades que definen el software libre y su materialización en diferentes licencias.

AGRADECIMIENTOS

Gracias a Pablo Costa Vallés y Jordi Rovira Jofre, de AURENSIS, S.L; a Luis Aso y Mercè Escolà de ALTRAN; a Òscar Fonts Bartolomé, del Institut Cartogràfic de Catalunya; y a Antonio Francisco Padilla Cano de I&IMS.

REFERENCIAS

1. González Barahona, Jesús M.; Seoane Pascual, Joaquín, Robles Martínez, G. (2003) *Introducción al software libre*. Barcelona. Editorial UOC. <http://www.uoc.edu/masters/oficiales/img/693.pdf>.
2. Bain, Malcolm; Gallego Rodríguez, Manuel; Martínez Ribas, Manuel; Rius Sanjuán, Judit (2004). *Aspectos legales y de explotación del software libre*. Barcelona. Editorial UOC. <http://www.uoc.edu/masters/oficiales/img/908.pdf>, <http://www.uoc.edu/masters/oficiales/img/908.pdf>.
3. Allué Pont, Eduard (2005), *Creació d'un Sistema d'Informació Geogràfica de Carreteres*. Projecte Final de Carrera. UOC.

4. Lantada Zarzosa, Nieves; Núñez Andrés, M. Amparo. (2003) Sistemas de información geográfica. Prácticas con Arc View. Barcelona. Edicions de la UPC.
5. Universidad de Cádiz (2004), *Introducción a los SIG*. Universidad de Cádiz. España. <http://www.uca.es/dept/filosofia/TEMA%201.pdf>; <http://www.uca.es/dept/filosofia/TEMA%202.pdf>.